

# A Stigmergy-Based Algorithm for Black-Box Optimization: Noiseless Function Testbed

Peter Korošec  
Jožef Stefan Institute  
Jamova cesta 39  
SI-1000 Ljubljana, Slovenia  
peter.korosec@ijs.si

Jurij Šilc  
Jožef Stefan Institute  
Jamova cesta 39  
SI-1000 Ljubljana, Slovenia  
jurij.silc@ijs.si

## ABSTRACT

In this paper, we present a stigmergy-based algorithm for solving optimization problems with continuous variables, labeled Differential Ant-Stigmergy Algorithm (DASA). The performance of the DASA is evaluated on the set of benchmark problems provided for Black-Box Optimization Benchmarking (BBOB) 2009, a GECCO Workshop for Real-Parameter Optimization. Benchmarking for noiseless function testbed is presented.

## 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, Stigmergy

## 1. INTRODUCTION

Numerical optimization problems are an important field of research and have a wide number of applications, from the mathematical optimization of functions to the real-world engineering problems. Many engineering problems involve choosing the best configuration of a set of parameters to achieve a specified objective. Numerical optimization refers to the case when these parameters take continuous values, as opposed to combinatorial optimization, which deals with discrete values.

In this paper we consider the following numerical minimization problem:

$$\min(f(\vec{x})), \vec{B}_l \leq \vec{x} \leq \vec{B}_u$$

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'09, July 8–12, 2009, Montréal Québec, Canada.  
Copyright 2009 ACM 978-1-60558-505-5/09/07 ...\$5.00.

where  $\vec{x} = (x_1, x_2, \dots, x_D)$  is the variable vector in  $\mathbb{R}^D$ ,  $f(\vec{x})$  denotes the cost function to minimize and  $\vec{B}_l = (B_{l1}, B_{l2}, \dots, B_{lD})$ ,  $\vec{B}_u = (B_{u1}, B_{u2}, \dots, B_{uD})$  represent, respectively, the lower and the upper bound of the variables, such that  $x_i \in [B_{li}, B_{ui}]$ .

Usually, numerical optimization problems are black-box optimizations, in which the cost function's form as well as its derivatives are unknown. Normally, this occurs when the cost function is computed using a complex simulation about which the optimization algorithm has no information. Executing a black-box simulation in order to evaluate a candidate solution is usually very expensive and can take up to several minutes or even hours. This is particularly problematic because optimization algorithms for black-box problems are necessarily blind search algorithms that must repeatedly sample points in a solution space, evaluate them by running the simulation, and apply various heuristics in order to choose the next points to sample.

In the past two decades, many bio-inspired optimization algorithms have been proposed to solve this kind of optimization problem, e.g., real-parameter genetic algorithm [10], evolution strategies [3], differential evolution [9], particle swarm optimization [7], immunological algorithm [2], ant-colony optimization [1], etc. These algorithms have been used to solve problems in several research fields due to the fact that do not require previous considerations regarding the problem to be optimized and offers a high degree of parallelism.

Although ant-based optimization has been proven to be one of the best metaheuristics in some combinatorial optimization problems, the application to the numerical optimizations appears more challenging, since the pheromone laying method is not straightforward. There are several possibilities to use ants for numerical optimization. We can use simplified direct simulation of real ants' behavior or we can extend method to explore continuous spaces. This extension can be done by the suitable discretization of a search space or by probabilistic sampling. In this way a fine-grained discrete form of continuous domain is created. With it we are able to represent this problem as a graph, which enables the use of ant-based approach for solving numerical optimization problems.

The remainder of this paper is organized as follows: Section 2 introduces the optimization algorithm. Sections 3 and 4 present the experimental procedure and black-box optimization benchmarking for noiseless function testbed, respectively. CPU timing experiment is presented in Section 5. Finally, Section 6 concludes the paper.

## 2. STIGMERGY-BASED ALGORITHM

### 2.1 Parameter Differences

Let  $x'_i$  be the current value of the  $i$ -th parameter. During the searching for the optimal parameter value, the new value,  $x_i$ , is assigned to the  $i$ -th parameter as follows:

$$x_i = x'_i + \delta_i. \quad (1)$$

Here,  $\delta_i$  is the so-called *parameter difference* and is chosen from the set

$$\Delta_i = \Delta_i^- \cup \{0\} \cup \Delta_i^+,$$

where

$$\Delta_i^- = \{\delta_{i,k}^- \mid \delta_{i,k}^- = -b^{k+B_{li}-1}, k = 1, 2, \dots, d_i\}$$

and

$$\Delta_i^+ = \{\delta_{i,k}^+ \mid \delta_{i,k}^+ = b^{k+B_{li}-1}, k = 1, 2, \dots, d_i\}.$$

Here,  $d_i = B_{ui} - B_{li} + 1$ . Therefore, for each parameter  $x_i$ , the parameter difference,  $\delta_i$ , has a range from  $b^{B_{li}}$  to  $b^{B_{ui}}$ , where  $b$  is the so-called *discrete base*,  $B_{li} = \lfloor \lg_b(\epsilon_i) \rfloor$ , and  $B_{ui} = \lfloor \lg_b(\max(x_i) - \min(x_i)) \rfloor$ . With the parameter  $\epsilon_i$ , the maximum precision of the parameter  $x_i$  is set. The precision is limited by the computer's floating-point arithmetic. To enable a more flexible movement over the search space, the weight  $\omega$  is added to Eq. 1:

$$x_i = x'_i + \omega \delta_i \quad (2)$$

where  $\omega = \text{RandomInteger}(1, b - 1)$ .

