
A Comparison of Global Search Algorithms for Continuous Black-Box Optimization

Petr Pošík

Faculty of Electrical Eng., Czech Technical University in Prague, Czech Republic

posik@labe.felk.cvut.cz

Waltraud Huyer

Faculty of Mathematics, University of Wien, Austria

Waltraud.Huyer@univie.ac.at

László Pál

Faculty of Business and Humanities, Sapientia – Hungarian University of Transylvania, Miercurea-Ciuc, Romania

pallaszlo@sapientia.siculorum.ro

Abstract

Four methods for global numerical black-box optimization with the origins in mathematical programming community are described and experimentally compared with the state-of-the-art evolutionary method, BIPOP-CMA-ES. The methods chosen for the comparison exhibit various features potentially interesting for the evolutionary computation community: systematic sampling of the search space (DIRECT, MCS) possibly combined with a local search method (MCS), or a multistart approach (NEWUOA, GLOBAL) possibly equipped with a careful selection of points to run a local optimizer from (GLOBAL). The recently proposed “comparing continuous optimizers” (COCO) methodology was adopted as the basis for the comparison. Based on the results, we draw suggestions about which algorithm should be used depending on the available budget of function evaluations, and we propose several possibilities for hybridizing evolutionary algorithms with features of the other compared algorithms.

Keywords

Real-parameter optimization, continuous domain, black-box optimization, benchmarking, deterministic global optimization, stochastic global optimization.

1 Introduction

Global optimization is a subfield of applied numerical analysis which studies methods that should be able to find the globally optimal solution to an optimization problem. The issues and methods of global optimization are studied in several different communities. This article focuses on mathematical programming (MP) and evolutionary computation (EC) communities.

The ultimate goal of global black-box optimization—to make the fastest possible progress towards the best possible solution—is certainly common to both communities. There are, however, certain differences. The MP community strives for methods with sound theoretical properties. The methods often search the space systematically, build and use models of the objective function, and/or store all the points sampled during the run. Maintaining the models or using the archive of sampled points is very time and space-consuming. Due to practical limits in available CPU time and storage space, the MP community usually tests these algorithms using relatively small budgets of allowed function evaluations. As a result, the MP methods are designed to show a good

progress right from the beginning of the search. The EC community studies algorithms with roots in nature and biology, often using randomly initialized populations. An EC algorithm needs some time to move the population to a promising region of the search space, and its performance in the initial phases is usually not as good as it could be. On the other hand, EC methods usually do not use any complex model or solution-archive maintenance procedures and can be tested with higher evaluations budgets. As a consequence of these differences, the findings based on experimental results in these communities are often contradicting. This contributes to the gap between these two communities, and despite the fact we can learn a lot from each other, it does not happen very often.

This article adds a brick to the bridge between the MP and EC communities. It (re-)introduces several MP algorithms to the EC community and by means of experimental comparison, it highlights the differences in them, identifies the suitable algorithms for various goals and situations, and finally points out the features that may be profitable for the members of the EC community, and vice versa.

The article focuses on three MP methods with features not widely known in the EC community. These are complemented with two reference algorithms which are not discussed in such a detail as the three main algorithms since the experiments were not performed by us (the authors of the paper) and we have thus only a limited experience with them. The first method chosen for the comparison is the DIRECT algorithm (Jones et al., 1993). It systematically samples points from the search space and does not contain any dedicated local search method. The second algorithm, MCS (Huyer and Neumaier, 1999), works on similar principles as DIRECT, but contains also a specialized local search procedure. Both algorithms were also described as “good complete general purpose global optimization algorithms” by Neumaier (2004, sec. 9). The third algorithm, GLOBAL (Csendes, 1988), is an example of a multistart method equipped with a filter trying to prevent starting a local search in the basin of attraction of an already known local optimum. To contrast the effect of this filter with a usual multistart method, the restarted version of the NEWUOA algorithm (Powell, 2006) was also included in the comparison. It is a local optimizer proposed quite recently, but its reported results are promising. The final algorithm is the BIPOP-CMA-ES by Hansen (2009), a restarted version of the state-of-the-art CMA-ES algorithm using different population sizes in individual restarts. It represents the only evolutionary approach in the comparison and serves as the baseline algorithm. All the methods are described in Sec. 2.

A suitable experimental framework must be chosen to discover the potentially profitable features of the algorithms. The framework must be able to show the differences among the algorithms at all stages of the search, not just after certain number of evaluations, as is the usual practice. The COCO (Comparing Continuous Optimizers) methodology (Hansen et al., 2009a) was chosen since it fulfills these requirements. It was used as the basis of the Black-Box Optimization Benchmarking (BBOB) workshops of the GECCO 2009 and 2010 conferences. The testbed consists of 24 carefully chosen noiseless benchmark functions (Hansen et al., 2009b) which represent various types of difficulties observed in real-world problems (ill-conditioning, multimodality, etc.). Up to 40-dimensional search space was used during the experiments. The COCO experimental framework is described in Sec. 3.

The results of the algorithms were already separately presented as the workshop papers (Hansen, 2009; Pošík, 2009; Ros, 2009b), or as unpublished reports (Huyer and Neumaier, 2009; Pál et al., 2009). One of the original contributions of this article is to

collect these results, compare them conveniently in one place, and provide a discussion of the pros and cons of the algorithms compared to each other. In the original papers, the discussion (if any) was based solely on the results of the respective algorithm and no comparison was made. We also discuss the results in more detail than the summary paper of Hansen et al. (2010).

The setup of the experiments and the algorithm settings are described in Sec 4. The results of the comparison are presented in Sec. 5. Sections 6 and 7 contain the discussion of the results broken down by the function group and by the algorithm, respectively. Sec. 8 summarizes the article and suggests several possible ways of using some of the MP principles to improve the evolutionary algorithms.

2 The Compared Algorithms

All the described algorithms are iterative. They sequentially sample points from the real-valued search space $X \subset \mathcal{R}^D$, where D is the search space dimensionality. It is assumed hereafter that the points are evaluated as soon as they are sampled, and that the variables holding the best point found so far, \mathbf{x}^{best} , its function value, f_{best} , and the number of objective function evaluations are updated accordingly.

2.1 DIRECT

The DIRECT algorithm was introduced by Jones et al. (1993). The algorithm name not only expresses that it belongs to the class of direct search algorithms, it also describes the main principle of the algorithm: the DIRECT acronym stands for *DI*viding *RE*CTan-gles. A slightly modified MATLAB implementation of DIRECT by Finkel (2003) is used. Only the basic algorithm design principles are described here; for the detailed description, see the original article by Jones et al. (1993) or the implementation description by Finkel (2003). The pseudocode of the algorithm is shown as Alg. 1.

