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

Jakub Ruszkowski, Mateusz Kaczmarski

19 maja 2015

1 Wstęp

Hybrydyzacja algorytmu *Particle Swarm Optimization* (PSO) z algorytmem *Differential Evolution* (DE).

2 Opis algorytmu

2.1 Optymalizacja Rojem Cząsteczek

Optymalizacja rojem cząsteczek (ang. Particle Swarm Optimization) jest algorytmem meta heurystycznym służącym do optymalizacji zadanego problemu. Inspiracją dla tego algorytmu była obserwacja zachowań organizmów żywych w populacjach (np. kolonia mrówek, ławica ryb). Cząsteczka (osobnik roju) posiada swoją aktualną pozycję, prędkość oraz najlepszą pozycję, której wartość zostaje zmieniona gdy cząstka znalazła położenie lepiej ocenione. Na początku wartości pozycji oraz prędkości inicjowane są losowymi liczbami. W każdej kolejnej iteracji algorytmu cząsteczki przemieszczają się do nowych położeń symulując adaptację roju do środowiska. Aktualizowane są wówczas najlepsze pozycje cząstek oraz wyznaczany jest lider roju, czyli osobnik o dotychczasowym najlepszym położeniu. Dla każdej cząsteczki obliczany jest także nowy wektor prędkości na podstawie jej bieżącej prędkości oraz położenia lidera roju. Iteracje są powtarzane dopóki nie spełniony zostanie warunek stopu.

Schemat działania algorytmu PSO:

dla każdej cząsteczki w roju:

zainicjuj wartości położeń i prędkości liczbami losowymi;

while(!stop)

{

za pomocą odpowiedniej funkcji dopasowania dokonaj oceny położenia cząstek w roju;

wyznacz lidera roju;

dla każdej cząsteczki w roju:

zaktualizuj wektor prędkości oraz położenie;

}

Wektor położenia: $X_i = (x_{i1}, x_{i2}, ..., x_{iD})$
Wektor prędkości: $V_i = (v_{i1}, v_{i2}, ..., v_{iD})$
Zmiana położenia w poszczególnych iteracjach:

$x_{id} = x_{id} + v_{id}$

Zmiana wartości prędkości w poszczególnych iteracjach:

$v_{id} = \omega \cdot v_{id} + c_1 \cdot r_1 \cdot (p_{id} - x_{id}) + c_2 \cdot r_2 \cdot (p_{gd} - x_{id})$,

gdzie ω – stały współczynnik określający stopień kontynuacji ruchu cząstki w dotychczasowym kierunku, c_1, c_2 – współczynniki akceleracji, r_1, r_2 – losowe liczby z przedziału [0,1], p_i, p_g - odpowiednio najlepsza dotychczasowa pozycja cząstki i oraz globalna najlepsza dotychczasowa pozycja wszystkich cząstek.
Algorytm PSO parametryzowany jest zatem trzema wartościami: ω, c_1 oraz c_2.

2.2 Ewolucja Różnicowa

Algorytm ewolucji różnicowej jest podobnie jak PSO meta heurystycznym algorytmem optymalizacji numerycznej. Algorytm operuje na populacji osobników (odpowiednik cząsteczek w PSO). Każdy osobnik, analogicznie do poprzedniego algorytmu jest reprezentowany przez D-wymiarowy wektor liczb rzeczywistych. W każdym kroku algorytmu, dla każdego osobnika x_i jest tworzony osobnik próbny u_i, który powstaje poprzez zastosowanie operatorów mutacji oraz krzyżowania. Następnie u_i jest porównywany z x_i i jeśli jego dopasowanie jest lepsze, to wtedy zastępuje x_i w populacji. W przeciwnym przypadku osobnik u_i jest odrzucany.
Wynikiem mutacji jest wektor m_i otrzymany w następujący sposób:

$m_i = x_{r_1} + F \cdot (x_{r_2} - x_{r_3})$,

gdzie $0 \leq F \leq 1$ jest stałym parametrem, zwanym współczynnikiem aplifikacji, natomiast r_1, r_2, r_3 to trzy losowo wygenerowane numery osobników, przy czym spełniona jest zależność $i \neq r_1 \neq r_2 \neq r_3$. Tak powstały wektor m_i jest nazywany osobnikiem mutantem.
Z kolei wynikiem krzyżowania operującego na rodzicu x_i oraz mutancie m_i jest osobnik próbny u_i, którego każdy element jest wyznaczony w następujący sposób:

$u_{i,j} = \begin{cases} m_{i,j} \ gdy \ rnd_j < CR \ lub \ j{=}d \\ x_{i,j} \ w \ przeciwnym \ przyadku \end{cases}$

gdzie rnd_j jest liczbą losową z przedziału [0, 1) losowaną niezależnie dla każdego j, natomiast $0 \leq CR \leq 1$ jest stałym parametrem algorytmu, a d jest losowym numerem elementu wektora.
Algorytm DE parametryzowany jest zatem dwiema wartościami: F oraz CR.