### 2.2 Graph Representation

From all the sets  $\Delta_i$ ,  $1 \leq i \leq D$ , where  $D$  represents the number of parameters, the so-called *differential graph*  $\mathcal{G} = (V, E)$  with a set of vertices,  $V$ , and a set of edges,  $E$ , between the vertices is constructed. Each set  $\Delta_i$  is represented by the set of vertices,  $V_i = \{v_{i,1}, v_{i,2}, \dots, v_{i,2d_i+1}\}$ , and  $V = \bigcup_{i=1}^D V_i$ . Then we have that

$$\Delta_i = \{\delta_{i,d_i}^-, \dots, \delta_{i,d_i-j+1}^-, \dots, \delta_{i,1}^-, 0, \delta_{i,1}^+, \dots, \delta_{i,j}^+, \dots, \delta_{i,d_i}^+\}$$

corresponds to

$$V_i = \{v_{i,1}, \dots, v_{i,j}, \dots, v_{i,d_i+1}, \dots, v_{i,d_i+1+j}, \dots, v_{i,2d_i+1}\},$$

where

$$\begin{aligned} v_{i,j} &\xrightarrow{\delta} \delta_{i,d_i-j+1}^-, \\ v_{i,d_i+1} &\xrightarrow{\delta} 0, \\ v_{i,d_i+1+j} &\xrightarrow{\delta} \delta_{i,j}^+, \end{aligned}$$

and  $j = 1, 2, \dots, d_i$ . Each vertex of the set  $V_i$  is connected to all the vertices that belong to the set  $V_{i+1}$ . Therefore, this is a directed graph, where each path  $\vec{p}$  from the starting vertex,  $v_1 \in V_1$ , to any of the ending vertices,  $v_D \in V_D$ , is of equal length and can be defined with  $v_i$  as  $\nu = (v_1 v_2 \dots v_i \dots v_D)$ , where  $v_i \in V_i$ ,  $1 \leq i \leq D$ .

### 2.3 Algorithm Implementation

The optimization consists of an iterative improvement of the temporary best solution,  $\vec{x}^{\text{tb}}$ , by constructing an appropriate path  $\vec{p}$ . New solutions are produced by applying  $\vec{p}$  to  $\vec{x}^{\text{tb}}$  (Eq. 2).

First a solution  $\vec{x}^{\text{tb}}$  is randomly chosen by uniform sampling and evaluated. Then a search graph is created and an

initial amount of pheromone is deposited on search graph according to the Cauchy probability density function

$$C(z) = \frac{1}{s\pi(1 + (\frac{z-l_i}{s})^2)},$$

where  $l_i$  is the location offset for the  $i$ -th parameter and

$$s = s_{\text{global}} - s_{\text{local}}$$

is the scale factor. For an initial pheromone distribution the Cauchy distribution with  $s_{\text{global}} = 10$ ,  $s_{\text{local}} = 0$ , and  $l_i = 0, i = 1, 2, \dots, D$  is used and each parameter vertices are equidistantly arranged between  $z = [-4, 4]$ .

There are  $m$  ants in a colony, all of which begin simultaneously from the starting vertex. Ants use a probability rule to determine which vertex will be chosen next. The rule is based on a simple ACO. More specifically, ant  $a$  in step  $i$  moves from a vertex in set  $V_{i-1}$  to vertex  $v_{i,j} \in V_i$  with a probability given by:

$$\text{prob}(a, v_{i,j}) = \frac{\tau(v_{i,j})}{\sum_{1 \leq k \leq 2d_i+1} \tau(v_{i,k})},$$

where  $\tau(v_{i,k})$  is the amount of pheromone in vertex  $v_{i,k}$ .

The ants repeat this action until they reach the ending vertex. For each ant  $i$ , path  $\vec{p}_i$  is constructed. If for some predetermined number of tries (in our case  $m^2$  for all ants) we get  $\vec{p}_i = \mathbf{0}$  the search process is reset by randomly choosing new  $\vec{x}^{\text{tb}}$  and pheromone re-initialization. Otherwise, a new solution  $\vec{x}_i$  is constructed.

After all ants have created solutions, they are being evaluated with a calculation of  $y_i = f(\vec{x}_i)$ . The information about the best among them is stored as currently best information ( $\vec{x}^{\text{cb}}$ ,  $\vec{p}^{\text{cb}}$ , and  $y_i^{\text{cb}}$ ).