Algorithm 1: The DIRECT optimization algorithm

```

1 Normalize the search space to be the unit box, and evaluate its base point.
2 while the termination condition is not satisfied do
3   Identify the set  $S$  of potentially optimal boxes.
4   foreach potentially optimal box  $b$  in  $S$  (with the base point  $\mathbf{c}$ ) do
5     Determine the maximal side length  $\delta$  of box  $b$ .
6     Determine the set  $I$  of dimensions where  $b$  has the side length  $\delta$ .
7     Sample the points  $\mathbf{c} \pm \frac{1}{3}\delta\mathbf{e}_i$  for all  $i \in I$ , where  $\mathbf{e}_i$  are the unit vectors.
8     Divide the box containing  $\mathbf{c}$  into thirds along the dimensions in  $I$ : start
       with the dimension with the lowest  $w_i = \min(\mathbf{c} + \frac{1}{3}\delta\mathbf{e}_i, \mathbf{c} - \frac{1}{3}\delta\mathbf{e}_i)$  and
       continue to the dimension with the largest  $w_i$ .
9 return  $\mathbf{x}^{\text{best}}$ ,  $f_{\text{best}}$ , the number of evaluations needed.
```

The algorithm is a branching scheme which recursively divides the search space and forms a tree of hyperrectangles (boxes). The leaves of the tree form a set of non-overlapping boxes; in each time instant, the whole search space is completely covered by all the leaves. The point \mathbf{c} in the middle of each box—the base point—is evaluated. Each box thus has two important characteristics: (1) the function value of its base point, and (2) the size of the box. There are many possible definitions of the box size, here the distance from the basepoint to the box corner is used.

In each iteration the algorithm decides which of the existing boxes should be split (see Alg. 1, line 3). The potentially optimal boxes are identified using 2 design principles. It is expected that the chance of finding an improvement inside a box is proportional to

- the fitness of the base point (exploitation), and to
- the box size (exploration, global search).

The identification of the potentially optimal boxes is thus basically a multi-objective problem. Each iteration, all the non-dominated boxes described by their size and their base point function value are divided by the algorithm (see Alg. 1, lines 5–8). The division of boxes which are not potentially optimal, i.e. small boxes and boxes with worse base points, is thus postponed to the following iterations.

The DIRECT algorithm does not contain any local search method which could be used to improve its efficiency. The algorithm is guaranteed to eventually sample a point arbitrarily close to the global optimum, if it is allowed to run sufficiently long and if the splitting procedure is not constrained by a maximal depth.

2.2 MCS

Inspired by DIRECT, the global optimization algorithm *multilevel coordinate search* (MCS) was developed by Huyer and Neumaier (1999) to minimize an objective function on a box $[\mathbf{u}, \mathbf{v}]$ with finite or infinite bounds. The algorithm proceeds by splitting the search space into smaller boxes, but the splitting procedure is much more irregular than the one in DIRECT. By starting a local search from certain good points, an improved result is obtained. The pseudocode of the basic steps of MCS can be found in Alg. 2. The implementation used for the experiments can be downloaded from

<http://www.mat.univie.ac.at/~neum/software/mcs/>.

Each box in the partitioning process is characterized by (1) its bounds, (2) its “base point”, and by (3) its level $s \in \{1, \dots, s_{\max}\}$. The function is evaluated at the base points. They may lie on the box boundary and therefore the same base point can be shared by two or more boxes. The level of a box is a rough measure of the number of times the box has been processed. Like DIRECT, the MCS algorithm combines exploration (splitting boxes with a large unexplored territory) and exploitation (splitting boxes with good function values). Boxes with the level s_{\max} are considered too small for further splitting. Whenever a box with the level $s < s_{\max}$ is split, its descendants get the level $s + 1$ or $s + 2$. At each stage of the algorithm, the partitioning of the search space consists of a set of boxes with levels between 1 and s_{\max} .

The algorithm starts with the so-called initialization procedure (lines 2–5 of Alg. 2). For each coordinate $i = 1, \dots, D$, at least three values $x_i^1 < x_i^2 < \dots < x_i^{L_i}$ in $[u_i, v_i]$ are needed, where $L_i \geq 3$ and the coordinates x_i^0 of the initial point belong to $\{x_i^1, \dots, x_i^{L_i}\}$. These values are used whenever a box is split in the coordinate i for the first time (in the initialization procedure or later). Splits are made (at the values of the initialization list and between) into at least $2(L_i - 1) \geq 4$ parts along each coordinate $i = 1, \dots, D$.

The main iteration loop (lines 6–20 of Alg. 2) proceeds (in the absence of other stopping criteria) until all boxes of the current partitioning have the level s_{\max} . Additional stopping criteria like reaching a target function value or a limit on the number of function evaluations are implemented but not shown in the pseudocode. In each iteration, the algorithm splits one box at each level, starting with the smallest non-empty level (i.e., with the largest boxes). When a box with the base point \mathbf{x} is split, that is done along a single coordinate i and the function is evaluated at one or more points differing

Algorithm 2: The MCS algorithm

Input: \mathbf{u}, \mathbf{v} , function f , $\mathbf{x}^0 \in [\mathbf{u}, \mathbf{v}]$, init. list x_i^j ($j = 1, \dots, L_i, i = 1, \dots, D$), s_{\max}

- 1 $\mathbf{x} \leftarrow \mathbf{x}^0; b \leftarrow [\mathbf{u}, \mathbf{v}]; level(b) \leftarrow 1$
- 2 **for** $i \leftarrow 1$ **to** D **do**
- 3 $\mathbf{x} \leftarrow$ the best of $\{\mathbf{x}^j\}_{j=1}^{L_i}$, where \mathbf{x}^j is \mathbf{x} with x_i changed to x_i^j .
- 4 Split b along the i^{th} coordinate at x_i^j ($j = 1, \dots, L_i$) and between.
- 5 $b \leftarrow$ the largest box containing \mathbf{x} .
- 6 **while** there are boxes of level $s < s_{\max}$ **do**
- 7 **for** all non-empty levels $s \leftarrow 2$ **to** $s_{\max} - 1$ **do**
- 8 Choose the box b at the level s with the lowest function value.
- 9 $i \leftarrow$ the coordinate used least often when producing b .
- 10 **if** $s > 2D(i + 1)$ **then** // *splitting by rank*
- 11 Split the box b along the i^{th} coordinate.
- 12 **else if** box not tagged as not promising **then** // *split. by exp. gain*
- 13 Determine the most promising splitting coordinate i .
- 14 Compute the (minimal) expected function value f_{exp} at new point(s).
- 15 **if** $f_{\text{exp}} < f_{\text{best}}$ **then**
- 16 Split b along the i^{th} coordinate.
- 17 **else**
- 18 Tag b as not promising, increase its level by 1.
- 19 **for** base points \mathbf{x} of all the new boxes at level s_{\max} **do**
- 20 Start a local search from \mathbf{x} if improvement is expected.
- 21 **return** $\mathbf{x}^{\text{best}}, f_{\text{best}}$, number of evaluations needed.

from \mathbf{x} only in the coordinate i .