2.3 Algorytm hybrydowy

Dane wejściowe:
PSO_DE(*JNIfgeneric* fgeneric, *int* dim, *doubl*e maxfunevals, *Random* rand),
gdzie:
JNIfgeneric fgeneric – klasa z danymi definiującymi problem,
int dim – wymiar problemu,
double maxfunevals – maksymalna liczba iteracji,
Random rand – generator liczb losowych

Oznaczenia używane w algorytmie:
N - liczebność populacji
D - wymiar wektora osobnika
$rand()$ - liczba losowa z $[0,1]$
x_i – wektor o wymiarze D definiujący położenie osobnika i
p_i – najlepszy dotychczasowy wektor położenia osobnika i (p_g - najlepszy globalny)
v_i – wektor prędkości osobnika i

Schemat działania algorytmu hybrydowego łączącego DE oraz PSO:

```
    przypisz losowo początkowe wartości: x_i, v_i oraz p_i i p_g dla i = 1,2,...,N
while(!stop)
{
    for i = 1 to N
    {
        wybierz losowo r_1, r_2, r_3, takie że i ≠ r_1 ≠ r_2 ≠ r_3
        m_i = x_{r_1} + F · (x_{r_2} − x_{r_3})
        for j = 1 to D
        {
            wybierz losowo j_rand z przedziału [1, D]
            if (rand() < CR or j == j_rand)
                u[j] = m_i[j]
            else
                u[j] = x_i[j]
        }
        if (f(u) < f(x_i))
            x_i = u
        else
        {
            użyj PSO do wyznaczenia nowego kandydata – TX_i
            {
                oblicz wektor prędkości cząsteczki x_i:
                v_i = ω · v_i + c_1 · r_1 · (p_i − x_i) + c_2 · r_2 · (p_g − x_i)
                TX_i = x_i + v_i
            }
            if (f(TX_i) < f(x_i))
```

$x_i = TX_i$
}
if (f(x_i) < f(p_i))
$p_i = x_i$
if (f(x_i) < f(p_g))
$p_g = x_i$
}
}

