Black-Box Optimization Benchmarking for Noiseless Function Testbed Using A Direction-Based RCGA

Yao-Chen Chuang
Department of Chemical Engineering
Feng Chia University
No. 100, Wenhwa Rd., Seatwen
Taichung, Taiwan 40724, R.O.C.
ycchuang@mail.fcu.edu.tw

Chyi-Tsong Chen
Department of Chemical Engineering
Feng Chia University
No. 100, Wenhwa Rd., Seatwen
Taichung, Taiwan 40724, R.O.C.
ctchen@fcu.edu.tw

ABSTRACT

This paper benchmarks a novel and efficient real-coded genetic algorithm (RCGA) enhanced from our previous work [1] on the noisefree BBOB 2012 testbed. The enhanced algorithm termed as direction-based RCGA (DBRCGA) uses relative fitness information to direct the crossover toward a direction that significantly improves the objective fitness. As a base of performance evaluation and comparisons, the maximum number of function evaluations (#FEs) for each test run is set to 10^5 times to the problem dimension. Extensive benchmarking test results reveal that all functions can be solved by DBRCGA in the low search dimensions. Although the DBRCGA shows the difficulty in getting a solution with the desired accuracy 10^{-8} for high conditioning and multimodal functions within the specified maximum #FEs, the DBRCGA presents good performance in separable function and functions with low or moderate conditioning.

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

Keywords

Benchmarking, Black-box optimization, Evolutionary algorithms, Real-coded genetic algorithm

1. INTRODUCTION

Owing to its broad applicability and powerful performance, numerical optimization evolutionary algorithms (EAs) have been widely applied to solve real-word optimization problems arose from the fields of science, economic and engineering. Many different types of EAs have been proposed over the past few decades. According to the mechanism used, they can be classified into the following categories: genetic algorithms (GAs), evolutionary strategies (ESs), genetic programming (GP), evolutionary programming (EP) and differential evolution (DE). Among the developed EAs, the real-coded genetic algorithms (RCGAs) appear to be one of the most popular optimization methods due to ease of implementation and global perspective.

RCGAs consists of three main functional operators: selection, crossover and mutation. The selection operation selects those solutions that have better fitness among the population for mating process. The crossover operation generates new candidates by using the genetic information of selected chromosomes. To prevent premature convergence to a suboptimum, the mutation operation randomly changes the gene of chromosomes. In the RCGA framework, these three main operations are operated iteratively, making the population achieve better and better fitness and arrive at the optimal solution. In regard to the evolutionary mechanisms of RCGAs, the crossover operation has been considered as the most important scheme to produce candidate solutions. In our previous work [1], we introduced a direction-based crossover (DBX) operator for RCGAs. In this paper we enhanced the DBX and developed an efficient RCGA termed as DBRCGA for the solution of single-objective optimization problems. The DBRCGA uses relative fitness information derived from the objective function to guide the crossover toward a direction that significantly improves the fitness. The direction vector formed from each paired parents can carry out 2^n-1 possible crossover scenarios to effectively generate offsprings. Moreover, the crossover step size, which is controlled by the ratio of fitness distance of the paired parents and the maximum fitness difference among the current population, can be adjusted automatically to conduct the crossover during the evolution. Therefore, the proposed DBRCGA has a great ability to successfully direct the generation of offspring chromosomes that have much better fitness than their parents.

2. ALGORITHM PRESENTATION

The proposed DBRCGA integrates three specific functional operators - ranking selection (RS), direction-based crossover (DBX) and dynamic random mutation (DRM). The configuration of the proposed DBRCGA is shown in Figure 1. Before introducing these main operational functions, the design concept and notations are presented below. Let $\mathbf{x}^*$ be the global optimum solution that minimizes the objective function $f(\mathbf{x})$ in the admissible search space $\mathbf{\Omega}$, i.e., $\forall \mathbf{x} \in \mathbf{\Omega} : f(\mathbf{x}^*) \leq f(\mathbf{x})$ and $f(\mathbf{x}) : \mathbf{\Omega} \subseteq \Re^n \rightarrow \Re$. To solve

GECCO'12 Companion, July 7–11, 2012, Philadelphia, PA, USA.

this problem using DBRCGA, we let $\boldsymbol{\theta} = [x_1, x_2, \ldots, x_n]$ be a solution termed as chromosome/individual in the sense of RCGAs. In the chromosome, each x_j is called a gene and represented as real number, where $j \in \bar{n}$ and $\bar{n} = \{1, 2, \ldots, n\}$. The admissible parameter space for $\boldsymbol{\theta}$ is defined as follows

$$\boldsymbol{\Omega_\theta} = \{ \boldsymbol{\theta} \in \Re^n | x_{1,\min} \le x_1 \le x_{1,\max}, x_{2,\min} \le x_2 \le x_{2,\max}, \cdots, x_{n,\min} \le x_n \le x_{n,\max} \} \tag{1}$$

All the genes x_j are confined in the region of $\boldsymbol{\Omega_\theta}$ between the lower bound $\boldsymbol{\theta}^L \equiv [x_{1,\min},\ x_{2,\min}, \ldots,\ x_{n,\min}]$ and the upper bound $\boldsymbol{\theta}^U \equiv [x_{1,\max},\ x_{2,\max}, \ldots,\ x_{n,\max}]$. Furthermore, let N be the number of the chromosomes in the population, and λ represents the threshold parameter for the control of crossover probability and mutation probability in the population. Having completed the description of those required notations and basic concept, we are ready to introduce the operational procedure and operators associated with the DBRCGA optimization scheme. .