The current best solution,  $\vec{x}^{\text{cb}}$  is compared to the temporary best solution  $\vec{x}^{\text{tb}}$ . If  $y^{\text{cb}}$  is better than  $y^{\text{tb}}$ , then temporally best information is replaced with currently best information. In this case  $s_{\text{global}}$  is increased according to the global scale increase factor,  $s_+$ :

$$s_{\text{global}} \leftarrow (1 + s_+)s_{\text{global}},$$

$s_{\text{local}}$  is set to

$$s_{\text{local}} = \frac{1}{2}s_{\text{global}}$$

and pheromone amount is redistributed according to the associated path  $\vec{p}^{\text{cb}}$ , where  $l_i = z(p_i^{\text{cb}})$ , so that the peak of Cauchy distribution is over with path selected vertex. Furthermore, if new  $y^{\text{tb}}$  is better then global best  $y^{\text{b}} = f(x^{\text{b}})$ , then globally best information is replaced with temporally best information. So, global best solution is stored. If no better solution is found  $s_{\text{global}}$  is decreased according to the global scale decrease factor,  $s_-$ :

$$s_{\text{global}} \leftarrow (1 - s_-)s_{\text{global}}.$$

Pheromone evaporation is defined by some predetermined percentage  $\rho$ . The probability density function  $C(z)$  is changed in the following way:

$$l_i \leftarrow (1 - \rho)l_i$$

and

$$s_{\text{local}} \leftarrow (1 - \rho)s_{\text{local}}.$$

Here we must emphasize that  $\rho > s_-$ , because otherwise we might get negative scale factor.

The whole procedure is then repeated until some ending condition is met.

The pseudocode of the Differential Ant-Stigmergy Algorithm (DASA) is presented as follows:

---

**Algorithm 1** The DASA

---

```

1:  $\vec{x}^{\text{tb}} = \text{Rnd\_Solution}()$ 
2:  $y^{\text{b}} = f(\vec{x}^{\text{tb}})$ 
3:  $y^{\text{tb}} = \text{inf}$ 
4:  $\mathcal{G} = \text{Graph\_Initialization}(\vec{x}^{\text{tb}}, \vec{e})$ 
5:  $\text{Pheromone\_Initialization}(\mathcal{G})$ 
6: while not ending condition met do
7:    $k = 0$ 
8:   for all  $m$  ants do
9:     repeat
10:     $\vec{p}_i = \text{Find\_Path}(\mathcal{G})$ 
11:     $k = k + 1$ 
12:    if  $k > m^2$  then
13:       $\vec{x}^{\text{tb}} = \text{Rnd\_Solution}()$ 
14:       $\text{Pheromone\_Initialization}(\mathcal{G})$ 
15:      goto line 7
16:    end if
17:    until  $(\vec{p}_i = 0)$ 
18:     $\omega = \text{Random\_Integer}(1, b - 1)$ 
19:     $\vec{x}_i = \vec{x}^{\text{tb}} + \omega\delta(\vec{p})$ 
20:  end for
21:   $y^{\text{cb}} = \text{inf}$ 
22:  for all  $m$  ants do
23:     $y = f(\vec{x}_i)$ 
24:    if  $y < y^{\text{cb}}$  then
25:       $y^{\text{cb}} = y$ 
26:       $\vec{p}^{\text{cb}} = \vec{p}_i$ 
27:       $\vec{x}^{\text{cb}} = \vec{x}_i$ 
28:    end if
29:  end for
30:  if  $y^{\text{cb}} < y^{\text{tb}}$  then
31:     $y^{\text{tb}} = y^{\text{cb}}$ 
32:     $\vec{x}^{\text{tb}} = \vec{x}^{\text{cb}}$ 
33:     $s = \text{Update\_Scales}(s_{\text{global}}, s_{\text{local}})$ 
34:     $\text{Pheromone\_Redistribution}(\vec{p}^{\text{cb}}, s)$ 
35:    if  $y^{\text{tb}} < y^{\text{b}}$  then
36:       $y^{\text{b}} = y^{\text{tb}}$ 
37:       $\vec{x}^{\text{b}} = \vec{x}^{\text{tb}}$ 
38:    end if
39:  else
40:     $\text{Update\_Scale}(s_{\text{global}})$ 
41:  end if
42:   $\text{Pheromone\_Evaporation}(\mathcal{G}, \rho)$ 
43: end while

```

---

### 3. EXPERIMENTAL PROCEDURE

The DASA includes six parameters: the number of ants,  $m$ , the pheromone dispersion factor,  $\rho$ , the global scale-increasing factor,  $s_+$ , the global scale-decreasing factor,  $s_-$ , the maximum parameter precision,  $\varepsilon$ , and the discrete base,

$b$ . The single setting for parameters' was used for all functions, i.e., the crafting effort was zero. We set  $m = 30$  and  $\rho = 0.2$ , while other parameters' settings are standard [8]:  $s_+ = 0.01$ ,  $s_- = 0.02$ ,  $\varepsilon = 1.0 \times 10^{-15}$ , and  $b = 10$ .

Maximal number of restarts was set to 1000 and maximal number of FEs was set to  $D \times 10^6$ .

### 4. RESULTS

Results from experiments according to [5] on the benchmark functions given in [4, 6] are presented in Figures 1 and 2 and in Table 1.