3 Procedura eksperymentu

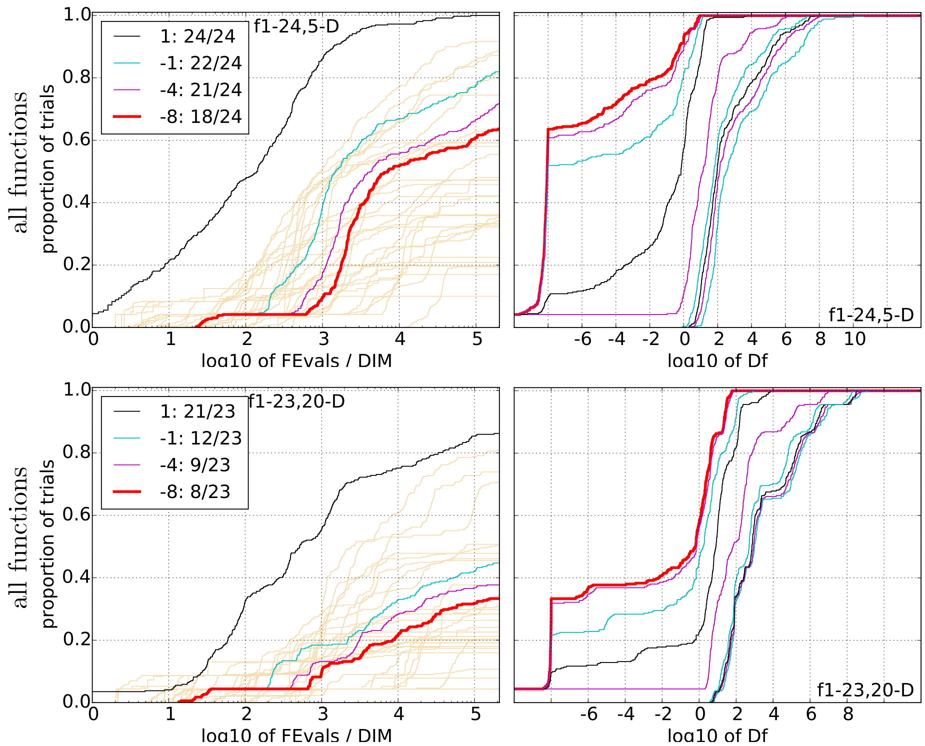
experimental procedure

4 CPU Timing

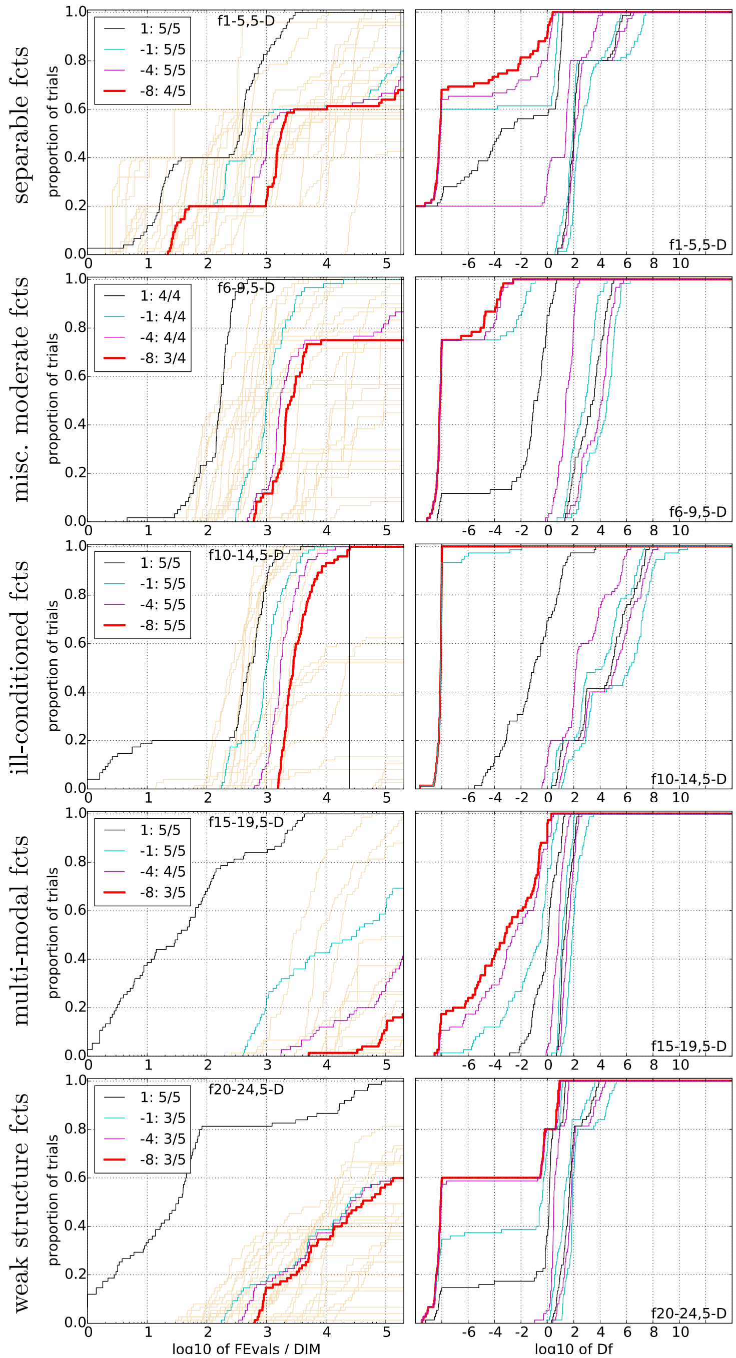
Algorytm był uruchamiany na komputerze z systemem Windows 8 Intel(R) Core(TM) i7-4500U CPU @ 2.39GHz. Czasy ewaluacji funkcji o wymiarach 2, 3, 5, 10, 20 wynosiły odpowiednio $1,9e^{-10}$, $2,2e^{-10}$, $2,4e^{-10}$, $3,5e^{-10}$ and $6,1e^{-10}$ sekund.