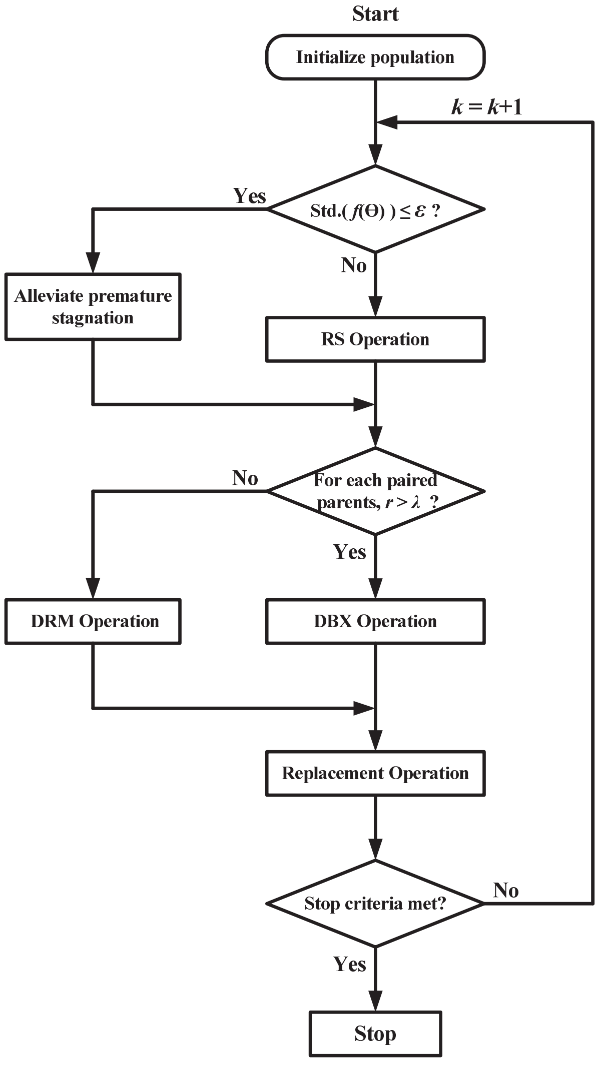

Figure 1: The flow chart of the proposed DBRCGA optimization scheme.

2.1 Ranking Selection

Before the ranking section operator (RS) is applied, we first rank the population according to their fitness. By using RS, the amount of $\lfloor pN \rfloor$ chromosomes that have relatively worse fitness function values are discarded, while at the same time reproducing$\lfloor pN \rfloor$ chromosomes with better fitness. Note that the parameter p is a user-defined proportional parameter in the interval [0, 1]. Since the RS is operated on the base of fitness ranking, it is very simple and fast to compute, and at the same time keeps the total number of chromosomes in the population constant. Most importantly, the RS operator ensures to preserve those potential solutions.

2.2 Direction-Based Crossover

Crossover operation, which blends parents' information to generate the offspring chromosomes, is the most important search function in RCGAs. With a special feature of using fitness information of paired parents, an efficient crossover operator, named direction-based crossover (DBX) operator, is developed in this paper. Using DBX, the chromosomes in the population are firstly sorted according to the fitness ranking. Let the sorted fitness ranking be denoted by $\tilde{\boldsymbol{\Theta}} = \{\theta_1, \theta_2, \ldots, \theta_N\}$, where the chromosomes satisfy the relation of $\theta_1 \preceq \theta_2 \preceq \ldots \preceq \theta_N$. Note that the notation $A \preceq B$ means that the fitness of A is better or equal to that of B. According to $\tilde{\boldsymbol{\Theta}}$, we divide the N chromosomes into two groups as follows:
GroupA (the leading group):

$$\begin{aligned} \boldsymbol{\Theta}^A &= \{\theta_1, \theta_2, \ldots \theta_{N/2}\} \\ &\equiv \{\theta_1^A, \theta_2^A, \ldots \theta_M^A\} \end{aligned} \tag{2.a}$$

GroupB:

$$\begin{aligned} \boldsymbol{\Theta}^B &= \{\theta_{N/2+1}, \theta_{N/2+2}, \ldots \theta_N\} \\ &\equiv \{\theta_1^B, \theta_2^B, \ldots \theta_M^B\} \end{aligned} \tag{2.a}$$

