Modyfikacje/hybrydyzacje algorytmu PSO w zadaniu optymalizacji globalnej wielowymiarowej funkcji ciaglej

PSO-DE Hybrid

Jakub Ruszkowski, Mateusz Kaczmarski

ABSTRACT

Dokumentacja uzyskanych wynikow hybrydy PSO-DE

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, PSODE, Optymalizacja wielowymiarowej funkcji ciaglej

1. CPU TIMING

In order to evaluate the CPU timing of the algorithm, we have run the PSO-DE Hybrid on the function f_8 with restarts for at least 30 seconds and until a maximum budget equal to $400(D+2)$ is reached. The code was run on a Mac Intel(R) Core(TM) i5-2400S CPU @ 2.50GHz with 1 processor and 4 cores. The time per function evaluation for dimensions 2, 3, 5, 10, 20, 40 equals $x.x$, $x.x$, $x.x$, xx, xxx, and xxx milliseconds respectively.

repeat the above for the second algorithm

2. RESULTS

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

GECCO'13, July 6-10, 2013, Amsterdam, The Netherlands.

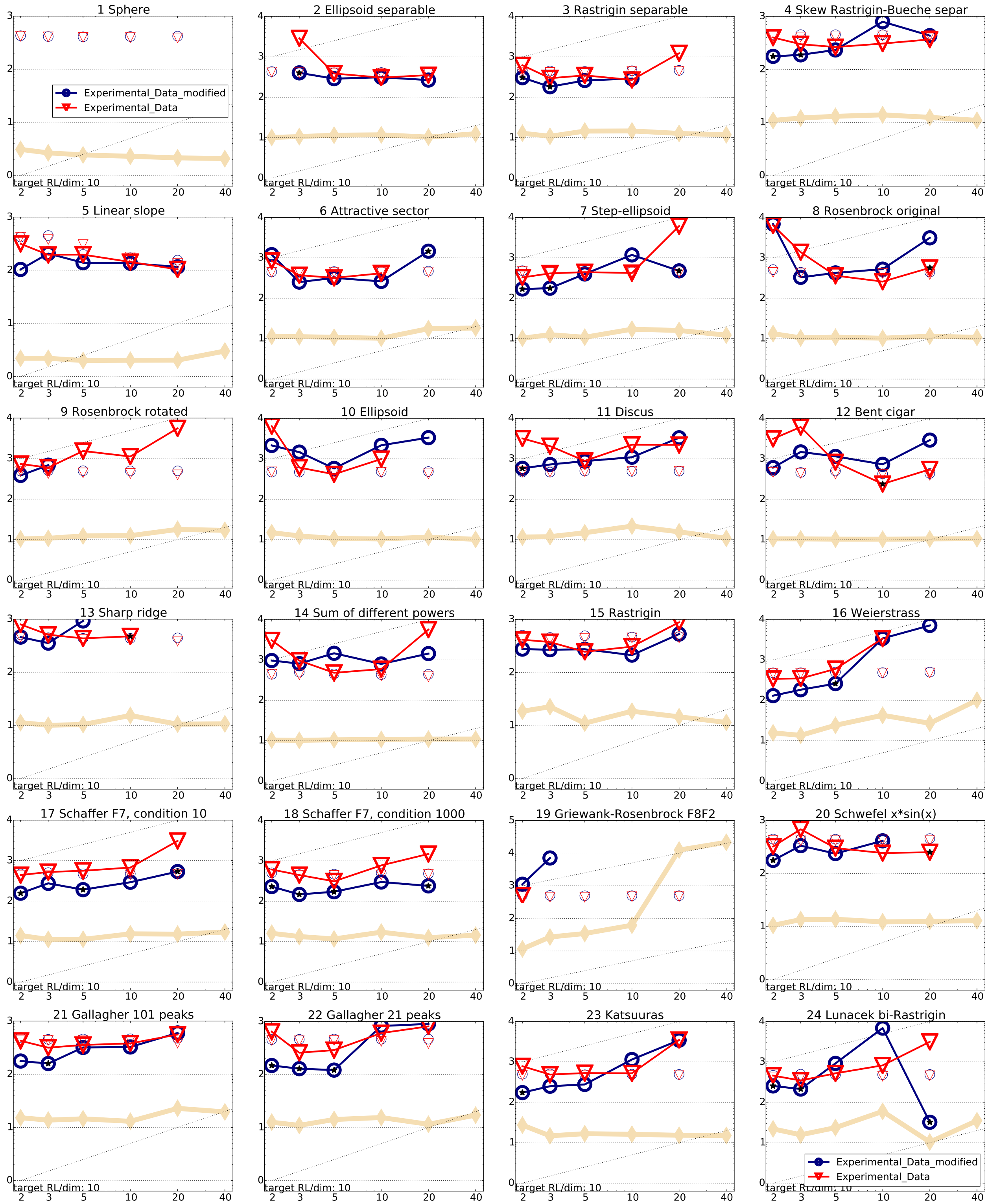

Figure 1: Expected running time (ERT in number of f-evaluations as $\log_{10}$ value) divided by dimension versus dimension. The target function value is chosen such that the bestGECCO2009 artificial algorithm just failed to achieve an ERT of $10 \times \mathrm{DIM}$. Different symbols correspond to 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. Black stars indicate a statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Legend: $\circ$:Experimental Data modified, $\triangledown$:Experimental Data.