### 5. CPU TIMING EXPERIMENT

For the the timing experiment the DASA was run on  $f_8$ . The computer platform used to perform the experiments was based on Intel Core i7 processors at 3.5 GHz, 6 GB of RAM, and the Microsoft Windows Vista x64 operating system. The DASA was implemented in Borland Delphi and for the function testbed the original implementation of functions in C were used in a form of dynamic link library. The results were 1.0; 1.1; 2.0; 3.4; 6.5 and  $11.6 \times 10^{-6}$  seconds per function evaluation in dimension 2; 3; 5; 10; 20 and 40, respectively. We can see a proportional dependency of CPU time on the search space dimensionality.

### 6. CONCLUSIONS

In this paper a stigmergy-based algorithm called Differential Ant-Stigmergy Algorithm (DASA) for solving optimization problems with continuous variables was presented. The performance of the DASA was evaluated on the noiseless function testbed provided for Black-Box Optimization Benchmarking (BBOB).

The DASA performs very good on separable ( $f_1, \dots, f_5$ ) and moderate functions ( $f_6, \dots, f_9$ ) with exception of Step-ellipsoid function ( $f_7$ ). With weak structured functions ( $f_{20}, \dots, f_{24}$ ) it has troubles with higher dimensions for functions  $f_{20}, f_{23}$ , and  $f_{24}$ . It performs poorly on ill-conditioned ( $f_{10}, \dots, f_{14}$ ) and multi-modal ( $f_{15}, \dots, f_{19}$ ) functions with exception of Discus function ( $f_{11}$ ).

### 7. REFERENCES

- [1] G. Bilchev and I. C. Parmee. The ant colony metaphor for searching continuous design spaces. In T. C. Fogarty, editor, *Evolutionary Computing*, volume 993 of *Lecture Notes in Computer Science*, pages 25–39, Sheeld, UK, April 3-4 1995.
- [2] V. Cutello, G. Narzisi, G. Nicosia, and M. Pavone. An immunological algorithm for global numerical optimization. In E.-G. Talbi, P. Liardet, P. Collet, E. Lutton, and M. Schoenauer, editors, *Proceedings of the 7th International Conference on Artificial Evolution, Evolution Artificielle, EA 2005*, volume 3871 of *Lecture Notes in Computer Science*, pages 284–295, Lille, France, 2006. Springer-Verlag.
- [3] K. Deb, A. Anand, and D. Joshi. A computationally efficient evolutionary algorithm for real-parameter optimization. *Evolutionary Computation*, 10(4):371–395, December 2002.
- [4] S. Finck, H. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the Noiseless Functions.,