Note that *groupA* has better fitness than that of B. By using the DBX operator, we first chose $\boldsymbol{\theta}_i^A$ from *groupA* and $\boldsymbol{\theta}_i^B$ from *groupB* to make $N/2$ pairs for crossover. Let r be a random number ($0 \leqslant r \leqslant 1$), the DBX operation is performed to generate two offspring chromosomes by the following rule

$$\boldsymbol{\theta}_i^{A*} \leftarrow \boldsymbol{\theta}_i^A + s_{c,i}\overrightarrow{\mathbf{D}}_i \tag{3.a}$$

$$\boldsymbol{\theta}_i^{B*} \leftarrow \boldsymbol{\theta}_i^B + s_{c,i}\overrightarrow{\mathbf{D}}_i \tag{3.b}$$

when r_i is greater than the specified crossover probability threshold λ, i.e., $r_i > \lambda$. In the above, $s_{c,i}$ denotes the step size control of the crossover, which is given by

$$s_{c,i} = \frac{\left| f\left(\theta_i^A\right) - f\left(\theta_i^B\right) \right|}{\max.\{f(\boldsymbol{\Theta})\} - \min.\{f(\boldsymbol{\Theta})\}} \tag{4}$$

and $\overrightarrow{\mathbf{D}}_i$, the crossover direction, is determined by

$$D_{i,j} = \left\{ \begin{array}{ll} 0 & : \ r_j < 0.5 \\ \theta_{i,j}^A - \theta_{i,j}^B & : \ r_j \ge 0.5 \end{array} \right. , \quad j = 1, 2, \ldots, n. \tag{5}$$

From Eq. (4), it can be observed that the value of s_c is in the range of $0 \leqslant s_c \leqslant 1$ and is determined by the ratio made by the fitness distance between the chosen paired-parents and the maximum fitness difference among the current population. Eq. (5) shows that each paired genes has the same probability to generate crossover direction for the DBX operation. Because at least one paired genes will be operated to produce a crossover direction, the direction vector $\overrightarrow{\mathbf{D}}_i$ can't be null, and that makes 2^n-1 possible search directions for the generation of offspring chromosomes. To increase the evolution efficiency and provide the diversity, a dynamic mutation operator (DRM) (to be introduced later) is used to replace the DBX operation when the chromosomes of paired parents have "the same properties". By same properties of two chromosomes, we mean that they have identical genes and/or fitness, i.e. $\boldsymbol{\theta}_i^A$=$\boldsymbol{\theta}_i^B$ and/or $f(\boldsymbol{\theta}_i^A)$=$f(\boldsymbol{\theta}_i^B)$. The replacement of DBX with DRM when two parents have same properties constitutes a parallel loop for the proposed DBRCGA. In practice, the DBX operator might guide the genes of offspring chromosome to lie outside the variable bounds since the search is made along the suggested direction contributed by the paired parents. This situation implies that a better solution might exist outside the admissible search space $\mathbf{\Omega}_{\boldsymbol{\theta}}$. However, due to the physical limitations of design variables, those genes outside $\mathbf{\Omega}_{\boldsymbol{\theta}}$ are adjusted to fix at their individual bounds to ensure the satisfaction of the required lower and upper bounds.

2.3 Dynamic Random Mutation

To provide the population diversity, we suggest using the following dynamic random mutation (DRM) operator:

$$\boldsymbol{\theta}_i^{A*} \leftarrow \boldsymbol{\theta}_i^A + s_m \mathbf{\Phi}_0 \left(\boldsymbol{\theta}^U - \boldsymbol{\theta}^L\right) \tag{6.a}$$

$$\boldsymbol{\theta}_i^{B*} \leftarrow \boldsymbol{\theta}_i^B + s_m \mathbf{\Phi}_0 \left(\boldsymbol{\theta}^U - \boldsymbol{\theta}^L\right) \tag{6.b}$$

In the above, $\mathbf{\Phi}_0$ is a random perturbation vector in the n-dimensional cube $[-\phi_0, +\phi_0]^n$ where ϕ_0 is an user-defined number in the interval (0, 1]. In Eq. (6), the mutation step size s_m is dynamically tuned by

$$s_m = (1 - k/k_{\max})^b \tag{7}$$

where the parameter b (> 0) is used to control the decay rate of s_m and, k_{max} denotes a maximum generation number of evolution. Note that the decay rate b governs the shape of the allowable mutation region and the allowable mutation range dynamically decreases as the number of generation k increases. The idea behind the use of a dynamic mutation size is similar to that of simulated annealing. With the proposed DRM operator, a large step size for mutation is used at the beginning of evolution to provide a great chance of variations in population, and as a consequence the searching can avoid being trapped by a local optimum. As population gradually converge to an optimum solution, a small feasible mutation region produced by the DRM operator can then enhance the precision of the obtained solution. Note that the DRM is operated if $r_i \leq \lambda$ or the paired parents have same properties. Similarly to the DRM operation, those mutated genes outside $\mathbf{\Omega}_{\boldsymbol{\theta}}$ will also be individually fixed at their bounds.

2.4 Replacement Operator

Since the population size is kept constant, the survivor selection of both parent and offspring population is an important way to preserve the currently found best solution for use in subsequent generations. Two replacement strategies, named generational and steady-state replacement, are commonly used in associated with RCGAs. Generational replacement systematically replaces the parent population with the offspring population, whereas steady-state replacement only replaces one parent chromosome with the best offspring chromosome. In the proposed DBRCGA, we utilize the generational replacement with elitism strategy by which at each generation the offspring chromosome is individually compared with its parent chromosome ($\boldsymbol{\theta}_i^{A*}$ compare with $\boldsymbol{\theta}_i^A$ and $\boldsymbol{\theta}_i^{B*}$ compare with $\boldsymbol{\theta}_i^B$) to retain better ones for next generation.