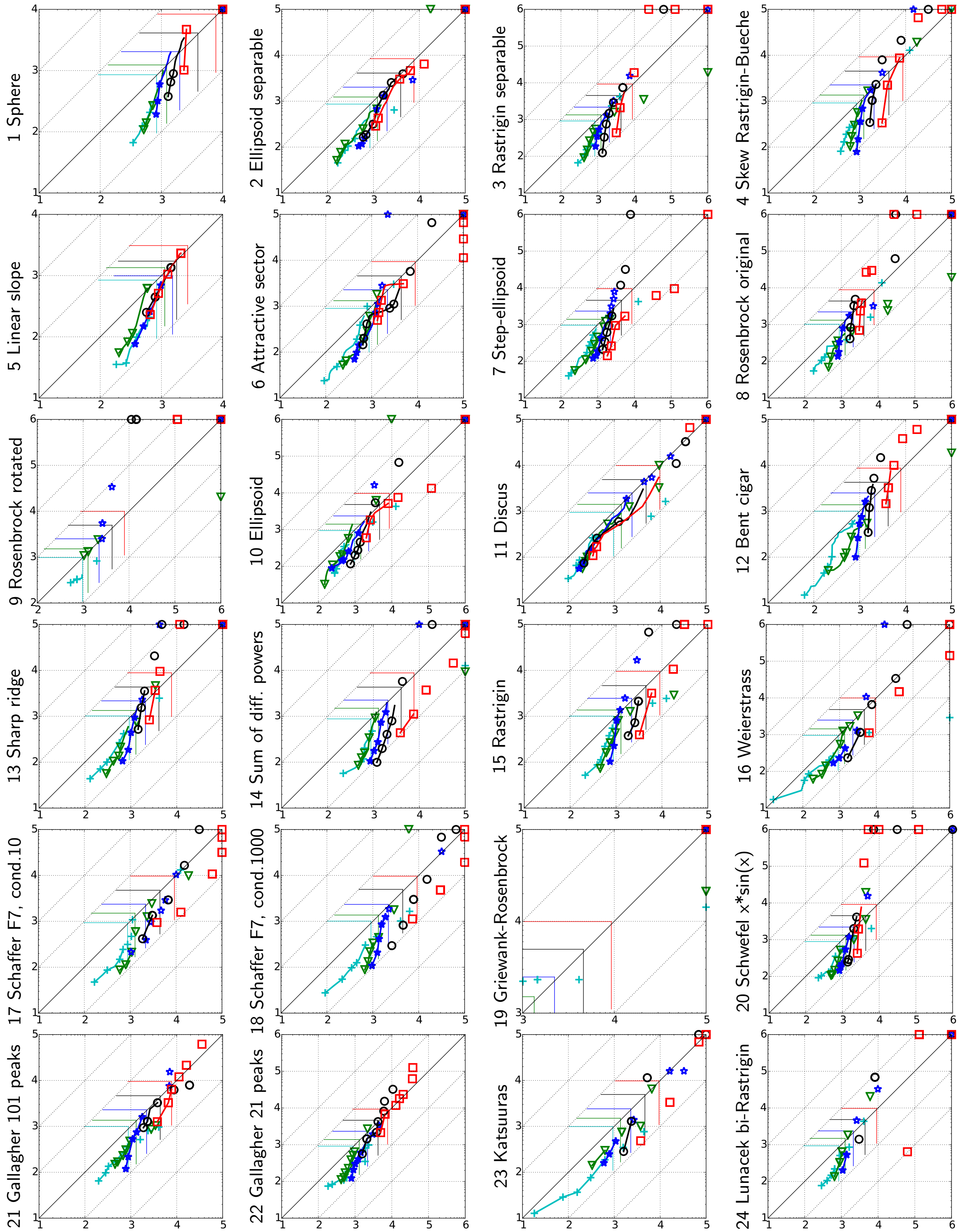

Figure 2: Expected running time (ERT in $\log_{10}$ of number of function evaluations) of Experimental Data modified (y-axis) versus Experimental Data (x-axis) for 8 runlength-based target function values for budgets between $0.5 \times \mathrm{DIM}$ and $50 \times \mathrm{DIM}$ evaluations. Each runlength-based target f-value is chosen such that the ERTs of the bestGECCO2009 artificial algorithm for the given and a slightly easier target bracket the reference budget. Markers on the upper or right edge indicate that the respective target value was never reached. Markers represent dimension: 2:+, 3:▽, 5:⋆, 10:○, 20:□, 40:◇.