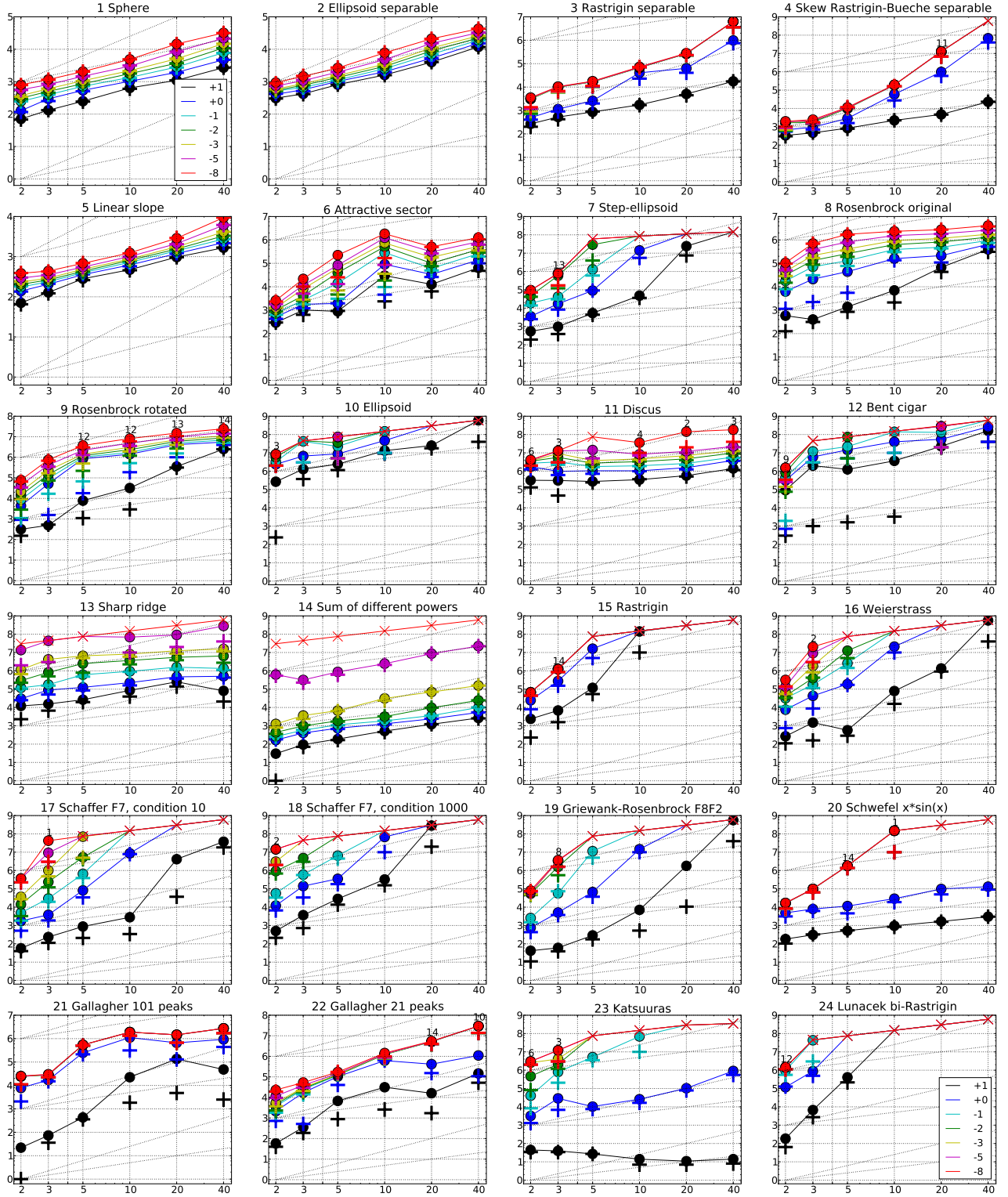
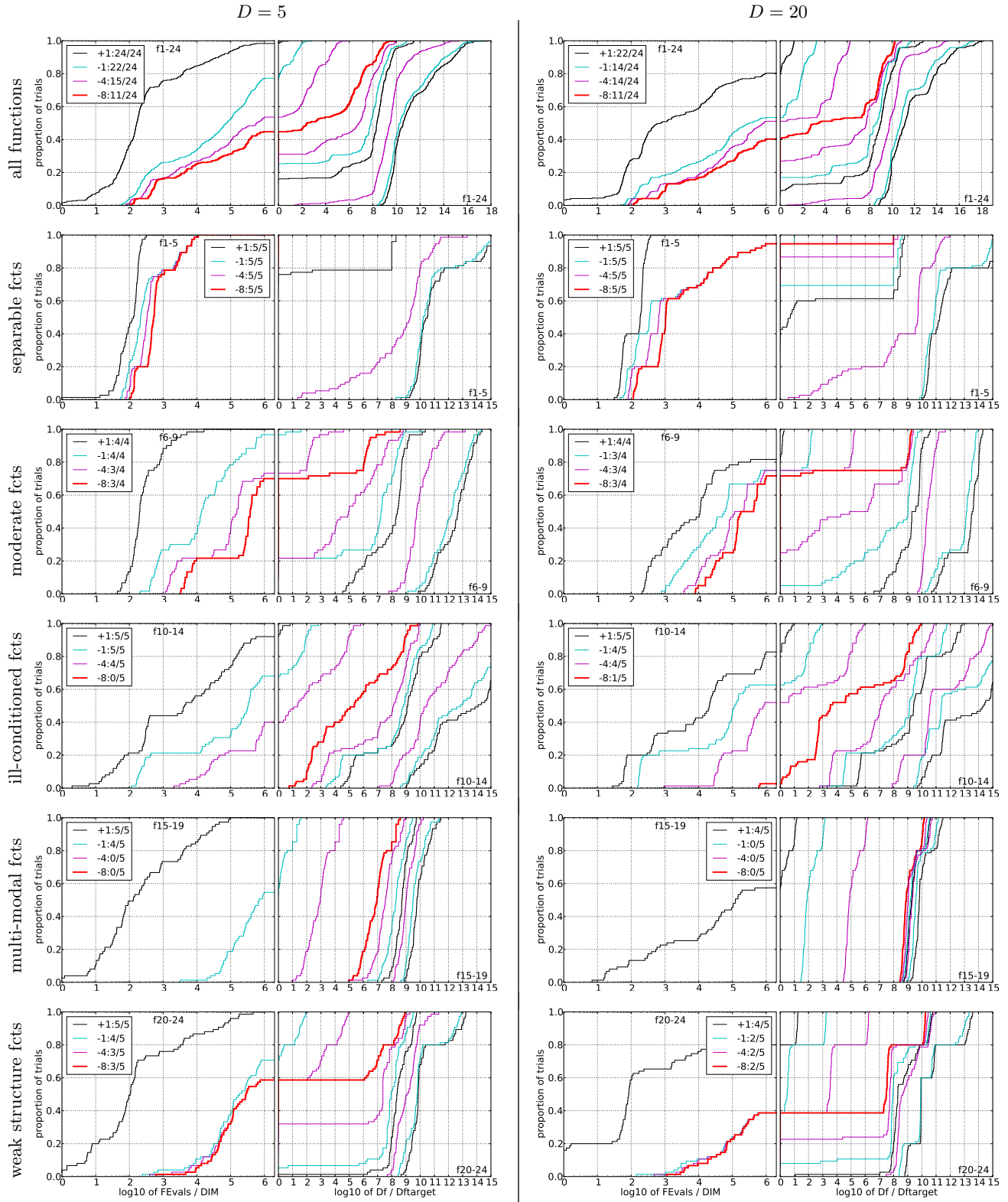