2.5 Alleviate Stagnation

It is well known that premature convergence and stagnation can happen when the convergence speed is too fast and/or the solution has been trapped by a local optimum. To alleviate premature stagnation and make the DBRCGA algorithm more efficient to find the global optimum solution, we examine the population diversity by calculating the standard deviation of chromosomes' fitness. Once stagnation has occured, i.e., Std. Dev.$(f(\mathbf{\Theta})) \leq \varepsilon$, where ε is a very small positive value, we refresh all chromosomes except the current best one are refreshed by using a simple random procedure.

2.6 Parameters Tuning Guideline

The proposed DBRCGA consists of four parameters, λ, p, ϕ_0 and b. The parameter λ controls the crossover and mutation probability of the population. As mentioned previously, the DBX operation is the most important function to generate candidate solutions. To ensure that paired parents should are to be performed by the DBX operation, we recommand to use the value of λ=0.1. Besides, the proportional parameter p used in RS operation strongly affects the search directions of DBX operation. Thus, a small value of p, say $0 \leqslant p \leqslant 0.1$, is suggested to maintain the population diversity. The parameter ϕ_0, which is used to control the dynamic variation of the DRM operator, is generally suggested as ϕ_0=0.5 to ensure the condition that at least half of search space of each gene can be covered by DRM operation at the beginning of evolution, making the chromosome have great chances to jump out from the suboptimum region. In addition, the value of b is experimentally recommended to be in the range of $2 \leqslant b \leqslant 4$ to preserve the solution efficiency. A very small positive value for the convergence level parameter ε should be used to alleviate stagnation. Generally, a value between 10^{-12} and 10^{-10} is used for the case that the desired precision of solution is set below 10^{-8}.

3. EXPERIMENTAL PROCEDURE

According to [3], the experimental procedure has been carried out on the benchmark functions given in [2, 4]. An independent restart strategy was implemented in the proposed DBRCGA. For each restart, the initial population $\mathbf{\Theta}$ is uniformly and randomly sampled within the search space $[-5, 5]^n$. Whenever the restart condition is met, the algorithm will be reinitialized without using any information about the last test run. This process is iterated until the stopping criteria are met, i.e., maximum number of function evalutions ($10^5 * n$) has been reached, or the function value is less than the target precision (10^{-8}). The restart condition used in the DBRCGA is that the best solution obtained so far does not vary more than 10^{-12} during the last $(50 + 25 * n)$ generations, indicating that there is no significant improvement in the population and the algorithm may have been stuck on suboptimal region.

4. PARAMETER SETTINGS

According to the tuning guideline mentioned above, the DBRCGA parameters are given as follows: $\lambda = 0.1$, $pN = 1$, $\phi_0 = 0.5$, $b = 4$ and $\varepsilon = 10^{-12}$. To take problem dimensions into account in DBRCGA, a dimension-dependent population size $N = min(10 * n, 100)$ is used. Because that the above mentioned parameter settings are identical for all benchmarking functions, i.e., no specific parameter tuning has been conducted for each function, the crafting effort [3] is CrE=0 in using DBRCGA.

5. RESULTS

Results of DBRCGA from experiments are presented in Figures 2, 3 and 4 and in Tables 1 and 2. Extensive benchmark results reveal that all functions can be solved by DBRCGA in the low search dimensions. Although some functions in high search dimensions is unable to find a solution satisfying the target precision 10^{-8}, the DBRCGA provides good performance in separable function and functions with low or moderate conditioning.

6. CPU TIMING EXPERIMENT

For the CPU timing experiment, the same DBRCGA algorithm was run on $f8$ until at least 30 seconds had passed. These experiments have been conducted with an Intel(R) Core(TM) i7 CPU 920 processor, running at 2.61 GHz, under Microsoft Windows 7 Enterprise x64 SP1 with 12GB RAM and Matlab 7.12 (R2011a). The time per function evaluation was 1.5; 1.3; 1.2; 1.0; 1.1; 1.2 times 10^{-5} seconds for DBRCGA in dimensions 2; 3; 5; 10; 20; 40 respectively.

7. REFERENCES

[1] Y.-C. Chuang and C.-T. Chen. A study on real-coded genetic algorithm for process optimization using ranking selection, direction-based crossover and dynamic mutation. In *2011 IEEE Congress on Evolutionary Computation (CEC)*, pages 2488–2495, New Orleans, LA, USA, 2011.

[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. Updated February 2010.

[3] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2012: Experimental setup. Technical report, INRIA, 2012.

[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. Updated February 2010.