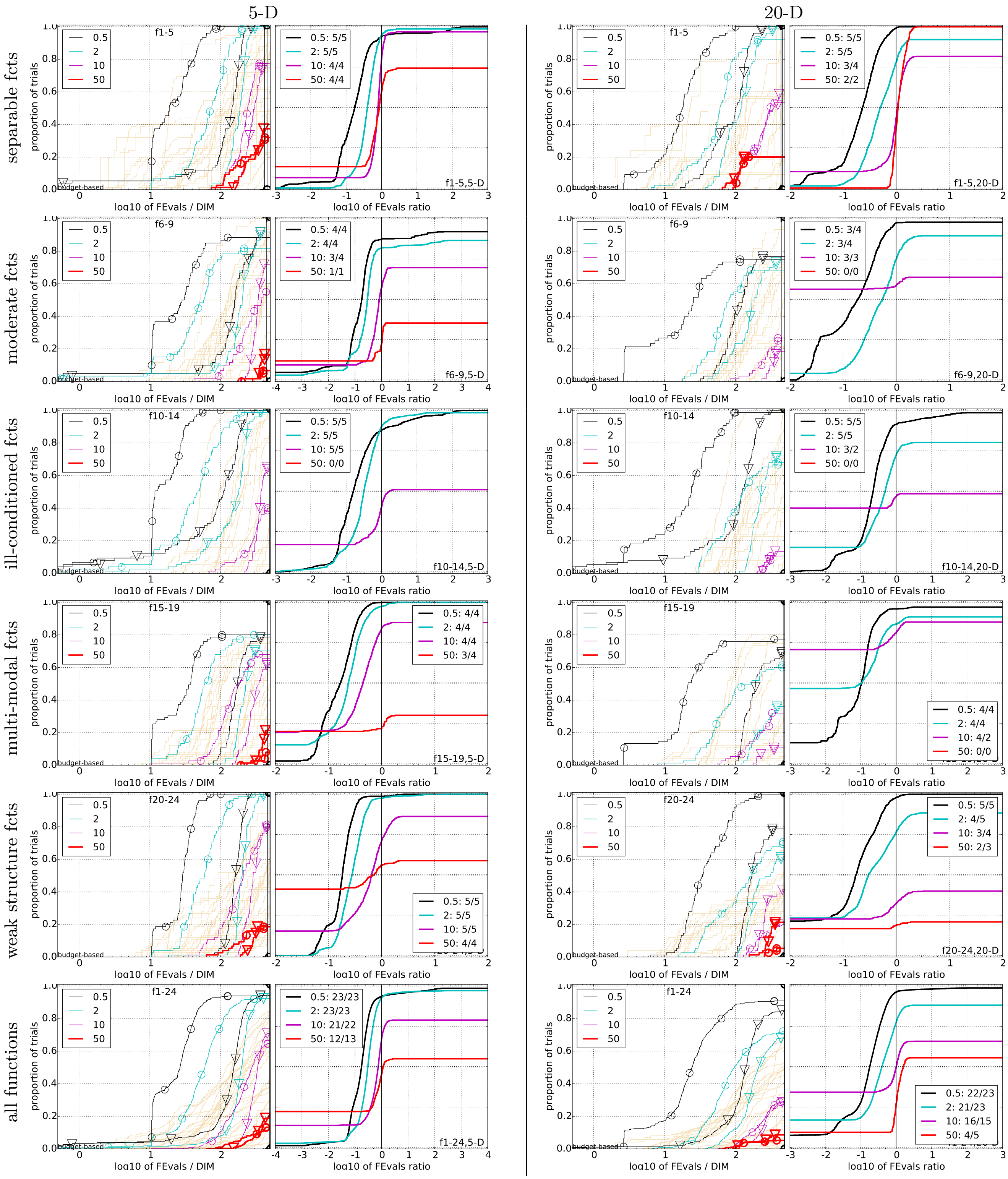

Figure 3: Empirical cumulative distributions (ECDF) of run lengths and speed-up ratios in 5-D (left) and 20-D (right). Left sub-columns: ECDF of the number of function evaluations divided by dimension D (FEvals/D) to fall below $f_{\rm opt} + \Delta f$ for Experimental Data modified ($\circ$) and Experimental Data ($\triangledown$) where Δf is the target just not reached by the GECCO-BBOB-2009 best algorithm within a budget of $k \times \mathrm{DIM}$ evaluations, with k being the value in the legend. Right sub-columns: ECDF of FEval ratios of Experimental Data modified divided by Experimental Data for run-length-based targets; 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 target budget of $k \times \mathrm{DIM}$ evaluations and, after the colon, the number of functions that were solved in at least one trial (Experimental Data modified first).