Figure 1: Expected Running Time (ERT,  $\bullet$ ) to reach  $f_{\text{opt}} + \Delta f$  and median number of function evaluations of successful trials (+), shown for  $\Delta f = 10, 1, 10^{-1}, 10^{-2}, 10^{-3}, 10^{-5}, 10^{-8}$  (the exponent is given in the legend of  $f_1$  and  $f_{24}$ ) versus dimension in log-log presentation. The ERT( $\Delta f$ ) equals to  $\text{\#Fes}(\Delta f)$  divided by the number of successful trials, where a trial is successful if  $f_{\text{opt}} + \Delta f$  was surpassed during the trial from all respective trials (successful and unsuccessful), and  $f_{\text{opt}}$  denotes the optimal function value. Crosses ( $\times$ ) indicate the total number of function evaluations  $\text{\#Fes}(-\infty)$ . 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.

f1 in 5-D, N=15, mFE=2473						f1 in 20-D, N=15, mFE=16589						f2 in 5-D, N=15, mFE=3014						f2 in 20-D, N=15, mFE=22468					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
	10	15	2.5e2	2.0e2	2.9e2	2.5e2	15	1.1e3	1.0e3	1.2e3	1.1e3	10	15	8.1e2	7.5e2	8.6e2	8.1e2	15	4.0e3	3.9e3	4.1e3	4.0e3	
	1	15	5.4e2	4.8e2	6.0e2	5.4e2	15	1.9e3	1.8e3	2.0e3	1.9e3	1	15	1.0e3	9.5e2	1.1e3	1.0e3	15	5.3e3	5.2e3	5.4e3	5.3e3	
1e-1	15	7.2e2	6.6e2	7.7e2	7.2e2	15	2.8e3	2.7e3	2.9e3	2.8e3	1e-1	15	1.1e3	1.1e3	1.2e3	1.1e3	15	7.0e3	6.9e3	7.2e3	7.0e3		
1e-3	15	1.1e3	1.0e3	1.1e3	1.1e3	15	5.1e3	5.0e3	5.2e3	5.1e3	1e-3	15	1.5e3	1.5e3	1.6e3	1.5e3	15	1.1e4	1.1e4	1.1e4	1.1e4		
1e-5	15	1.5e3	1.4e3	1.5e3	1.5e3	15	8.5e3	8.2e3	8.8e3	8.5e3	1e-5	15	2.0e3	1.9e3	2.1e3	2.0e3	15	1.5e4	1.5e4	1.6e4	1.5e4		
1e-8	15	2.0e3	2.0e3	2.1e3	2.0e3	15	1.5e4	1.4e4	1.5e4	1.5e4	1e-8	15	2.6e3	2.6e3	2.7e3	2.6e3	15	2.1e4	2.1e4	2.1e4	2.1e4		
f3 in 5-D, N=15, mFE=54700						f3 in 20-D, N=15, mFE=1032930						f4 in 5-D, N=15, mFE=34629						f4 in 20-D, N=15, mFE=20000028					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
	10	15	8.7e2	8.1e2	9.3e2	8.7e2	15	5.1e3	4.7e3	5.4e3	5.1e3	10	15	8.4e2	7.2e2	9.6e2	8.4e2	15	4.7e3	4.4e3	5.1e3	4.7e3	
	1	15	2.6e3	1.9e3	3.4e3	2.6e3	15	6.3e4	4.2e4	8.5e4	6.3e4	1	15	2.9e3	2.0e3	3.8e3	2.9e3	15	9.9e5	6.8e5	1.3e6	9.9e5	
1e-1	15	1.7e4	1.1e4	2.2e4	1.7e4	15	2.7e5	1.9e5	3.5e5	2.7e5	1e-1	15	1.0e4	7.0e3	1.3e4	1.0e4	15	1.3e7	9.0e6	1.7e7	1.0e7		
1e-3	15	1.7e4	1.1e4	2.2e4	1.7e4	15	2.7e5	1.9e5	3.6e5	2.7e5	1e-3	15	1.0e4	7.3e3	1.3e4	1.0e4	15	1.3e7	9.4e6	1.7e7	1.0e7		
1e-5	15	1.7e4	1.2e4	2.3e4	1.7e4	15	2.7e5	1.9e5	3.6e5	2.7e5	1e-5	15	1.1e4	7.6e3	1.4e4	1.1e4	15	1.3e7	9.2e6	1.7e7	1.0e7		
1e-8	15	1.8e4	1.3e4	2.3e4	1.8e4	15	2.8e5	2.0e5	3.6e5	2.8e5	1e-8	15	1.1e4	8.5e3	1.4e4	1.1e4	15	1.3e7	9.4e6	1.7e7	1.0e7		
f5 in 5-D, N=15, mFE=864						f5 in 20-D, N=15, mFE=4064						f6 in 5-D, N=15, mFE=2126106						f6 in 20-D, N=15, mFE=1024998					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
	10	15	2.8e2	2.5e2	3.1e2	2.8e2	15	9.7e2	9.2e2	1.0e3	9.7e2	10	15	8.9e2	7.5e2	1.0e3	8.9e2	15	1.3e4	9.4e3	1.6e4	1.3e4	
	1	15	3.6e2	3.2e2	3.9e2	3.6e2	15	1.2e3	1.1e3	1.2e3	1.2e3	1	15	1.9e3	1.8e3	2.1e3	1.9e3	15	3.4e4	2.7e4	4.1e4	3.4e4	
1e-1	15	4.0e2	3.7e2	4.3e2	4.0e2	15	1.4e3	1.3e3	1.5e3	1.4e3	1e-1	15	1.4e4	2.9e3	2.6e4	1.4e4	15	6.1e4	4.9e4	7.3e4	6.1e4		
1e-3	15	4.9e2	4.6e2	5.1e2	4.9e2	15	1.8e3	1.7e3	1.8e3	1.8e3	1e-3	15	6.3e4	7.0e3	1.2e5	6.3e4	15	1.8e5	1.5e5	2.2e5	1.8e5		
1e-5	15	5.5e2	5.3e2	5.8e2	5.5e2	15	2.1e3	2.0e3	2.2e3	2.1e3	1e-5	15	8.4e4	1.4e4	1.5e5	8.4e4	15	3.1e5	2.5e5	3.8e5	3.1e5		
1e-8	15	6.8e2	6.5e2	7.1e2	6.8e2	15	2.9e3	2.7e3	3.0e3	2.9e3	1e-8	15	2.2e5	3.1e4	4.2e5	2.2e5	15	5.1e5	4.2e5	6.1e5	5.1e5		
f7 in 5-D, N=15, mFE=4128942						f7 in 20-D, N=15, mFE=7811742						f8 in 5-D, N=15, mFE=2153325						f8 in 20-D, N=15, mFE=3514407					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