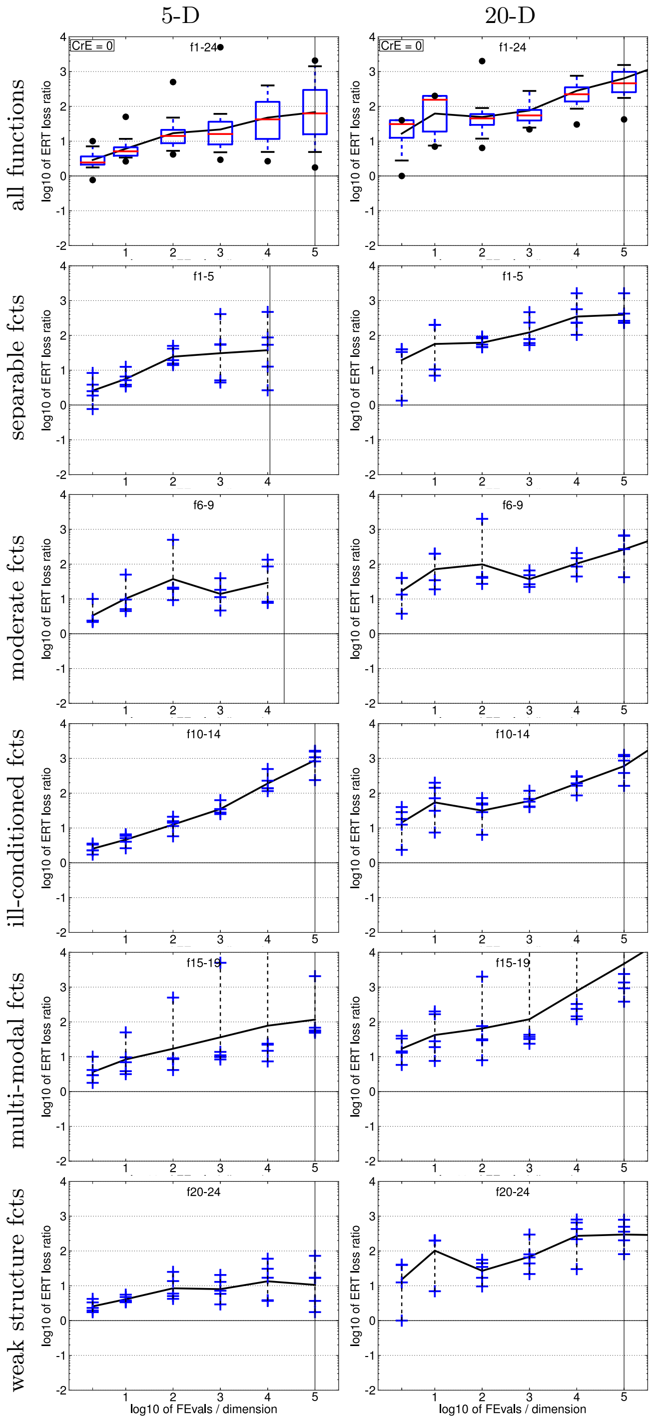

Figure 4: ERT loss ratio vs. a given budget FEvals**. Each cross (+) represents a single function. The target value f_t used for a given** FEvals **is the smallest (best) recorded function value such that** $\mathrm{ERT}(f_t) \leq$ FEvals **for the presented algorithm. Shown is** FEvals **divided by the respective best** $\mathrm{ERT}(f_t)$ **from BBOB-2009 for functions f_1–f_{24} in 5-D and 20-D. 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 a single trial in this function subset.**