5-D

#FEs/D	0.5	1.2	3	10	50	#succ
f1	*2.5e+1*:4.8	*1.6e+1*:7.6	*1.0e-8*:12	*1.0e-8*:12	*1.0e-8*:12	15/15
1: Exp	**40**(12)^{*3}	**42**(21)^{*3}	∞	∞	∞*2000*	0/15
2: Exp	171(35)	115(27)	∞*	∞*	∞***2000****	0/15
f2	*1.6e+6*:2.9	*4.0e+5*:11	*4.0e+4*:15	*6.3e+2*:58	*1.0e-8*:95	15/15
1: Exp	**37**(18)^{*2}	**14**(7)^{*3}	**22**(11)^{*3}	25(5)	∞*2000*	0/15
2: Exp	166(129)	58(24)	63(21)	34(4)	∞*1900*	0/15
f3	*1.6e+2*:4.1	*1.0e+2*:15	*6.3e+1*:23	*2.5e+1*:73	*1.0e+1*:716	15/15
1: Exp	**45**(45)^{*3}	**24**(12)^{*3}	**23**(10)^{*3}	18(4)	22(26)	2/15
2: Exp	211(51)	66(14)	47(14)	24(10)	10(9)	4/15
f4	*2.5e+2*:2.6	*1.6e+2*:10	*1.0e+2*:19	*4.0e+1*:65	*1.6e+1*:434	15/15
1: Exp	**30**(18)^{*3}	**15**(8)^{*3}	**19**(14)^{*3}	18(13)	10(9)	7/15
2: Exp	328(85)	96(27)	55(10)	20(5)	7.1(5)	9/15
f5	*6.3e+1*:4.0	*4.0e+1*:10	*1.0e-8*:10	*1.0e-8*:10	*1.0e-8*:10	15/15
1: Exp	19(18)	15(13)	69(23)	69(24)	69(28)	15/15
2: Exp	93(79)	52(31)	98(36)	98(27)	98(30)	15/15
f6	*1.0e+5*:3.0	*2.5e+4*:8.4	*1.0e+2*:16	*2.5e+1*:54	*2.5e-1*:254	15/15
1: Exp	23(21)	13(12)	**27**(12)^{*2}	30(46)	∞*2100*	0/15
2: Exp	140(138)	58(38)	68(17)	30(14)	∞*2100*	0/15
f7	*1.6e+2*:4.2	*1.0e+2*:6.2	*2.5e+1*:20	*4.0e+0*:54	*1.0e+0*:324	15/15
1: Exp	**29**(20)^{*3}	**23**(17)^{*3}	**22**(14)^{*3}	37(17)	24(24)	4/15
2: Exp	179(47)	147(45)	62(15)	41(49)	8.8(4)	10/15
f8	*1.0e+4*:4.6	*6.3e+3*:6.8	*1.0e+3*:18	*6.3e+1*:54	*1.6e+0*:258	15/15
1: Exp	**30**(19)^{*3}	**26**(17)^{*3}	40(63)	39(38)	∞*2100*	0/15
2: Exp	174(45)	124(22)	59(10)	33(13)	∞*2000*	0/15
f9	*2.5e+1*:20	*1.6e+1*:26	*1.0e+1*:35	*4.0e+0*:62	*1.6e-2*:256	15/15
1: Exp	127(188)	212(214)	302(321)	∞	∞*2200*	0/15
2: Exp	131(139)	102(153)	89(62)	125(138)	∞*2000*	0/15
f10	*2.5e+6*:2.9	*6.3e+5*:7.0	*2.5e+5*:17	*6.3e+3*:54	*2.5e+1*:297	15/15
1: Exp	31(25)	20(11)	**10**(9)*	54(59)	∞*2300*	0/15
2: Exp	81(58)	62(57)	34(30)	39(45)	∞*2300*	0/15
f11	*1.0e+6*:3.0	*6.3e+4*:6.2	*6.3e+2*:16	*6.3e+1*:74	*6.3e-1*:298	15/15
1: Exp	19(9)	19(12)	44(68)	59(51)	∞*2300*	0/15
2: Exp	57(67)	39(25)	56(30)	60(59)	∞*2300*	0/15
f12	*4.0e+7*:3.6	*1.6e+7*:7.6	*4.0e+6*:19	*1.6e+4*:52	*1.0e+0*:268	15/15
1: Exp	**28**(22)^{*3}	**35**(15)^{*3}	39(61)	112(215)	∞*2100*	0/15
2: Exp	226(65)	121(33)	57(13)	80(49)	∞*2000*	0/15