5 Uzyskane wyniki

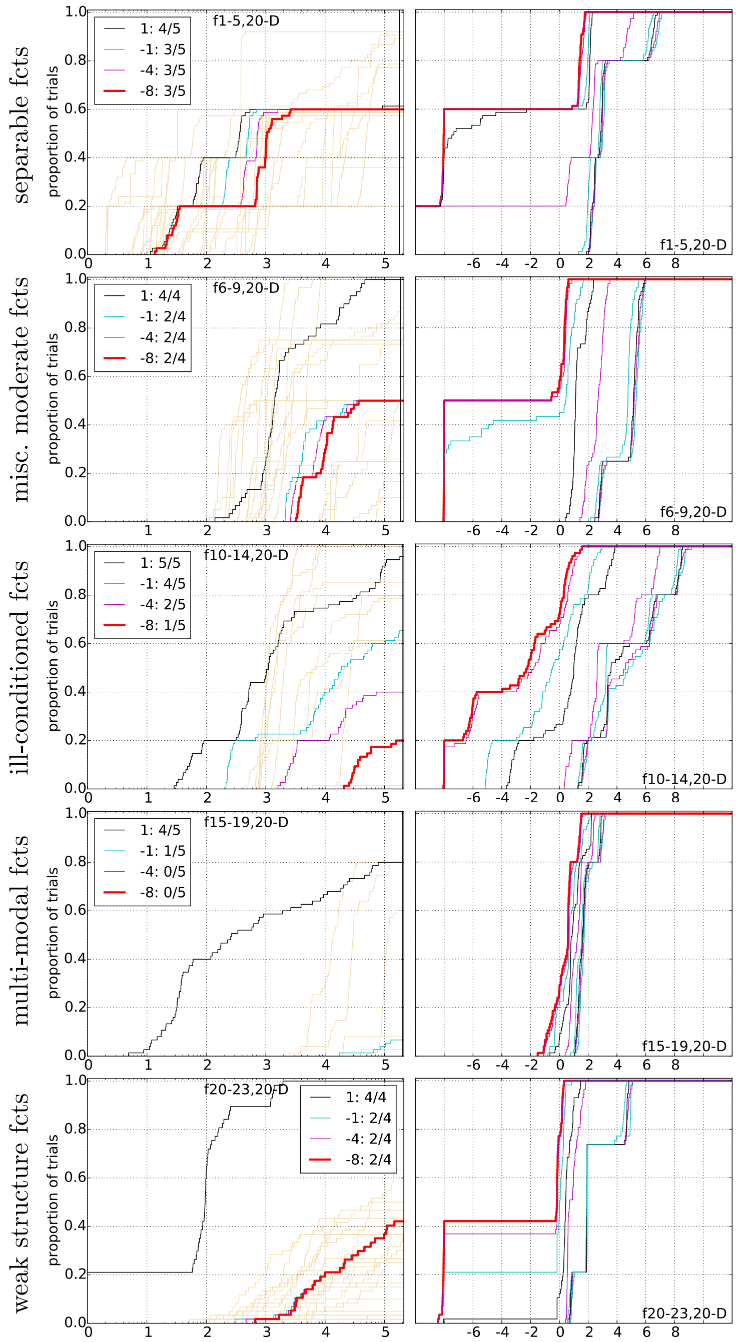
Results from experiments according to [**?**] on the benchmark functions given in [**?**, **?**] are presented in Figures 1, 2, 3, 4, 5, and 6 and Tables 1 and 2. The **expected running time (**ERT**)**, used in the figures and table, depends on a given target function value, $f_t = f_{opt} + \Delta f$, 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 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 if available.



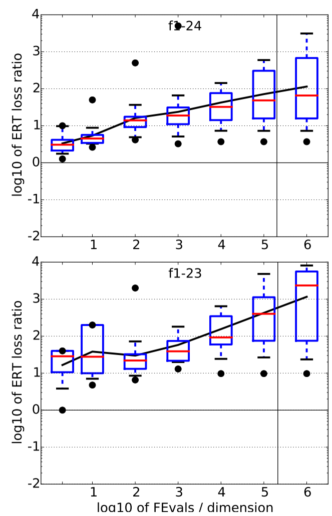
Rysunek 1: Empirical cumulative distribution functions (ECDF), plotting the fraction of trials with an outcome not larger than the respective value on the x-axis. Left subplots: ECDF of the number of function evaluations (FEvals) divided by search space dimension D, to fall below $f_{\rm opt} + \Delta f$ with $\Delta f = 10^k$, where k is the first value in the legend. The thick red line represents the most difficult target value $f_{\rm opt} + 10^{-8}$. Legends indicate for each target the number of functions that were solved in at least one trial within the displayed budget.Right subplots: ECDF of the best achieved Δf for running times of $0.5D, 1.2D, 3D, 10D, 100D, 1000D, \ldots$ function evaluations (from right to left cycling cyan-magenta-black...) and final Δf-value (red), where Δf and Df denote the difference to the optimal function value. Light brown lines in the background show ECDFs for the most difficult target of all algorithms benchmarked during BBOB-2009. The top row shows results for 5-D and the bottom row for 20-D.



Rysunek 2: Subgroups of functions 5-D. See caption of Figure 1 for details.