To split a box at the level s with the base point \mathbf{x} and the bounds $\underline{\mathbf{x}}$ and $\bar{\mathbf{x}}$, the algorithm has to choose the splitting dimension i and the position of the split (based on information gained from already sampled points). Two kinds of splits can occur:

1. *Splitting by rank*: If a box has already reached a high level but still hasn't been split very often in some coordinate i , the function is evaluated at a point obtained by changing the i^{th} coordinate of \mathbf{x} to a value depending on x_i, \underline{x}_i and \bar{x}_i , and the box is split into three parts.
2. *Splitting by expected gain*: Otherwise, the splitting coordinate i and the i^{th} coordinate of the new point are determined by building a separable local quadratic model around \mathbf{x} and minimizing it, with safeguards to prevent too narrow splits. Two or three subboxes are obtained.

In both cases, the given recipes only apply to the case that the box has already been split along the coordinate i . If that is not the case, the function is evaluated at the points obtained by changing x_i to the other values of the initialization list. The splits between two points where the function has been evaluated (according to the initialization list or otherwise) are not made symmetrically: the part with the lower function value gets the larger space. The larger parts of splits get the level $s + 1$ and the smaller parts get the level $\min(s + 2, s_{\max})$.

MCS with local search (line 20) tries to accelerate the convergence of the algorithm by starting local searches from the points belonging to boxes of level s_{\max} . The local search algorithm essentially consists of building a local quadratic model by triple searches, then defining a promising search direction by minimizing the quadratic model on a suitable box and finally making a line search along this direction. This is repeated until the maximal number of iterations $n_{s_{\text{loc}}}$ for the local search algorithm is reached, the algorithm does not make any progress any more, or the estimated gradient becomes too small (unless one of the stopping criteria used for MCS is satisfied first).

If the number of levels s_{\max} goes to infinity, MCS is guaranteed to converge to the globally optimal function value if the objective function is continuous in the neighborhood of a global optimizer. This follows from the fact that then the set of points sampled by MCS is a dense subset of the search space.

2.3 GLOBAL

The stochastic global optimization method called GLOBAL (Csendes, 1988) was inspired by Boender et al. (1982) and was developed to solve bound constrained global optimization problems with black-box type objective functions. The goal of GLOBAL is to find all local minima that are potentially global. For this purpose it is equipped with a multistart strategy and clustering to promote finding distinct local optima.

Based on the old GLOBAL method (Csendes, 1988), a new version (Csendes et al., 2008) was developed in MATLAB. The algorithm was carefully studied and it was modified in some places to achieve better reliability and efficiency while allowing higher dimensional problems to be solved. In the new version we use the quasi-Newton local search method with the BFGS update instead of the earlier DFP. The algorithm implementation is available for academic and nonprofit purposes at

<http://www.inf.u-szeged.hu/~csendes/Reg/regform.php>.

The main steps of GLOBAL are summarized in Alg. 3. As a multistart method, GLOBAL iteratively samples new points from the search space X according to the uniform distribution (global phase, line 3 of Alg. 3), and executes a local search procedure starting from some of those points (local phase, line 8). The GLOBAL differs from the other multistart methods in two important aspects:

- Not all the points sampled during the global phase (the cumulated sample S_C) are considered as good candidates for starting the local search. Only the best 100γ % of them are used (the reduced sample S_R , line 4).
- These selected points are further filtered. The algorithm tries to prevent running a local search in the basin of attraction of an already detected local minimizer.

The filter is realized by a clustering procedure. The goal of the clustering procedure is to maintain one cluster per basin of attraction of a local optimum. GLOBAL uses the Single Linkage clustering rule (Boender et al., 1982; Rinnooy Kan and Timmer, 1987). The clusters are updated each iteration and only grow with time. A new point is added to the cluster if it is within a critical distance from a point already in the cluster initiated by a seed point. If the new point is not close enough to any already clustered point, it remains unclustered and thus a candidate for the local search initiation. The seed points are the local optima found so far, i.e. the members of the X^* set. The distribution of all the clustered points approximates the level set of the function; each connected component of the level set (each cluster) then approximates the shape of one basin of attraction initiated by its respective local optimum $\mathbf{x}^* \in X^*$.

The filtering (using clustering) is applied each iteration after the reduced set cre-

Algorithm 3: The GLOBAL optimization algorithm

```

1  $X^* \leftarrow \emptyset; S_C \leftarrow \emptyset; S_R \leftarrow \emptyset$ 
2 while the global stopping rule is not satisfied do
3   Sample  $N$  new points from  $X$  and add them to the cumulated sample  $S_C$ .
4    $S_R \leftarrow$  the  $\gamma|S_C|$  best points from the cumulated sample  $S_C$ .
5   Cluster the unclustered points from  $S_R$  using  $X^*$  as seed points.
6   while there are unclustered points in  $S_R$  do
7      $\mathbf{x} \leftarrow$  the best unclustered point from  $S_R$ .
8      $\mathbf{x}^* \leftarrow$  the local optimum found by running the local search from  $\mathbf{x}$ .
9     Add  $\mathbf{x}$  to the cluster of  $\mathbf{x}^*$ .
10    if  $\mathbf{x}^*$  is a new local minimizer (i.e.,  $\mathbf{x}^* \notin X^*$ ) then
11      Add  $\mathbf{x}^*$  to  $X^*$ .
12    Cluster the unclustered points from  $S_R$  using  $\mathbf{x}^*$  as the seed point.
13 return  $\mathbf{x}^{\text{best}}, f_{\text{best}}$ , the number of evaluations needed.