f13	*1.0e+3*:2.8	*6.3e+2*:8.4	*4.0e+2*:17	*6.3e+1*:52	*6.3e-2*:264	15/15
1: Exp	**38**(20)^{*3}	**22**(13)^{*3}	**26**(12)^{*3}	89(56)	∞*2100*	0/15
2: Exp	236(63)	105(21)	61(8)	42(13)	∞*2100*	0/15
f14	*1.6e+1*:3.0	*1.0e+1*:10	*6.3e+0*:15	*2.5e-1*:53	*1.0e-5*:251	15/15
1: Exp	**35**(26)^{*3}	**18**(15)^{*3}	**18**(6)^{*3}	138(102)	∞*2100*	0/15
2: Exp	284(129)	105(47)	81(32)	45(23)	∞*2000*	0/15
f15	*1.6e+2*:3.0	*1.0e+2*:13	*6.3e+1*:24	*4.0e+1*:55	*1.6e+1*:289	5/5
1: Exp	**34**(33)^{*3}	**17**(14)^{*3}	**33**(34)*	25(22)	58(81)	2/15
2: Exp	247(66)	69(27)	42(11)	22(6)	**10**(5)^{*2}	10/15
f16	*4.0e+1*:4.8	*2.5e+1*:16	*1.6e+1*:46	*1.0e+1*:120	*4.0e+0*:334	15/15
1: Exp	**35**(17)^{*3}	**15**(5)^{*3}	**9.2**(5)^{*3}	**11**(3)^{*2}	∞*2300*	0/15
2: Exp	140(48)	62(12)	30(11)	25(12)	50(63)	2/15
f17	*1.0e+1*:5.2	*6.3e+0*:26	*4.0e+0*:57	*2.5e+0*:110	*6.3e-1*:412	15/15
1: Exp	**41**(37)^{*3}	**15**(12)^{*3}	**17**(22)^{*2}	16(14)	25(15)	3/15
2: Exp	202(109)	85(63)	50(60)	43(61)	24(39)	3/15
f18	*6.3e+1*:3.4	*4.0e+1*:7.2	*2.5e+1*:20	*1.6e+1*:58	*1.6e+0*:318	15/15
1: Exp	**32**(22)^{*3}	**29**(9)^{*3}	**21**(13)^{*2}	**15**(15)*	104(81)	1/15
2: Exp	282(51)	174(99)	69(59)	27(6)	99(85)	1/15
f19	*1.6e-1*:172	*1.0e-1*:242	*6.3e-2*:675	*4.0e-2*:3078	*2.5e-2*:4946	15/15
1: Exp	∞	∞	∞	∞	∞*2500*	0/15
2: Exp	∞	∞	∞	∞	∞*2100*	0/15
f20	*6.3e+3*:5.1	*4.0e+3*:8.4	*4.0e+1*:15	*2.5e+0*:69	*1.0e+0*:851	15/15
1: Exp	**28**(7)^{*3}	**22**(7)^{*3}	**27**(8)^{*3}	17(9)	∞*2200*	0/15
2: Exp	162(36)	107(19)	75(10)	22(3)	∞*2000*	0/15
f21	*4.0e+1*:3.9	*2.5e+1*:11	*1.6e+1*:31	*6.3e+0*:73	*1.6e+0*:347	5/5
1: Exp	**30**(19)^{*3}	**21**(6)^{*3}	**17**(21)^{*2}	22(7)	22(19)	4/15
2: Exp	198(43)	83(9)	35(16)	25(17)	20(22)	4/15
f22	*6.3e+1*:3.6	*4.0e+1*:15	*2.5e+1*:32	*1.0e+1*:71	*1.6e+0*:341	5/5
1: Exp	**34**(18)^{*3}	**14**(4)^{*3}	**9.4**(4)^{*3}	**8.5**(8)*	10(10)	7/15
2: Exp	220(39)	61(14)	31(11)	20(11)	13(13)	6/15
f23	*1.0e+1*:3.0	*6.3e+0*:9.0	*4.0e+0*:33	*2.5e+0*:84	*1.0e+0*:518	15/15
1: Exp	**53**(18)^{*2}	**28**(7)^{*2}	**14**(5)*	17(14)	31(21)	2/15
2: Exp	198(55)	86(50)	32(10)	32(24)	64(69)	1/15
f24	*6.3e+1*:15	*4.0e+1*:37	*4.0e+1*:37	*2.5e+1*:118	*1.6e+1*:692	15/15
1: Exp	**14**(9)^{*3}	**15**(6)^{*3}	**15**(5)^{*3}	39(69)	47(48)	1/15
2: Exp	75(9)	38(11)	38(10)	22(13)	14(11)	3/15