	10	15	5.3e3	3.2e3	7.7e3	5.3e3	4	2.4e7	2.1e7	2.7e7	6.1e6	10	15	1.4e3	9.5e2	1.9e3	1.4e3	15	6.7e4	4.4e4	9.3e4	6.7e4	
	1	15	9.1e4	6.6e4	1.2e5	9.1e4	0	13e+0	79e-1	17e+0	2.0e6	1	15	4.4e4	2.1e4	6.8e4	4.4e4	15	2.1e5	1.4e5	2.8e5	2.1e5	
1e-1	14	1.3e6	8.4e5	1.7e6	1.3e6						1e-1	15	1.3e5	8.2e4	1.8e5	1.3e5	15	4.9e5	3.8e5	5.9e5	4.9e5		
1e-3	0	16e-3	10e-3	79e-3	2.0e6						1e-3	15	4.1e5	3.3e5	4.9e5	4.1e5	15	1.2e6	1.0e6	1.3e6	1.2e6		
1e-5											1e-5	15	8.2e5	7.3e5	9.0e5	8.2e5	15	1.8e6	1.7e6	1.9e6	1.8e6		
1e-8											1e-8	15	1.7e6	1.7e6	1.8e6	1.7e6	15	2.7e6	2.6e6	2.9e6	2.7e6		
f9 in 5-D, N=15, mFE=5000012						f9 in 20-D, N=15, mFE=20000012						f10 in 5-D, N=15, mFE=5000011						f10 in 20-D, N=15, mFE=20000011					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
	10	15	7.8e3	1.7e3	1.4e4	7.8e3	15	3.6e5	3.1e5	4.0e5	3.6e5	10	12	2.4e6	1.8e6	3.1e6	2.2e6	10	2.5e7	2.3e7	2.7e7	1.7e7	
	1	14	8.5e5	3.0e5	1.5e6	8.5e5	15	4.1e6	2.5e6	5.7e6	4.1e6	1	6	8.8e6	7.4e6	1.1e7	5.0e6	0	72e-1	34e-1	59e+0	2.0e7	
1e-1	14	1.0e6	5.0e5	1.6e6	1.0e6	15	4.9e6	3.3e6	6.7e6	4.9e6	1e-1	3	2.2e7	2.0e7	2.4e7	5.0e6							
1e-3	14	1.4e6	8.9e5	2.0e6	1.4e6	15	7.2e6	5.5e6	8.9e6	7.2e6	1e-3	1	7.3e7	7.1e7	7.5e7	5.0e6							
1e-5	13	2.2e6	1.6e6	2.8e6	2.1e6	15	9.7e6	8.2e6	1.1e7	9.7e6	1e-5	1	7.4e7	7.4e7	7.5e7	5.0e6							
1e-8	12	3.8e6	3.1e6	4.3e6	3.2e6	13	1.5e7	1.3e7	1.7e7	1.3e7	1e-8	0	47e-1	75e-4	20e+0	4.5e6							
f11 in 5-D, N=15, mFE=5000011						f11 in 20-D, N=15, mFE=20000012						f12 in 5-D, N=15, mFE=5000011						f12 in 20-D, N=15, mFE=20000011					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
	10	15	2.7e5	2.1e5	3.4e5	2.7e5	15	5.7e5	5.0e5	6.4e5	5.7e5	10	12	1.3e6	4.2e5	2.1e6	1.3e6	7	2.3e7	1.7e7	3.1e7	5.7e6	
	1	15	1.0e6	8.4e5	1.3e6	1.0e6	15	1.5e6	1.3e6	1.8e6	1.5e6	1	4	1.4e7	1.2e7	1.7e7	3.8e6	4	5.5e7	4.5e7	6.5e7	1.0e7	
1e-1	15	1.8e6	1.6e6	2.0e6	1.8e6	15	2.6e6	2.2e6	3.1e6	2.6e6	1e-1	3	2.2e7	2.0e7	2.4e7	3.3e6	2	1.3e8	1.1e8	1.5e8	1.0e7		
1e-3	13	3.8e6	3.4e6	4.2e6	3.3e6	15	7.1e6	6.0e6	8.2e6	7.1e6	1e-3	0	32e-1	96e-3	13e+0	4.5e6	1	2.8e8	2.6e8	3.0e8	2.0e7		
1e-5	5	1.4e7	1.4e7	1.5e7	4.5e6	14	1.2e7	1.1e7	1.4e7	1.1e7	1e-5						1	2.8e8	2.6e8	3.0e8	2.0e7		
1e-8	0	17e-6	14e-7	12e-4	4.5e6	2	1.4e8	1.4e8	1.5e8	1.7e7	1e-8						0	13e+0	44e-3	31e+0	2.0e7		
f13 in 5-D, N=15, mFE=5000028						f13 in 20-D, N=15, mFE=20000029						f14 in 5-D, N=15, mFE=5000012						f14 in 20-D, N=15, mFE=20000011					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
	10	15	2.6e4	1.7e4	3.5e4	2.6e4	15	2.5e5	1.5e5	3.6e5	2.5e5	10	15	1.9e2	1.5e2	2.3e2	1.9e2	15	1.2e3	1.1e3	1.3e3	1.2e3	
	1	15	1.3e5	8.9e4	1.7e5	1.3e5	15	4.7e5	3.4e5	6.1e5	4.7e5	1	15	7.4e2	6.3e2	8.5e2	7.4e2	15	2.4e3	2.3e3	2.5e3	2.4e3	
1e-1	15	6.1e5	3.6e5	9.3e5	6.1e5	15	1.6e6	1.2e6	2.0e6	1.6e6	1e-1	15	1.1e3	1.0e3	1.3e3	1.1e3	15	3.6e3	3.5e3	3.8e3	3.6e3		
1e-3	7	6.6e6	5.0e6	8.0e6	3.2e6	12	1.3e7	1.0e7	1.5e7	9.5e6	1e-3	15	6.7e3	5.6e3	7.9e3	6.7e3	15	7.0e4	6.6e4	7.4e4	7.0e4		
1e-5	5	1.4e7	1.4e7	1.5e7	4.5e6	3	9.1e7	8.4e7	9.9e7	2.0e7	1e-5	15	8.7e5	7.0e5	1.1e6	8.7e5	15	8.6e6	7.9e6	9.2e6	8.6e6		
1e-8	0	23e-4	25e-6	14e-3	1.4e6	0	51e-6	43e-7	29e-4	7.9e6	1e-8	0	12e-7	18e-8	25e-7	4.5e6	0	43e-7	31e-7	55e-7	2.0e7		
f15 in 5-D, N=15, mFE=5000028						f15 in 20-D, N=15, mFE=20000028						f16 in 5-D, N=15, mFE=5000025						f16 in 20-D, N=15, mFE=20000027					
Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc	Δf	#	ERT	10%	90%	RTsucc	#	ERT	10%	90%	RTsucc		
	10	15	1.2e5	6.9e4	1.7e5	1.2e5	0	11e+1	85e+0	14e+1	1.1e7	10	15	5.8e2	2.9e2	8.7e2	5.8e2	15	1.4e6	1.0e6	1.7e6	1.4e6	
	1	4	1.6e7	1.4e7	1.8e7	4.7e6	0	20e-1	99e-2	30e-1	2.0e6	1	15	1.9e5	1.4e5	2.3e5	1.9e5	0	44e-1	32e-1	69e-1	1.1e7	
1e-1	0	20e-1	99e-2	30e-1	2.0e6						1e-1	13	2.6e6	1.9e6	3.2e6	2.4e6							
1e-3											1e-3	0	24e-3	38e-4	10e-2	2.0e6							