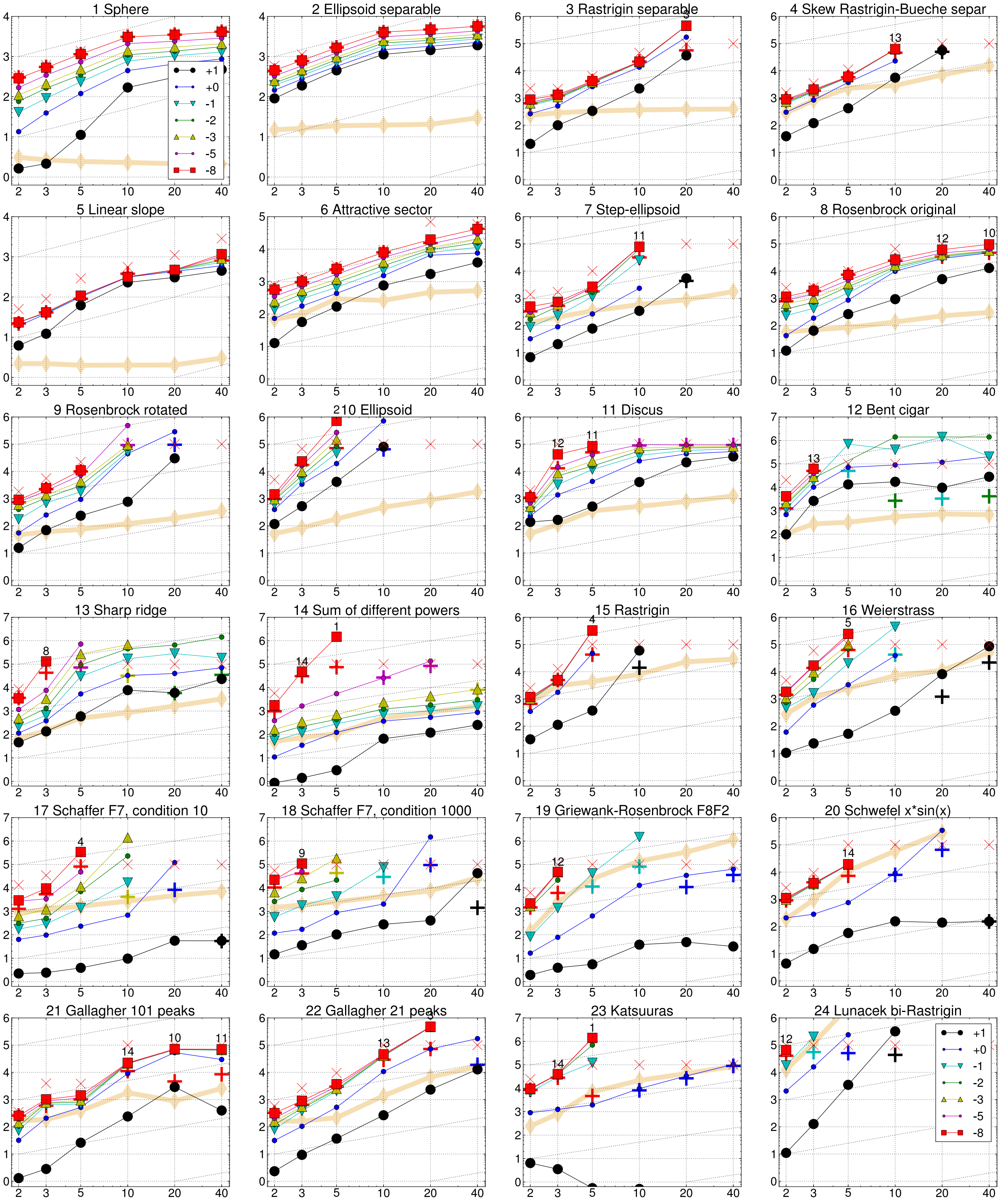

Figure 2: Expected number of f-evaluations (ERT, with lines, see legend) to reach $f_{\mathrm{opt}} + \Delta f$, median number of f-evaluations to reach the most difficult target that was reached at least once ($+$) and maximum number of f-evaluations in any trial ($\times$), all divided by dimension and plotted as $\log_{10}$ values versus dimension. Shown are $\Delta f = 10^{\{1,0,-1,-2,-3,-5,-8\}}$. Numbers above ERT-symbols indicate the number of successful trials. 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.