20-D

#FEs/D	0.5	1.2	3	10	50	#succ
f1	*6.3e+1*:24	*4.0e+1*:42	*1.0e-8*:43	*1.0e-8*:43	*1.0e-8*:43	15/15
1: Exp	**42**(13)^{*3}	112(209)	∞	∞	∞*8200*	0/15
2: Exp	97(9)	61(16)	∞^{*3}	∞^{*3}	∞***7700***^{*3}	0/15
f2	*4.0e+6*:29	*2.5e+6*:42	*1.0e+5*:65	*1.0e+4*:207	*1.0e-8*:412	15/15
1: Exp	**10**(4)^{*3}	**10**(9)^{*3}	39(4)	26(5)	∞*8400*	0/15
2: Exp	39(10)	30(7)	50(13)	34(29)	∞*7600*	0/15
f3	*6.3e+2*:33	*4.0e+2*:44	*1.6e+2*:109	*1.0e+2*:255	*2.5e+1*:3277	15/15
1: Exp	**13**(10)^{*3}	**48**(127)*	173(206)	∞	∞*9100*	0/15
2: Exp	97(131)	93(75)	90(15)	96(99)	∞*8600*	0/15
f4	*6.3e+2*:22	*4.0e+2*:91	*2.5e+2*:250	*1.6e+2*:332	*6.3e+1*:1927	15/15
1: Exp	**16**(9)^{*3}	**25**(11)*	35(21)	199(193)	∞*9100*	0/15
2: Exp	142(51)	44(17)	29(22)	57(62)	∞*8400*	0/15
f5	*2.5e+2*:19	*1.6e+2*:34	*1.0e-8*:41	*1.0e-8*:41	*1.0e-8*:41	15/15
1: Exp	**12**(7)^{*3}	**15**(5)^{*2}	57(8)	57(13)	57(6)	15/15
2: Exp	35(14)	26(5)	51(17)	51(11)	51(12)	15/15
f6	*2.5e+5*:16	*6.3e+4*:43	*1.6e+4*:62	*1.6e+2*:353	*1.6e+1*:1078	15/15
1: Exp	**31**(17)^{*3}	31(7)	36(14)	**83**(52)^{*2}	∞*9100*	0/15
2: Exp	83(46)	38(23)	29(10)	∞	∞*8400*	0/15
f7	*1.0e+3*:11	*4.0e+2*:39	*2.5e+2*:74	*6.3e+1*:319	*1.0e+1*:1351	15/15
1: Exp	**13**(18)^{*3}	**24**(9)^{*2}	**23**(8)^{*2}	**30**(29)*	∞*9300*	0/15
2: Exp	173(27)	78(20)	74(24)	384(431)	∞*8300*	0/15
f8	*4.0e+4*:19	*2.5e+4*:35	*4.0e+3*:67	*2.5e+2*:231	*1.6e+1*:1470	15/15
1: Exp	**37**(43)^{*3}	**67**(125)*	243(126)	271(301)	∞*8500*	0/15
2: Exp	163(25)	93(18)	65(21)	**49**(20)^{*2}	∞***7600***^{*2}	0/15
f9	*1.0e+2*:357	*6.3e+1*:560	*4.0e+1*:684	*2.5e+1*:756	*1.0e+1*:1716	15/15
1: Exp	∞	∞	∞	∞	∞*9300*	0/15
2: Exp	314(144)	202(147)	167(154)	∞	∞*7600*	0/15
f10	*1.6e+6*:15	*1.0e+6*:27	*4.0e+5*:70	*6.3e+4*:231	*4.0e+3*:1015	15/15
1: Exp	**40**(31)^{*3}	**67**(169)*	73(86)	290(235)	∞*9500*	0/15
2: Exp	135(30)	95(42)	113(69)	∞	∞*8300*	0/15
f11	*4.0e+4*:11	*2.5e+3*:27	*1.6e+2*:313	*1.0e+2*:481	*1.0e+1*:1002	15/15
1: Exp	10(9)	20(10)	213(168)	∞	∞*9800*	0/15
2: Exp	30(32)	52(32)	142(234)	∞	∞*9600*	0/15
f12	*1.0e+8*:23	*6.3e+7*:39	*2.5e+7*:76	*4.0e+6*:209	*1.0e+1*:1042	15/15
1: Exp	**63**(112)^{*2}	82(87)	132(155)	279(345)	∞*8300*	0/15
2: Exp	161(32)	107(22)	73(17)	53(36)	∞*7600*	0/15
f13	*1.6e+3*:28	*1.0e+3*:64	*6.3e+2*:79	*4.0e+1*:211	*2.5e+0*:1724	15/15
1: Exp	**30**(22)^{*3}	57(98)	121(133)	∞	∞*8600*	0/15
2: Exp	92(21)	53(11)	55(13)	∞^{*2}	∞***7600***^{*2}	0/15
f14	*2.5e+1*:15	*1.6e+1*:42	*1.0e+1*:75	*1.6e+0*:219	*6.3e-4*:1106	15/15
1: Exp	**30**(27)^{*3}	**26**(20)^{*3}	**50**(116)^{*2}	130(148)	∞*8600*	0/15