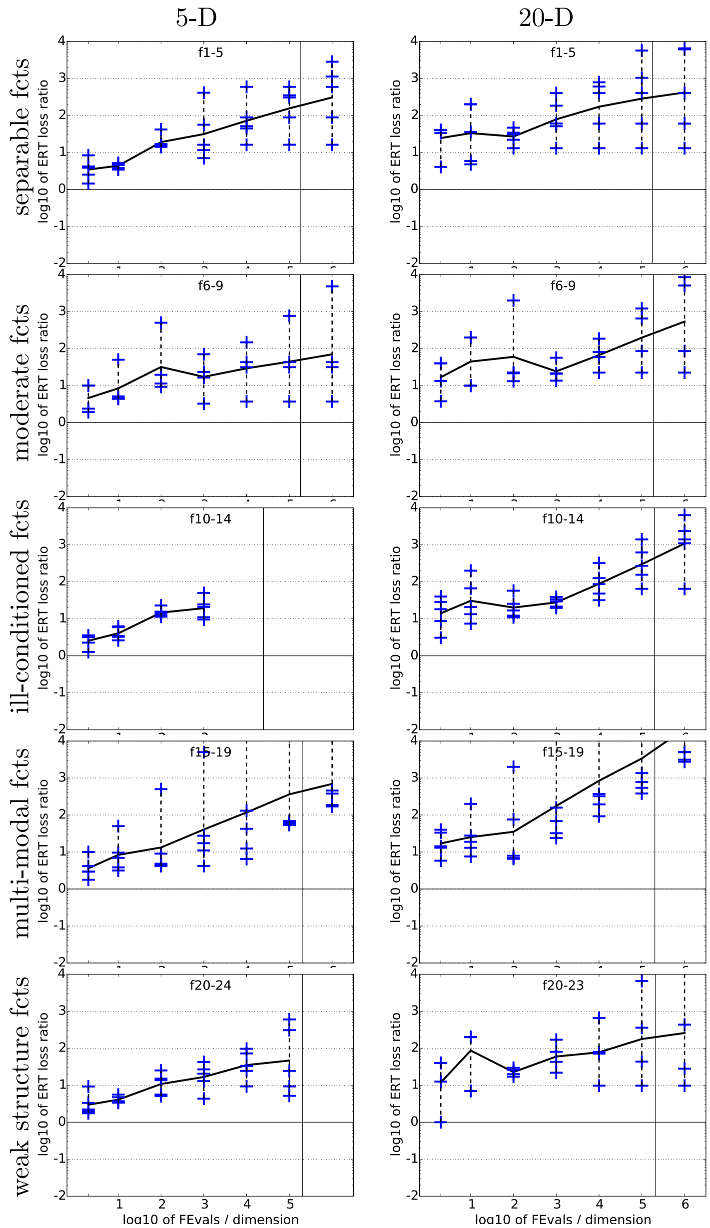
Rysunek 3: Subgroups of functions 20-D. See caption of Figure 1 for details.



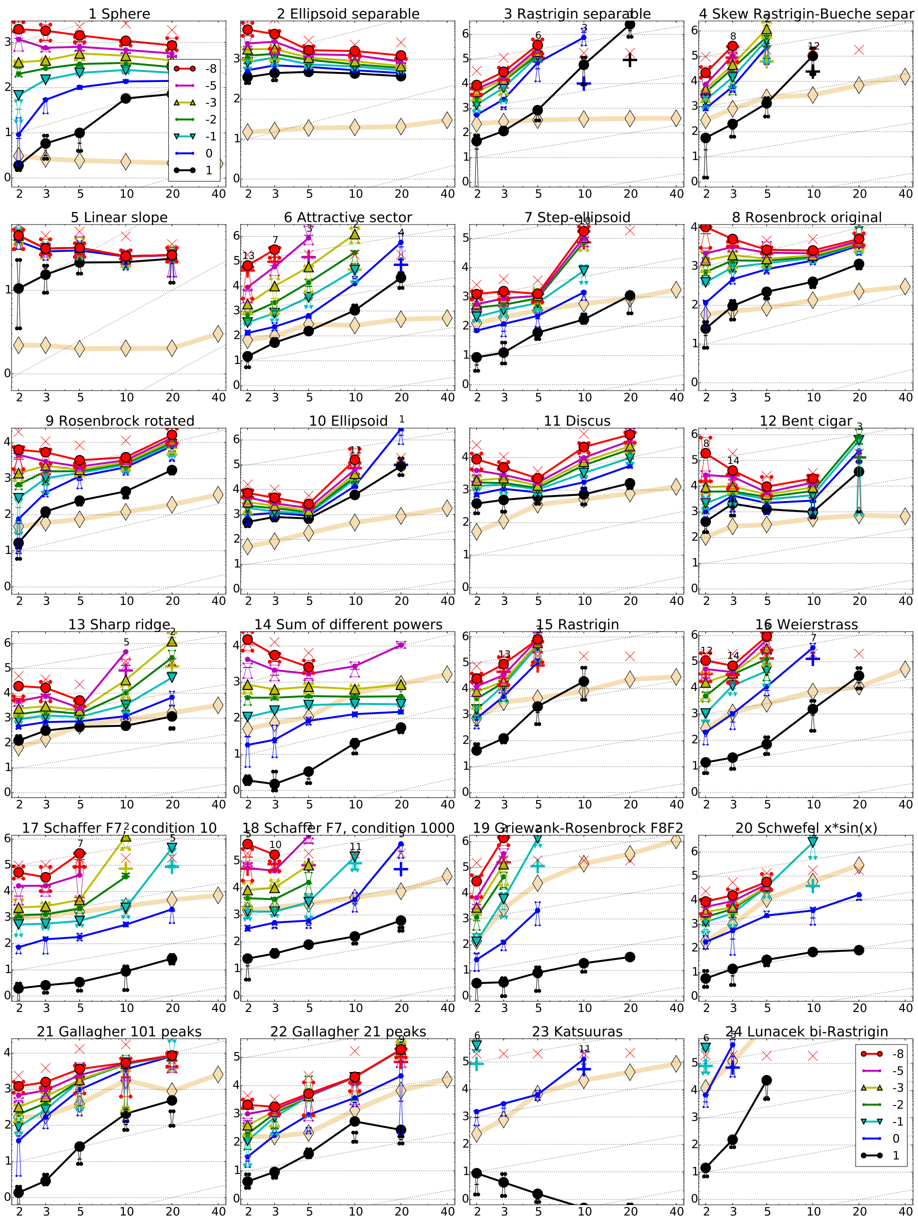
	f_1–f_{24} in 5-D, maxFE/D=201693					
#FEs/D	best	10%	25%	**med**	75%	90%
2	1.3	1.7	2.1	3.1	4.2	10
10	2.6	3.2	3.4	4.5	5.8	14
100	4.2	4.8	9.1	14	18	88
1e3	3.3	4.3	11	19	35	1.0e2
1e4	3.7	6.5	14	32	80	1.9e2
1e5	3.7	6.3	15	48	3.1e2	6.2e2
1e6	3.7	6.3	15	66	7.6e2	3.4e3
$\mathrm{RL_{US}}$/D	2e5	2e5	2e5	2e5	2e5	2e5