```

ation (line 4), and after the identification of a new seed point (line 12). In the first iteration, X^* is empty and thus no clustering takes place. The applied critical distance depends on the total sample size $|S_C|$ and it is constructed in such a way that the probability that the local method will be started tends to zero when the size of the sample grows (Boender et al., 1982; Rinnooy Kan and Timmer, 1987). The algorithm stops the search when it does not find any new local minimizer during the last iteration.

2.4 Reference algorithms

Two other optimization algorithms were selected as competitors for the just described global search algorithms: the NEWUOA procedure and the bi-population evolutionary strategy with covariance matrix adaptation (BIPOP-CMA-ES).

NEWUOA (Powell, 2006) was selected since it is a relatively recent optimization procedure with very promising reported results on various test functions. It is a deterministic (with the exception of initialization) local search procedure using quadratic modeling and a trust-region approach. The method maintains a quadratic model of the objective function in the trust region. Before each iteration, the model must interpolate the function at m points, with m typically equal to $2D + 1$, which is a much lower number of constraints than would be needed to specify a full quadratic model. The remaining degrees of freedom are taken up by minimizing the Frobenius norm of the difference between the new and the old quadratic model.

BIPOP-CMA-ES (Hansen, 2009) was chosen since it was one of the best algorithms in the BBOB-2009 comparison regarding the proportion of functions solved (Hansen et al., 2010). It is a multistart strategy using the original CMA-ES algorithm (with slightly modified parameter values) as the basic local search engine. The individual restarts differ only in the population size. Two strategies of population size setting are interlaced. The first strategy multiplies its population size by a factor of 2 each time it is executed. The second strategy chooses the population size randomly, somewhere between the initial minimal population size and the half of the last population size used by the first strategy. Increasing the population size slows down the algorithm convergence, on the other hand it results in a more global and robust search.

3 Experimental Framework Description

The experiments presented in this article were carried out using the Comparing Continuous Optimizers (COCO) framework (Hansen et al., 2009a), which was also used as the basis for the Black-box Optimization Benchmarking workshop at the GECCO 2009 and 2010 conferences.

The numerical experiments are performed on a testbed consisting of 24 noiseless test functions (Finck et al., 2009a; Hansen et al., 2009b). These functions have been constructed so that they reflect the real-world application difficulties and are categorized by function properties as multi-modality, ill-conditioning, global structure and separability. The role of the categories is to reveal the different aspects of the algorithms. All functions are scalable with the dimension D . The search domain is $[-5; 5]^D$, where $D = 2, 3, 5, 10, 20, 40$. Each of the functions has 5 instances which differ in rotation and offset. The experiment shall be repeated three times for each instance, which means 15 trials for an algorithm on each function. Since DIRECT is a deterministic algorithm, only 1 trial of each instance was carried out.

An *optimization problem* is defined as a particular (function, requested target value) pair. Each function is used to define several optimization problems differing in the requested target value $f_t = f_{\text{opt}} + \Delta f_t$, where f_{opt} is the optimal function value, and Δf_t is the precision (or tolerance) to reach. The success criterion of a trial (for each optimization problem) is to reach the requested target value f_t . Many precision levels $\Delta f_t \in [10^{-8}, 10^2]$ are defined. If the optimizer solves a function to the ultimate precision value 10^{-8} , it actually solves many optimization problems along the way, and we shall say that it has found the optimum of the function. If the optimizer cannot reach the ultimate precision, it can gain some points for optimizing the function at least partially.

The main performance measure used in the COCO framework is the **Expected Running Time**, ERT (Hansen et al., 2009a; Price, 1997). The ERT estimates the expected number of function evaluations needed to reach the particular target function value if the algorithm is restarted until a single success. The ERT thus depends on the given target function value, f_t , and is computed as “the number of function evaluations conducted in all trials, while the best function value was not smaller than f_t during the trial, divided by the number of trials that actually reached f_t ” (Hansen et al., 2009a).

The results are conveniently presented using the **Empirical Cumulative Distribution Function (ECDF)**. It shows the empirical cumulated probability of success on the problems considered depending on the allocated budget. The ECDF of the ERT is constructed as a bootstrap distribution of the ERT divided by the problem dimension D . In the bootstrapping process, 100 instances of ERT are generated by repeatedly drawing single trials with replacement until a successful trial is drawn for each optimization problem.

Since the ECDF graphs do not express reached function values, but rather the proportion of solved problems, it is possible to meaningfully aggregate the ECDF graphs for several functions of the same class into one graph. The downside of this aggregation is that we are not able to distinguish the individual functions. If a graph shows aggregated ECDFs of 5 functions for certain dimension D , reaching the 20% level of solved problems after n evaluations may mean many things. On the one hand, the algorithm could have found the minimum of one of the five functions, while the other functions may still remain completely unsolved. On the other hand, it may mean that only the problems related to the loose target levels were solved across all the aggregated functions. The latter case is the usual one. If the former explanation is the right one, we will point it out explicitly.

An additional measure used in COCO is the **crafting effort** (Price, 1997; Hoos and Stützle, 1998) that characterizes the parameter tuning effort for an algorithm. The crafting effort should be calculated for each dimension in the following way: $\text{CrE} = -\sum_{k=1}^K \frac{n_k}{n} \ln\left(\frac{n_k}{n}\right)$, where K is the number of different parameter settings, n is the number of functions in the testbed, and each n_k is the number of functions for which the k -th parameter setting was used. The CrE is zero in a given dimension D if the setting was identical for all functions.

4 Algorithm and Experiment Parameter Settings

The following subsections describe the experimental setup and the parameter settings of DIRECT, MCS, and GLOBAL. For the settings of the reference algorithms, we refer the reader to the original reports (Ros, 2009b; Hansen, 2009).

The COCO experimental procedure did not require the algorithms to be tested for $D = 40$. The time and space demands of many MP methods grow quickly with D . Due to practical reasons, the MCS and GLOBAL algorithms were not tested in 40-D.

4.1 DIRECT

The DIRECT algorithm was not restarted; a single run was carried out and stopped after reaching the final precision $\Delta f_t = 10^{-8}$ or after 10^5 function evaluations.

The Jones factor ϵ is the minimal amount of improvement which is considered to be significant by the algorithm. The value was set to $\epsilon = 10^{-10}$.

The maximal depth of the division tree was set to 21. It is roughly equivalent to setting the minimal allowed distance between two neighbouring sampled points (under assumption that the division always takes places along the shortest box side, which is not true). With the maximal depth set to 21, the theoretical minimal distance is of order 10^{-9} , but it is larger in practice.