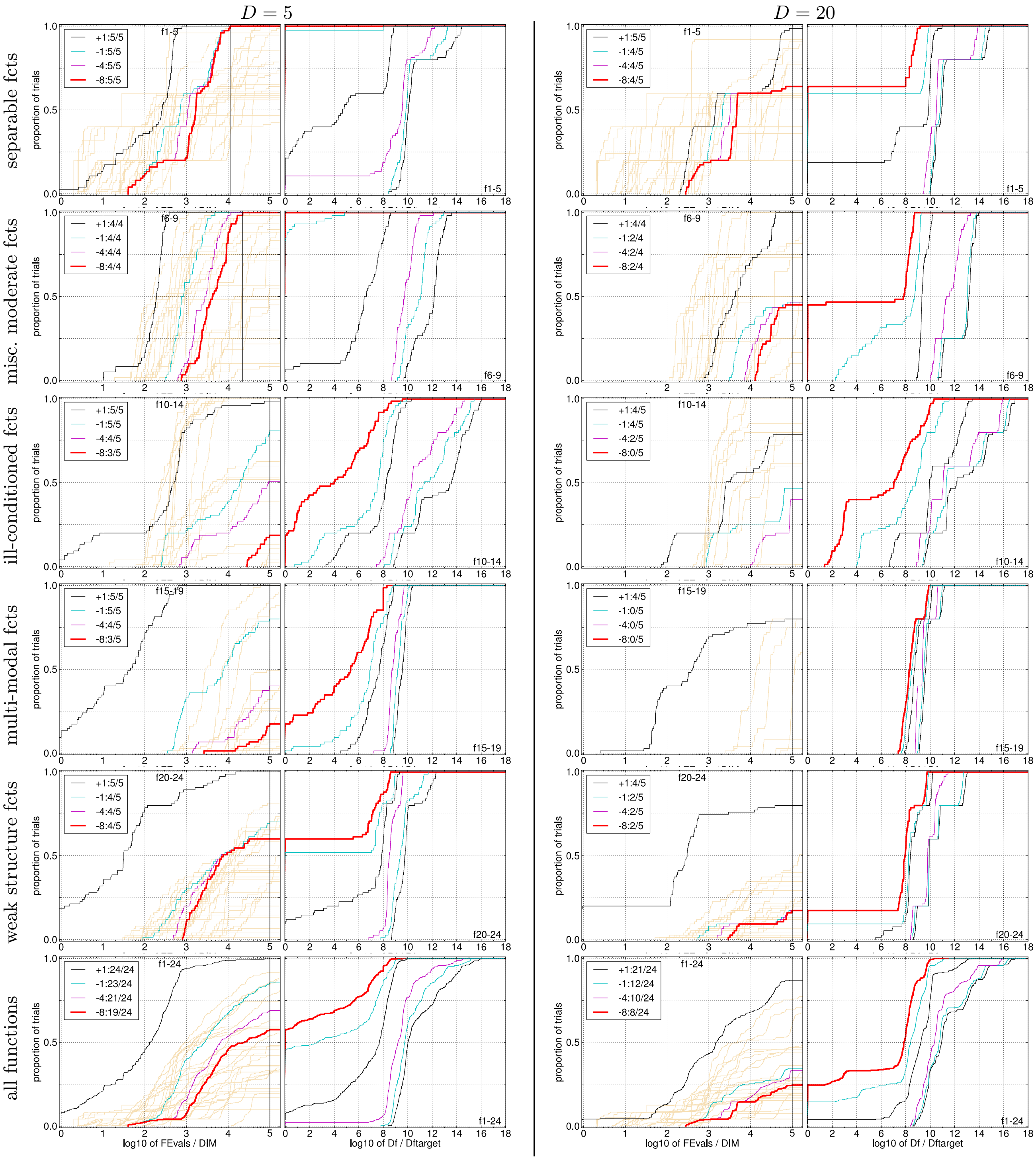

Figure 3: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials with an outcome not larger than the respective value on the x-axis. Left subplots: ECDF of number of function evaluations (FEvals) divided by search space dimension D, to fall below $f_{\rm opt} + \Delta f$ with $\Delta f = 10^k$, where k is the first value in the legend. Right subplots: ECDF of the 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). The thick red line represents the most difficult target value $f_{\rm opt} + 10^{-8}$. Legends indicate the number of functions that were solved in at least one trial. Light brown lines in the background show ECDFs for $\Delta f = 10^{-8}$ of all algorithms benchmarked during BBOB-2009.

5-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f_1	11	12	12	12	12	12	15/15
	5.1(7)	49(33)	96(26)	197(20)	301(33)	416(43)	15/15
f_2	83	87	88	90	92	94	15/15
	27(4)	33(4)	41(5)	54(6)	67(6)	81(6)	15/15
f_3	716	1622	1637	1646	1650	1654	15/15
	2.4(1.0)	8.0(4)	11(6)	11(6)	12(6)	12(6)	15/15
f_4	809	1633	1688	1817	1886	1903	15/15
	2.6(1)	11(7)	17(10)	16(10)	16(9)	16(9)	15/15
f_5	10	10	10	10	10	10	15/15
	31(33)	50(38)	54(40)	54(40)	54(40)	54(40)	15/15
f_6	114	214	281	580	1038	1332	15/15
	7.4(3)	10(3)	12(4)	10(2)	8.0(2)	8.2(1)	15/15
f_7	24	324	1171	1572	1572	1597	15/15
	16(14)	4.1(2)	4.9(6)	7.9(7)	7.9(7)	8.0(8)	15/15
f_8	73	273	336	391	410	422	15/15
	18(5)	16(7)	24(16)	43(16)	56(22)	75(30)	15/15
f_9	35	127	214	300	335	369	15/15
	34(10)	37(26)	42(30)	72(51)	109(77)	129(74)	15/15
f_{10}	349	500	574	626	829	880	15/15
	59(45)	194(275)	407(299)	1184(1040)	1630(1600)	4070(4546)	2/15
f_{11}	143	202	763	1177	1467	1673	15/15
	18(9)	106(111)	77(75)	100(79)	140(85)	177(152)	11/15
f_{12}	108	268	371	461	1303	1494	15/15
	615(1690)	1328(1901)	9445(10793)	∞	∞	∞ *5.0e5*	0/15
f_{13}	132	195	250	1310	1752	2255	15/15
	22(8)	138(161)	603(907)	1034(1035)	2061(2253)	∞ *5.0e5*	0/15
f_{14}	10	41	58	139	251	476	15/15
	1.5(1)	15(7)	25(3)	26(9)	110(97)	1315(1232)	1/15
f_{15}	511	9310	19369	20073	20769	21359	14/15
	3.7(2)	26(24)	84(86)	81(90)	79(87)	77(84)	4/15
f_{16}	120	612	2662	10449	11644	12095	15/15
	2.2(1)	27(41)	38(51)	50(55)	86(95)	101(117)	5/15
f_{17}	5.2	215	899	3669	6351	7934	15/15
	3.7(5)	5.4(0.8)	7.7(1)	16(14)	38(37)	88(92)	4/15
f_{18}	103	378	3968	9280	10905	12469	15/15
	5.0(3)	12(2)	5.4(9)	100(123)	∞	∞ *5.0e5*	0/15
f_{19}	1	1	242	1.2e5	1.2e5	1.2e5	15/15
	28(26)	3184(1818)	859(1117)	∞	∞	∞ *5.0e5*	0/15
f_{20}	16	851	38111	54470	54861	55313	14/15
	18(15)	4.4(6)	2.4(2)	1.7(1)	1.7(1)	1.7(1)	14/15
f_{21}	41	1157	1674	1705	1729	1757	14/15
	3.2(3)	2.2(5)	2.1(4)	2.7(3)	3.3(3)	3.8(3)	15/15
f_{22}	71	386	938	1008	1040	1068	14/15
	2.6(3)	6.7(5)	11(11)	12(10)	15(11)	16(11)	15/15
f_{23}	3.0	518	14249	31654	33030	34256	15/15
	0.91(0.8)	19(19)	43(42)	222(265)	212(242)	205(219)	1/15
f_{24}	1622	2.2e5	6.4e6	9.6e6	1.3e7	1.3e7	3/15
	11(12)	5.6(6)	∞	∞	∞	∞ *5.0e5*	0/15