	f_1–f_{23} in 20-D, maxFE/D=211832					
#FEs/D	best	10%	25%	**med**	75%	90%
2	1.0	3.6	9.6	29	40	40
10	4.8	6.8	9.9	28	2.0e2	2.0e2
100	6.5	7.7	13	22	33	4.6e2
1e3	13	18	22	39	77	2.3e2
1e4	9.7	21	59	92	3.6e2	6.8e2
1e5	9.7	21	70	4.0e2	1.2e3	5.8e3
1e6	9.7	21	70	2.3e3	5.8e3	1.4e4
$\mathrm{RL_{US}}$/D	2e5	2e5	2e5	2e5	2e5	2e5

Rysunek 4: ERT loss ratio. Left: plotted versus given budget FEvals = #FEs in log-log display. Box-Whisker plot shows 25-75%-ile (box) with median, 10-90%-ile (caps), and minimum and maximum ERT loss ratio (points). The black line is the geometric mean. The vertical line gives the maximal number of function evaluations. Right: tabulated ERT loss ratios in 5-D (top table) and 20-D (bottom table). maxFE/D gives the maximum number of function evaluations divided by the dimension. $\mathrm{RL_{US}}$/D gives the median number of function evaluations for unsuccessful trials.



Rysunek 5: ERT loss ratio versus given budget FEvals divided by dimension in log-log display. Crosses give the single values on the indicated functions, the line is the geometric mean. The vertical line gives the maximal number of function evaluations in the respective function subgroup.



Rysunek 6: Expected number of f-evaluations (ERT, lines) to reach $f_{\mathrm{opt}} + \Delta f$; median number of f-evaluations (+) to reach the most difficult target that was reached not always but at least once; maximum number of f-evaluations in any trial (×); interquartile range with median (notched boxes) of simulated runlengths to reach $f_{\mathrm{opt}} + \Delta f$; all values are divided by dimension and plotted as $\log_{10}$ values versus dimension. Shown are $\Delta f = 10^{\{1,0,-1,-2,-3,-5,-8\}}$. Numbers above ERT-symbols (if appearing) indicate the number of trials reaching the respective target. The light thick line with diamonds indicates the respective best result from BBOB-2009 for $\Delta f = 10^{-8}$. Horizontal lines mean linear scaling, slanted grid lines depict quadratic scaling.