The initial bounding hypercube was set to $\langle -6, 6 \rangle^D$ despite the fact that all the benchmark functions have the global optimum in $\langle -5, 5 \rangle^D$ (Hansen et al., 2009a). Several of the benchmark functions have the global optimum near (or directly on) the search space boundary. Since DIRECT is pretty bad in approaching such solutions, the larger box was chosen.¹

DIRECT is completely deterministic—only 1 run (instead of 3) for each function instance was carried out. The same parameter settings were used for all experiments on all functions, the crafting effort is $\text{CrE} = 0$ for all D .

4.2 MCS

MCS is equipped with meaningful default values for all parameters. We use the values $s_{\max} = 5D + 10$ (default) for the number of levels, $nf_{\max} = 500 \cdot \max(D, 10)$ for the limit on the overall number of function calls, $ns_{\text{loc}} = nf_{\max}/5$ (much larger than the default value 50) for the limit on the number of iterations in a local search, and reaching the final precision $\Delta f_t = 10^{-8}$ as an additional stopping criterion.

The bounding box is given by $\mathbf{u} = (-5, \dots, -5)^T$ and $\mathbf{v} = -\mathbf{u}$. The default MCS initialization list for finite \mathbf{u} and \mathbf{v} consists of the boundaries and the midpoint, with the midpoint as the starting point. The second initialization list for finite bounds uses $x_i^1 = \frac{5}{6}u_i + \frac{1}{6}v_i$, $x_i^2 = \frac{1}{2}(u_i + v_i)$, $x_i^3 = \frac{1}{6}u_i + \frac{5}{6}v_i$, $i = 1, \dots, D$, and again $\mathbf{x}^0 = \frac{1}{2}(\mathbf{u} + \mathbf{v})$. The third option is to use global line searches along each coordinate, starting from the

¹ It was observed that this choice improved the results e.g. for the linear slope function, while it did not worsen the results on the rest of the functions.

absolutely smallest point in the box, and generate at least three values for each coordinate. We call the MCS algorithm with these three kinds of initialization lists MCS1, MCS2, and MCS3, respectively. A user-defined initialization list is another option. After the initialization list has been chosen, MCS is purely deterministic.

In order to give MCS another chance to solve a problem in the case that the algorithm gets stuck in a nonglobal minimizer, and to introduce a random element in the algorithm at the same time (so that repeating the experiment three times for each instance becomes meaningful), we do not make a single call to MCS with a larger function evaluation budget but instead each experiment consists of up to 10 independent calls to MCS with the above parameters (i.e., each call to MCS does not use any results of the previous calls). First, MCS1, MCS2, and MCS3 are applied to the problem. Then initialization lists with the values $x_i^1 < x_i^2 < x_i^3$ drawn uniformly from $[u_i, v_i]$ for $i = 1, \dots, D$ and $x_i^0 = x_i^2$ are used for at most 7 times for the dimensions $D = 2, 3, 5$ and at most 5 times for the dimensions $D = 10, 20$ (in order to save CPU time).

Since the same parameter settings were used for all experiments on all functions, the crafting effort is $\text{CrE} = 0$ for all D .

4.3 GLOBAL

The COCO framework suggests to compare multistart versions of the base algorithms, i.e. to conduct independent restarts during each trial. However, the GLOBAL algorithm itself is a multistart procedure so that no restarts of GLOBAL were carried out.

GLOBAL has six parameters to set: the number N of points to sample in each iteration, the proportion γ of the best points selected for the reduced sample, the stopping criterion for the local search, the maximum number of function evaluations allowed for local search, the maximum number of local minima to be found (i.e. the maximum number of clusters to be maintained), and the type of the used local search method. All these parameters have a default value and usually it is enough to change only the first three of them.

In all dimensions and for all functions, we sampled $N = 300$ new points²; $\gamma = \frac{2}{N}$, so that the reduced sample contains less than 1 % best points ever sampled.

The following settings were used for $D = 2, 3, 5$. We used the Nelder-Mead simplex method (Nelder and Mead, 1965) implemented in MATLAB by Kelley (1999) as the local search procedure. The termination tolerance parameter TolFun was set to 10^{-8} and the maximum number of function evaluations was equal to 5,000.

For $D = 10, 20$, two different settings were used. For the functions f_3, f_4, f_7, f_{16} , and f_{23} , we used the previous settings with the TolFun parameter set to 10^{-9} . The reason for this choice was that the functions f_7, f_{16} , and f_{23} are not smooth and the BFGS method performs poorer on them. On the functions f_3 and f_4 the simplex method performs slightly better. For the remaining functions, we used the MATLAB `fminunc` function as the local search method using the BFGS update formula with 10,000 as the maximum number of function evaluations and with TolFun set to 10^{-9} . The meaning of the termination tolerance TolFun is different for each of the two local search methods. In case of the Nelder-Mead simplex method it is related to the diameter of the simplex, while in the case of BFGS it relates to the size of the gradient.

The crafting effort $\text{CrE} = 0$ for dimensions 2, 3, and 5. However, for $D = 10, 20$ two different setting were used. The crafting effort can be calculated as $\text{CrE}_{10} = \text{CrE}_{20} =$

² The value was chosen after a preliminary testing. The default value was too small in case of many functions (with many local minimizers) in smaller dimensions. On the other hand, larger values did not improve the results in larger dimensions.

$$-\left(\frac{5}{24} \ln \frac{5}{24} + \frac{19}{24} \ln \frac{19}{24}\right) = 0.5117.$$

5 Results

Results from experiments according to Hansen et al. (2009a) on the benchmark functions (Finck et al., 2009b; Hansen et al., 2009b) are presented in Figures 1 and 2. Only the results for $D = 5$ (exemplar of “low” dimensionality) and $D = 20$ (exemplar of “higher” dimensionality) are presented.

Tables 1 to 10 give the Expected Running Time (ERT) for the target precisions $10^{1, 0, -1, -3, -5, -7}$ divided by the best ERT obtained during BBOB 2009 (given in the ERT_{best} row), together with a measure of its spread (the value typeset in parentheses with a smaller font gives the half of the range between the 10th and 90th percentile). Bold entries correspond to the 3 best values among the algorithms compared. The median number of conducted function evaluations is additionally given in *italics*, if $\text{ERT}(10^{-7}) = \infty$. #succ is the number of trials that reached the final target $f_{\text{opt}} + 10^{-8}$.