2: Exp	264(92)	177(179)	187(85)	512(614)	∞*7600*	0/15
f15	*6.3e+2*:15	*4.0e+2*:67	*2.5e+2*:292	*1.6e+2*:846	*1.0e+2*:1671	15/15
1: Exp	**26**(25)^{*3}	47(68)	36(102)	∞	∞*9400*	0/15
2: Exp	210(71)	88(166)	61(96)	37(38)	∞*8600*	0/15
f16	*4.0e+1*:26	*2.5e+1*:127	*1.6e+1*:540	*1.6e+1*:540	*1.0e+1*:1384	15/15
1: Exp	**42**(99)^{*2}	117(74)	263(499)	263(342)	∞*9700*	0/15
2: Exp	242(235)	332(275)	∞	∞	∞*9200*	0/15
f17	*1.6e+1*:11	*1.0e+1*:63	*6.3e+0*:305	*4.0e+0*:468	*1.0e+0*:1030	15/15
1: Exp	**89**(16)^{*2}	**25**(77)^{*2}	**35**(35)*	**68**(83)*	∞*9500*	0/15
2: Exp	362(33)	202(202)	201(332)	∞	∞*8800*	0/15
f18	*4.0e+1*:116	*2.5e+1*:252	*1.6e+1*:430	*1.0e+1*:621	*4.0e+0*:1090	15/15
1: Exp	**10**(3)^{*2}	**19**(13)*	**45**(71)*	111(54)	∞*9500*	0/15
2: Exp	63(37)	118(125)	∞	∞	∞*9000*	0/15
f19	*1.6e-1*:2.5e5	*1.0e-1*:3.4e5	*6.3e-2*:3.4e5	*4.0e-2*:3.4e5	*2.5e-2*:3.4e5	3/15
1: Exp	∞	∞	∞	∞	∞*9900*	0/15
2: Exp	∞	∞	∞	∞	∞*9200*	0/15
f20	*1.6e+4*:38	*1.0e+4*:42	*2.5e+2*:62	*2.5e+0*:250	*1.6e+0*:2536	15/15
1: Exp	**11**(3)^{*3}	**46**(4)*	607(610)	∞	∞*8600*	0/15
2: Exp	66(10)	65(9)	**59**(10)*	**20**(4)^{*3}	**3.9**(2)^{*3}	10/15
f21	*6.3e+1*:36	*4.0e+1*:77	*4.0e+1*:77	*1.6e+1*:456	*4.0e+0*:1094	15/15
1: Exp	**35**(16)^{*3}	43(16)	43(23)	26(33)	56(86)	2/15
2: Exp	106(67)	86(58)	86(104)	25(18)	32(40)	3/15
f22	*6.3e+1*:45	*4.0e+1*:68	*4.0e+1*:68	*1.6e+1*:231	*6.3e+0*:1219	15/15
1: Exp	**47**(13)^{*2}	100(127)	100(60)	78(183)	51(42)	2/15
2: Exp	110(65)	97(85)	97(102)	70(65)	30(41)	3/15
f23	*6.3e+0*:29	*4.0e+0*:118	*2.5e+0*:306	*2.5e+0*:306	*1.0e+0*:1614	15/15
1: Exp	**17**(10)^{*3}	**28**(36)*	226(207)	226(219)	∞*9700*	0/15
2: Exp	131(59)	138(89)	229(207)	229(170)	∞*9400*	0/15
f24	*2.5e+2*:208	*1.6e+2*:918	*1.0e+2*:6628	*6.3e+1*:9885	*4.0e+1*:31629	15/15
1: Exp	**3.1**(1)^{*3}	∞	∞	∞	∞*9700*	0/15
2: Exp	302(408)	144(83)	∞	∞	∞*9000*	0/15

Table 1: Expected running time (ERT in number of function evaluations) divided by the respective best ERT measured during BBOB-2009 in dimensions 5 (left) and 20 (right). The ERT and in braces, as dispersion measure, the half difference between 90 and 10%-tile of bootstrapped run lengths appear for each algorithm and run-length based target, the corresponding best ERT (preceded by the target Δf-value in *italics*) in the first row. #succ is the number of trials that reached the target value of the last column. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. 1:Exp is Experimental Data modified and 2:Exp is Experimental Data. 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 algorithm of BBOB-2009.