Δf	1e+1	1e+0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
$\mathbf{f_1}$	11	12	12	12	12	12	12	15/15
	4.5(4)	42(8)	87(17)	137(78)	235(59)	330(164)	538(217)	15/15
$\mathbf{f_2}$	83	87	88	89	90	92	94	15/15
	29(4)	37(15)	42(19)	54(43)	61(10)	72(41)	83(11)	15/15
$\mathbf{f_3}$	716	1622	1637	1642	1646	1650	1654	15/15
	5.8(2)	219(189)	405(213)	452(376)	604(826)	782(982)	900(864)	6/15
$\mathbf{f_4}$	809	1633	1688	1758	1817	1886	1903	15/15
	8.3(6)	712(986)	1297(1780)	1645(1871)	3361(3709)	∞	∞ *8.9e5*	0/15
$\mathbf{f_5}$	10	10	10	10	10	10	10	15/15
	11(3)	15(4)	16(4)	16(6)	16(7)	16(5)	16(6)	15/15
$\mathbf{f_6}$	114	214	281	404	580	1038	1332	15/15
	6.8(2)	15(15)	63(119)	173(154)	507(764)	4163(3929)	∞ *9.0e5*	0/15
$\mathbf{f_7}$	24	324	1171	1451	1572	1572	1597	15/15
	13(6)	3.5(1.0)	2.7(2)	3.1(2)	3.5(2)	3.5(2)	3.9(4)	15/15
$\mathbf{f_8}$	73	273	336	372	391	410	422	15/15
	15(6)	15(11)	18(8)	19(7)	20(11)	28(12)	30(16)	15/15
$\mathbf{f_9}$	35	127	214	263	300	335	369	15/15
	36(23)	46(42)	34(16)	30(21)	29(25)	32(12)	38(13)	15/15
$\mathbf{f_{10}}$	349	500	574	607	626	829	880	15/15
	10(3)	9.0(2)	9.5(1)	10(2)	11(2)	11(5)	14(5)	15/15
$\mathbf{f_{11}}$	143	202	763	977	1177	1467	1673	15/15
	22(19)	21(10)	6.8(3)	6.0(2)	5.8(2)	5.7(2)	6.4(2)	15/15
$\mathbf{f_{12}}$	108	268	371	413	461	1303	1494	15/15
	59(9)	41(14)	42(25)	46(18)	51(31)	22(14)	31(28)	15/15
$\mathbf{f_{13}}$	132	195	250	319	1310	1752	2255	15/15
	17(8)	19(6)	22(9)	23(9)	7.7(3)	7.4(3)	11(2)	15/15
$\mathbf{f_{14}}$	10	41	58	90	139	251	476	15/15
	1.7(2)	10(5)	20(6)	23(10)	27(4)	34(29)	25(19)	15/15
$\mathbf{f_{15}}$	511	9310	19369	19743	20073	20769	21359	14/15
	20(20)	57(34)	118(197)	150(249)	147(192)	194(287)	189(193)	3/15
$\mathbf{f_{16}}$	120	612	2662	10163	10449	11644	12095	15/15
	2.9(5)	90(95)	82(80)	62(156)	154(150)	291(231)	385(332)	3/15
$\mathbf{f_{17}}$	5.2	215	899	2861	3669	6351	7934	15/15
	3.3(3)	4.2(2)	4.1(2)	3.8(2)	6.4(3)	33(33)	127(133)	7/15
$\mathbf{f_{18}}$	103	378	3968	8451	9280	10905	12469	15/15
	3.8(3)	8.2(4)	3.7(3)	10(5)	37(51)	382(427)	∞ *9.3e5*	0/15
$\mathbf{f_{19}}$	1	1	242	1.0e5	1.2e5	1.2e5	1.2e5	15/15
	40(47)	10659(5637)	27837(49148)	∞	∞	∞	∞ *9.5e5*	0/15
$\mathbf{f_{20}}$	16	851	38111	51362	54470	54861	55313	14/15
	10(7)	13(6)	4.7(3)	3.7(3)	3.5(2)	4.1(4)	4.5(4)	15/15
$\mathbf{f_{21}}$	41	1157	1674	1692	1705	1729	1757	14/15
	3.2(3)	4.2(5)	5.0(7)	5.2(5)	5.3(7)	7.1(5)	8.5(12)	15/15
$\mathbf{f_{22}}$	71	386	938	980	1008	1040	1068	14/15
	2.8(2)	12(27)	22(11)	22(54)	21(16)	22(31)	24(50)	15/15
$\mathbf{f_{23}}$	3.0	518	14249	27890	31654	33030	34256	15/15
	2.6(2)	63(55)	∞	∞	∞	∞	∞ *1.0e6*	0/15
$\mathbf{f_{24}}$	1622	2.2e5	6.4e6	9.6e6	9.6e6	1.3e7	1.3e7	3/15
	73(83)	∞	∞	∞	∞	∞	∞ *9.5e5*	0/15

Tabela 1: Expected running time (ERT in number of function evaluations) divided by the best ERT measured during BBOB-2009. The ERT and in braces, as dispersion measure, the half difference between 90 and 10%-tile of bootstrapped run lengths appear in the second row of each cell, the best ERT in the first. The different target Δf-values are shown in the top row. #succ is the number of trials that reached the (final) target $f_{\mathrm{opt}} + 10^{-8}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. **Bold** entries are statistically significantly better (according to the rank-sum test) compared to the best algorithm in BBOB-2009, with $p = 0.05$ or $p = 10^{-k}$ when the number $k > 1$ is following the ↓ symbol, with Bonferroni correction by the number of functions.
Results of PSO DE withResets in 5-D.