The BIPOP-CMA-ES algorithm was used as the baseline for the **statistical comparison** of the other algorithms studied in this article. Each algorithm is tested if it improved the results obtained by BIPOP-CMA-ES. The statistical significance is tested with the rank-sum test for a given target f_t using, for each trial, either the number of needed function evaluations to reach f_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. Entries with the \downarrow symbol are statistically significantly better (according to the rank-sum test) compared to the BIPOP-CMA-ES, with $p = 0.05$ or $p = 10^{-k}$ where $k > 1$ is the number following the \downarrow symbol, with Bonferroni correction of 24.

6 Discussion by Function Group

In this section, the discussion of the results is broken down by function groups. The discussion mostly applies to the presented results for 5-D and 20-D. For a discussion on the individual algorithms, see Sec. 7.

6.1 All functions aggregated

The results for all functions are aggregated in the ECDF graphs of ERT for the 5-D and 20-D functions in Figs. 1 and 2, respectively, in the upper left part.

In the 5-D space, for very low budgets of function evaluations ($\#\text{FEs} < 20D$), NEWUOA and MCS are (close to) the best of all algorithms ever compared using the BBOB methodology. They stay the best among the algorithms compared in this article for $\#\text{FEs} < 200D$. For $200D < \#\text{FEs} < 500D$, GLOBAL takes over solving the highest proportion of the problems. But for budgets larger than $500D$, BIPOP-CMA-ES is the best algorithm, solving almost 100 % of the problems, while the other algorithms solved about 65 % of the problems with GLOBAL being fastest, followed by NEWUOA, MCS and DIRECT.

In the 20-D space, the differences start being more pronounced. For low evaluation budgets ($\#\text{FEs} < 100D$), NEWUOA holds the lead closely followed by MCS. For $100D < \#\text{FEs} < 1000D$, GLOBAL followed by NEWUOA are most successful. And again, for budgets larger than $1000D$, BIPOP-CMA-ES is the best, solving about 92 % of the problems, followed by NEWUOA, GLOBAL, MCS and DIRECT solving about 60, 50, 40, and 20 % of the problems, respectively.

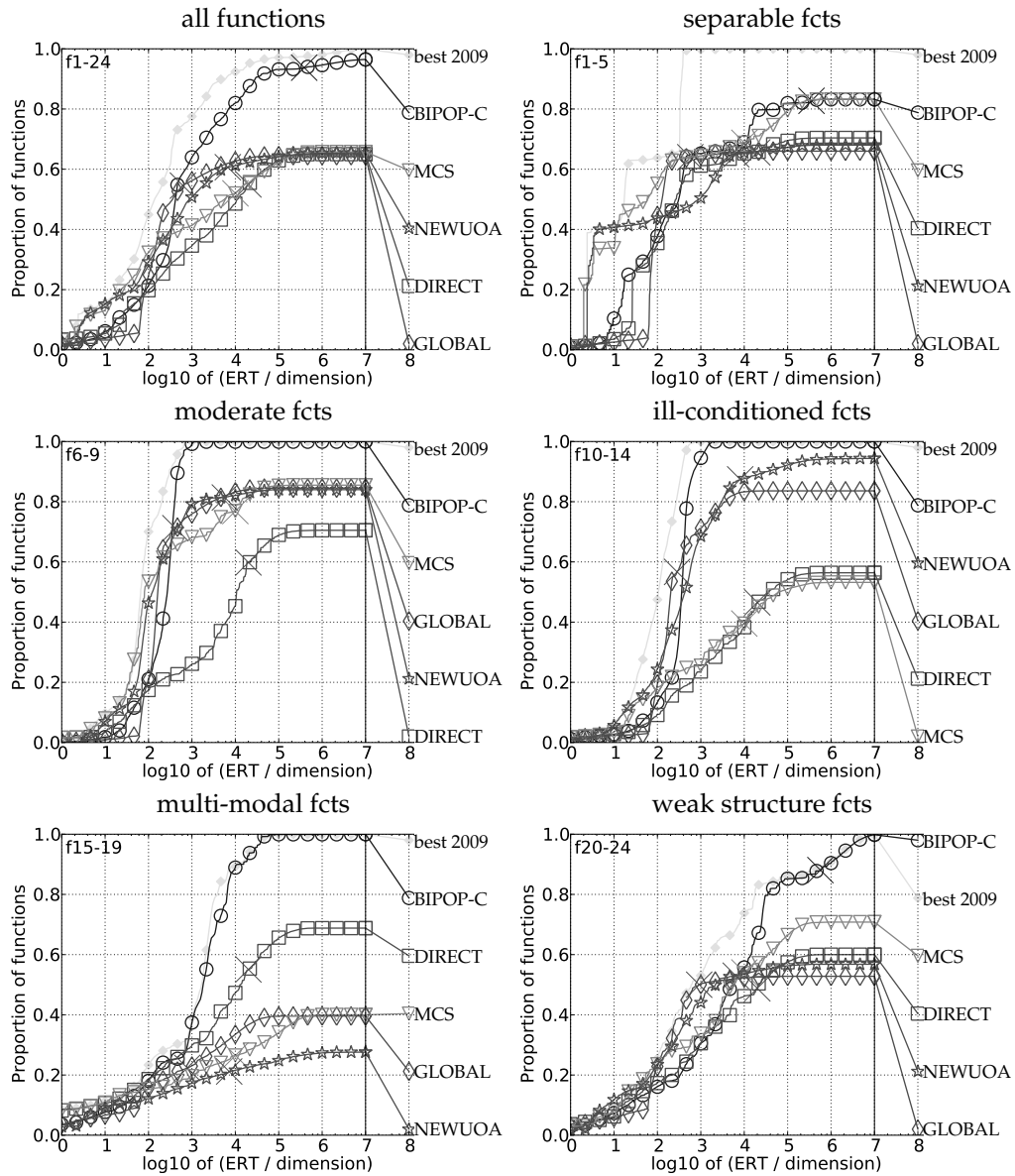


Figure 1: Empirical cumulative distribution of the bootstrapped distribution of ERT over dimension for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 5-D. The best ever line corresponds to the algorithms from BBOB-2009 with the best ERT for each of the targets considered.