**Figure 2: Empirical cumulative distribution functions (ECDFs), plotting the fraction of trials versus running time (left) or  $\Delta f$ .** 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  $\Delta f$  divided by  $10^k$  (upper left lines in continuation of the left subplot), and best achieved  $\Delta f$  divided by  $10^{-8}$  for running times of  $D, 10D, 100D \dots$  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  $\Delta f$  and Df denote the difference to the optimal function value.

Technical Report 2009/20, Research Center PPE, 2009.

- [5] N. Hansen, A. Auger, S. Finck, and R. Ros. Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA, 2009.
- [6] 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.
- [7] J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, volume IV, pages 1942–1948, Perth, Australia, December 1995. IEEE Service Center, Piscataway, NJ.
- [8] P. Korošec and J. Šilc. The differential ant-stigmergy algorithm applied to dynamic optimization problems. In *Proceedings of the IEEE Congress on Evolutionary Computation*, Trondheim, Norway, May 2009. IEEE, Piscataway, NJ.
- [9] R. Storn and K. V. Price. Differential evolution – a fast and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.
- [10] A. H. Wright. Genetic algorithms for real parameter optimization. In G. J. E. Rawlins, editor, *Foundations of Genetic Algorithms - 1*, pages 205–218, San Mateo, CA, 1991. Morgan Kaufman.