20-D

Δf	1e+1	1e+0	1e-1	1e-3	1e-5	1e-7	#succ
f_1	43	43	43	43	43	43	15/15
	142(31)	304(46)	485(63)	799(93)	1119(109)	1449(135)	15/15
f_2	385	386	387	390	391	393	15/15
	75(6)	93(7)	113(8)	147(10)	183(9)	220(9)	15/15
f_3	5066	7626	7635	7643	7646	7651	15/15
	144(64)	450(405)	1181(1319)	1180(1308)	1181(1177)	1182(1194)	3/15
f_4	4722	7628	7666	7700	7758	1.4e5	9/15
	236(100)	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_5	41	41	41	41	41	41	15/15
	151(38)	213(84)	228(106)	233(112)	234(112)	234(112)	15/15
f_6	1296	2343	3413	5220	6728	8409	15/15
	26(11)	55(8)	46(9)	43(8)	43(10)	43(11)	15/15
f_7	1351	4274	9503	16524	16524	16969	15/15
	80(75)	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_8	2039	3871	4040	4219	4371	4484	15/15
	50(40)	150(264)	159(253)	184(243)	213(233)	257(250)	12/15
f_9	1716	3102	3277	3455	3594	3727	15/15
	353(124)	1855(1640)	∞	∞	∞	∞ *2.0e6*	0/15
f_{10}	7413	8661	10735	14920	17073	17476	15/15
	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_{11}	1002	2228	6278	9762	12285	14831	15/15
	431(168)	396(54)	189(21)	165(8)	154(5)	∞ *2.0e6*	0/15
f_{12}	1042	1938	2740	4140	12407	13827	15/15
	185(16)	1209(1550)	10244(12227)	∞	∞	∞ *2.0e6*	0/15
f_{13}	652	2021	2751	18749	24455	30201	15/15
	183(295)	398(497)	2064(2546)	∞	∞	∞ *2.0e6*	0/15
f_{14}	75	239	304	932	1648	15661	15/15
	32(9)	45(10)	67(12)	92(18)	1640(1281)	∞ *2.0e6*	0/15
f_{15}	30378	1.5e5	3.1e5	3.2e5	4.5e5	4.6e5	15/15
	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_{16}	1384	27265	77015	1.9e5	2.0e5	2.2e5	15/15
	118(162)	∞	∞	∞	∞	∞ *2.0e6*	0/15
f_{17}	63	1030	4005	30677	56288	80472	15/15
	18(9)	2347(3013)	∞	∞	∞	∞ *2.0e6*	0/15
f_{18}	621	3972	19561	67569	1.3e5	1.5e5	15/15
	13(6)	7528(7806)	∞	∞	∞	∞ *2.0e6*	0/15
f_{19}	1	1	3.4e5	6.2e6	6.7e6	6.7e6	15/15
	975(165)	6.9e5(8e5)	∞	∞	∞	∞ *2.0e6*	0/15
f_{20}	82	46150	3.1e6	5.5e6	5.6e6	5.6e6	14/15
	34(6)	148(148)	∞	∞	∞	∞ *2.0e6*	0/15
f_{21}	561	6541	14103	14643	15567	17589	15/15
	103(6)	162(255)	97(142)	95(137)	90(128)	81(114)	10/15
f_{22}	467	5580	23491	24948	26847	1.3e5	12/15
	100(259)	259(275)	392(395)	372(385)	350(372)	70(74)	3/15
f_{23}	3.2	1614	67457	4.9e5	8.1e5	8.4e5	15/15
	1.7(2)	415(427)	∞	∞	∞	∞ *2.0e6*	0/15
f_{24}	1.3e6	7.5e6	5.2e7	5.2e7	5.2e7	5.2e7	3/15
	∞	∞	∞	∞	∞	∞ *2.0e6*	0/15

Table 1: Expected running time (ERT in number of function evaluations) divided by the best ERT measured during BBOB-2009 (given in the respective first row) for different Δf values for functions f_1–f_{24}. The median number of conducted function evaluations is additionally given in *italics*, if $\mathrm{ERT}(10^{-7}) = \infty$. #succ is the number of trials that reached the final target $f_{\mathrm{opt}} + 10^{-8}$.

Table 2: ERT loss ratio compared to the respective best result from BBOB-2009 for budgets given in the first column (see also Figure 4). The last row $\mathrm{RL_{US}}/\mathrm{D}$ 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). The ERT Loss ratio equals to one for the respective best algorithm from BBOB-2009. Typical median values are between ten and hundred.

	f_1–f_{24} in 5-D, maxFE/D=100018					
#FEs/D	best	10%	25%	**med**	75%	90%
2	0.77	1.7	2.1	2.4	3.7	8.5
10	2.6	3.3	3.8	5.1	6.8	16
100	4.2	5.0	8.7	14	21	95
1e3	2.9	4.6	7.8	16	37	98
1e4	2.7	3.8	11	42	1.4e2	4.7e2
1e5	1.8	3.6	15	63	3.6e2	1.6e3
$\mathrm{RL_{US}}/\mathrm{D}$	1e5	1e5	1e5	1e5	1e5	1e5

	f_1–f_{24} in 20-D, maxFE/D=100008					
#FEs/D	best	10%	25%	**med**	75%	90%
2	1.0	2.3	12	31	40	40
10	7.0	7.4	19	1.5e2	2.0e2	2.0e2
100	6.4	9.4	29	45	64	2.8e2
1e3	22	23	39	55	79	3.1e2
1e4	30	81	1.3e2	2.2e2	3.8e2	8.8e2
1e5	42	1.5e2	2.5e2	4.6e2	1.0e3	1.7e3
1e6	42	80	3.5e2	1.7e3	5.9e3	1.2e4
$\mathrm{RL_{US}}/\mathrm{D}$	1e5	1e5	1e5	1e5	1e5	1e5