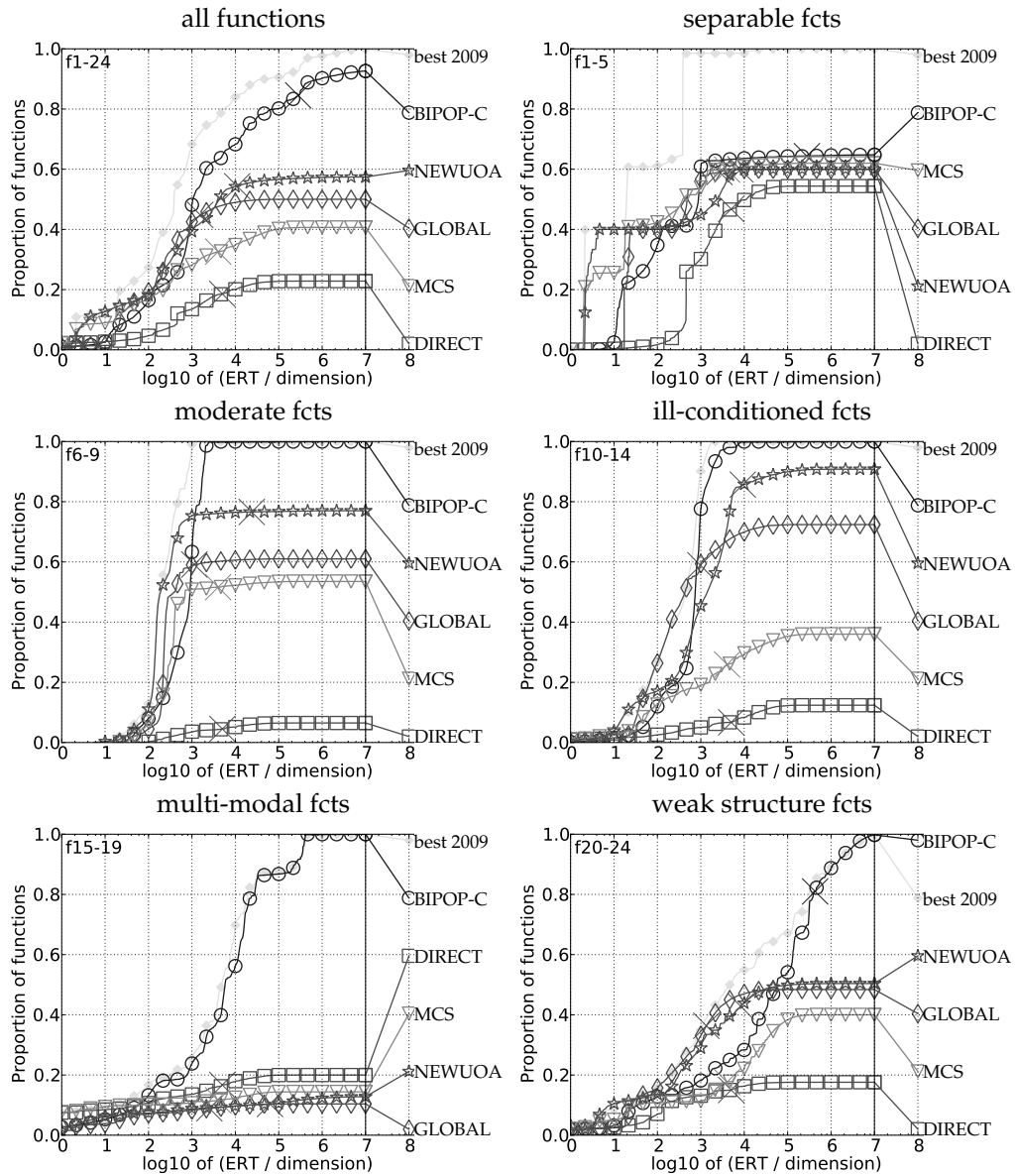


Figure 2: Empirical cumulative distribution of the bootstrapped distribution of ERT over dimension for 50 targets in $10^{[-8..2]}$ for all functions and subgroups in 20-D. The best ever line corresponds to the algorithms from BBOB-2009 with the best ERT for each of the targets considered.

Table 1: ERT on f_1-f_5 in 5-D over ERT_{best} obtained in BBOB-2009

1 Sphere							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	11	12	12	12	12	12	15/15
BIPOP-C	3.2(2.3)	9.0(3.6)	15(3.6)	27(4.8)	40(4.1)	53(5.6)	15/15
GLOBAL	6.8(9.4)	26(0.61)	28(0.74)	32(1.2)	35(1.0)	39(1.4) \downarrow^4	13/15
DIRECT	2.0(1)	7.0(3.4)	19(2.8)	44(11)	84(22)	153(29)	5/5
MCS	1(0.14)	1.8(1.6) \downarrow^3	2.5(1.6) \downarrow^4	2.6(1.6) \downarrow^4	2.6(1.6) \downarrow^4	2.6(1.6) \downarrow^4	15/15
NEWUOA	1.1(0.0)	1(0.0) \downarrow^4	1(0.0) \downarrow^4	1(0.0) \downarrow^4	1(0.0) \downarrow^4	1(0.0) \downarrow^4	15/15
2 Ellipsoid separable							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	83	87	88	90	92	94	15/15
BIPOP-C	13(4.3)	16(3.3)	18(2.1)	20(2.0)	21(1.7)	22(1.8)	15/15
GLOBAL	6.3(1.9) \downarrow^3	6.9(1.9) \downarrow^4	7.3(1.8) \downarrow^4	7.8(1.5) \downarrow^4	8.2(1.4) \downarrow^4	8.5(1.4) \downarrow^4	15/15
DIRECT	5.7(2.8) \downarrow	7.2(3.5) \downarrow	8.4(5.3) \downarrow	14(6.3)	22(12)	381(586)	4/5
MCS	1.1(0.71) \downarrow^4	1.5(1.3) \downarrow^4	2.2(1.2) \downarrow^4	4.7(1.2) \downarrow^4	6.5(1.9) \downarrow^4	29(53)	14/15
NEWUOA	5.7(4.0)	22(16)	45(30)	85(32)	129(33)	166(54)	15/15
3 Rastrigin separable							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	716	1622	1637	1646	1650	1654	15/15
BIPOP-C	1.4(1.4)	16(17)	139(107)	139(107)	139(107)	140(107)	14/15
GLOBAL	3.3(3.5)	∞	∞	∞	∞	∞ 2613	0/15
DIRECT	45(75)	304(308)	∞	∞	∞	∞ 1e5	0/5
MCS	1.2(1.2)	24(32)	216(231)	215(225)	214(239)	214(228)	2/15
NEWUOA	6.1(8.3)	229(257)	∞	∞	∞	∞ 3e4	0/15
4 Skew Rastrigin-Bueche separ							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	809	1633	1688	1817	1886	1903	15/15
BIPOP-C	2.7(3.1)	∞	∞	∞	∞	∞ 2e6	0/15
GLOBAL	8.3(8.7)	∞	∞	∞	∞	∞ 3167	0/15
DIRECT	192(251)	105(123)	249(296)	∞	∞	∞ 1e5	0/5
MCS	4.1(5.6)	∞	∞	∞	∞	∞ 5e4	0/15
NEWUOA	27(26)	305(329)	∞	∞	∞	∞ 3e4	0/15
5 Linear slope							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	10	10	10	10	10	10	15/15
BIPOP-C	4.5(2.0)	6.5(2.6)	6.6(2.4)	6.6(2.4)	6.6(2.4)	6.6(2.4)	15/15
GLOBAL	32(1.3)	33(2.6)	34(2.4)	34(2.4)	34(2.4)	34(2.4)	15/15
DIRECT	9.2(0.05)	12(0.05)	13(0.05)	13(0.05)	13(0.05)	13(0.05)	5/5
MCS	1(0.0) \downarrow^4	1(0.0) \downarrow^4	1(0.0) \downarrow^4	1(0.0) \downarrow^4	1(0.0) \downarrow^4	1(0.0) \downarrow^4	15/15
NEWUOA	1.3(0.15) \downarrow^4	1.5(0.25) \downarrow^4	1.5(0.20) \downarrow^4	1.5(0.25) \downarrow^4	1.5(0.25) \downarrow^4	1.5(0.25) \downarrow^4	15/15