Δf	1e+1	1e+0	1e-1	1e-2	1e-3	1e-5	1e-7	#succ
$\mathbf{f_1}$	43	43	43	43	43	43	43	15/15
	34(5)	66(7)	100(12)	133(15)	187(8)	267(154)	368(207)	15/15
$\mathbf{f_2}$	385	386	387	388	390	391	393	15/15
	20(3)	23(5)	27(4)	31(3)	34(3)	44(2)	54(23)	15/15
$\mathbf{f_3}$	5066	7626	7635	7637	7643	7646	7651	15/15
	10080(7120)	∞	∞	∞	∞	∞	∞ *3.5e6*	0/15
$\mathbf{f_4}$	4722	7628	7666	7686	7700	7758	1.4e5	9/15
	∞	∞	∞	∞	∞	∞	∞ *3.6e6*	0/15
$\mathbf{f_5}$	41	41	41	41	41	41	41	15/15
	12(3)	13(4)	13(4)	13(3)	13(5)	13(4)	13(5)	15/15
$\mathbf{f_6}$	1296	2343	3413	4255	5220	6728	8409	15/15
	334(258)	4934(8328)	∞	∞	∞	∞	∞ *3.7e6*	0/15
$\mathbf{f_7}$	1351	4274	9503	16523	16524	16524	16969	15/15
	17(30)	∞	∞	∞	∞	∞	∞ *3.7e6*	0/15
$\mathbf{f_8}$	2039	3871	4040	4148	4219	4371	4484	15/15
	11(3)	18(13)	18(12)	19(16)	19(18)	21(6)	22(12)	15/15
$\mathbf{f_9}$	1716	3102	3277	3379	3455	3594	3727	15/15
	20(9)	51(67)	55(42)	59(77)	64(49)	74(38)	82(58)	15/15
$\mathbf{f_{10}}$	7413	8661	10735	13641	14920	17073	17476	15/15
	240(85)	6468(8465)	∞	∞	∞	∞	∞ *3.9e6*	0/15
$\mathbf{f_{11}}$	1002	2228	6278	8586	9762	12285	14831	15/15
	32(20)	50(30)	31(23)	42(21)	46(30)	53(38)	57(9)	15/15
$\mathbf{f_{12}}$	1042	1938	2740	3156	4140	12407	13827	15/15
	704(1774)	2442(6148)	4350(3059)	5462(3035)	∞	∞	∞ *3.7e6*	0/15
$\mathbf{f_{13}}$	652	2021	2751	3507	18749	24455	30201	15/15
	36(67)	67(82)	310(487)	1537(1450)	1315(1778)	∞	∞ *3.4e6*	0/15
$\mathbf{f_{14}}$	75	239	304	451	932	1648	15661	15/15
	15(5)	13(3)	16(1)	18(2)	18(3)	129(46)	∞ *4.0e6*	0/15
$\mathbf{f_{15}}$	30378	1.5e5	3.1e5	3.2e5	3.2e5	4.5e5	4.6e5	15/15
	∞	∞	∞	∞	∞	∞	∞ *3.6e6*	0/15
$\mathbf{f_{16}}$	1384	27265	77015	1.4e5	1.9e5	2.0e5	2.2e5	15/15
	417(483)	∞	∞	∞	∞	∞	∞ *4.0e6*	0/15
$\mathbf{f_{17}}$	63	1030	4005	12242	30677	56288	80472	15/15
	8.6(5)	40(20)	2268(1714)	∞	∞	∞	∞ *3.7e6*	0/15
$\mathbf{f_{18}}$	621	3972	19561	28555	67569	1.3e5	1.5e5	15/15
	20(26)	2171(1607)	∞	∞	∞	∞	∞ *3.8e6*	0/15
$\mathbf{f_{19}}$	1	1	3.4e5	4.7e6	6.2e6	6.7e6	6.7e6	15/15
	661(293)	∞	∞	∞	∞	∞	∞ *3.7e6*	0/15
$\mathbf{f_{20}}$	82	46150	3.1e6	5.5e6	5.5e6	5.6e6	5.6e6	14/15
	20(4)	7.4(6)	∞	∞	∞	∞	∞ *3.5e6*	0/15
$\mathbf{f_{21}}$	561	6541	14103	14318	14643	15567	17589	15/15
	17(27)	25(39)	12(25)	12(16)	11(18)	11(12)	10(7)	15/15
$\mathbf{f_{22}}$	467	5580	23491	24163	24948	26847	1.3e5	12/15
	12(4)	79(66)	159(113)	155(131)	150(200)	140(110)	28(30)	9/15
$\mathbf{f_{23}}$	3.2	1614	67457	3.7e5	4.9e5	8.1e5	8.4e5	15/15
	1.8(0.9)	∞	∞	∞	∞	∞	∞ *4.2e6*	0/12

Tabela 2: Expected running time (ERT in number of function evaluations) divided by the best ERT measured during BBOB-2009. The ERT and in braces, as dispersion measure, the half difference between 90 and 10%-tile of bootstrapped run lengths appear in the second row of each cell, the best ERT in the first. The different target Δf-values are shown in the top row. #succ is the number of trials that reached the (final) target $f_{\mathrm{opt}} + 10^{-8}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. **Bold** entries are statistically significantly better (according to the rank-sum test) compared to the best algorithm in BBOB-2009, with $p = 0.05$ or $p = 10^{-k}$ when the number $k > 1$ is following the ↓ symbol, with Bonferroni correction by the number of functions.
Results of PSO DE withResets in 20-D.