6.2 Separable functions f_1-f_5

The results for the separable functions f_1-f_5 are aggregated in the ECDF graphs of ERT for the 5-D and 20-D functions in Figs. 1 and 2, respectively, in the upper right part. The detailed results are presented in Table 1 for the 5-D functions, and in Table 2 for the 20-D functions.

In the 5-D space, for $\#FEs < 10D$, NEWUOA and MCS are very fast solving about 40% of the problems (these are the problems associated with f_1 , sphere function, and f_5 , linear function), while the other algorithms solved only a few percent. Between $10D-100D$ evaluations, MCS beats NEWUOA reaching 60%, and around $100D$ evaluations, the other methods join the mainstream, solving about 40%. For $100D < \#FEs < 1000D$, MCS, BIPOP-CMA-ES, GLOBAL, and DIRECT showed similar performance by solving about 60% of the problems, while NEWUOA seems to be the least successful here. For budgets larger than 10^4D evaluations, BIPOP-CMA-ES and MCS increase the percentage of the solved problems above 80% (with BIPOP-

Table 2: ERT on f_1-f_5 in 20-D over ERT_{best} obtained in BBOB-2009

1 Sphere							
$\Delta ftarget$	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	43	43	43	43	43	43	15/15
BIPOP-C	7.9 ^(1.7)	14 ^(2.8)	20 ^(2.0)	33 ^(3.5)	45 ^(3.0)	57 ^(3.1)	15/15
GLOBAL	8.0 ^(0.0)	8.0 ^(0.0) ↓ ⁴	8.0 ^(0.0) ↓ ⁴	8.0 ^(0.0) ↓ ⁴	8.0 ^(0.0) ↓ ⁴	8.0 ^(0.0) ↓ ⁴	15/15
DIRECT	48 ⁽¹¹⁾	112 ⁽¹⁵⁾	225 ⁽⁴⁰⁾	485 ⁽³⁸⁾	874 ⁽⁶⁴⁾	1393 ⁽⁹⁸⁾	4/5
MCS	2.4 ^(2.8) ↓ ³	6.4 ^(3.5) ↓ ⁴	6.8 ^(3.5) ↓ ⁴	7.0 ^(3.4) ↓ ⁴	7.0 ^(3.4) ↓ ⁴	7.0 ^(3.4) ↓ ⁴	15/15
NEWUOA	1.0 ^(0.02) ↓ ⁴	1.0 ^(0.01) ↓ ⁴	1.0 ^(0.01) ↓ ⁴	1.0 ^(0.01) ↓ ⁴	1.0 ^(0.01) ↓ ⁴	1.0 ^(0.01) ↓ ⁴	15/15
2 Ellipsoid separable							
$\Delta ftarget$	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	385	386	387	390	391	393	15/15
BIPOP-C	35 ^(6.6)	40 ^(4.4)	44 ^(3.6)	47 ^(2.3)	48 ^(2.3)	50 ^(2.2)	15/15
GLOBAL	18 ^(3.7) ↓ ⁴	23 ^(3.0) ↓ ³	26 ⁽¹³⁾	33 ⁽¹⁴⁾	51 ⁽⁴⁰⁾	63 ⁽⁶⁵⁾	13/15
DIRECT	134 ⁽¹³³⁾	471 ⁽⁵²²⁾	487 ⁽⁵²⁰⁾	537 ⁽⁵¹⁶⁾	∞	∞ 1e5	0/5
MCS	5.4 ^(2.5) ↓ ⁴	14 ^(7.6) ↓ ⁴	21 ⁽¹⁶⁾ ↓ ³	43 ⁽²⁷⁾	45 ⁽²⁶⁾	∞ 8e4	0/15
NEWUOA	18 ^(7.9) ↓ ³	42 ⁽²¹⁾	71 ⁽³⁶⁾	125 ⁽⁴³⁾	174 ⁽⁵¹⁾	219 ⁽⁶⁷⁾	15/15
3 Rastrigin separable							
$\Delta ftarget$	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	5066	7626	7635	7643	7646	7651	15/15
BIPOP-C	12 ^(7.1)	∞	∞	∞	∞	∞ 6e6	0/15
GLOBAL	∞	∞	∞	∞	∞	∞ 5e4	0/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	28 ⁽³²⁾	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	∞	∞	∞	∞	∞	∞ 1e5	0/15
4 Skew Rastrigin-Bueche separ							
$\Delta ftarget$	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	4722	7628	7666	7700	7758	1.41e5	9/15
BIPOP-C	∞	∞	∞	∞	∞	∞ 6e6	0/15
GLOBAL	∞	∞	∞	∞	∞	∞ 8e4	0/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	∞	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	∞	∞	∞	∞	∞	∞ 2e5	0/15
5 Linear slope							
$\Delta ftarget$	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	41	41	41	41	41	41	15/15
BIPOP-C	5.1 ^(0.80)	6.2 ^(1.2)	6.3 ^(1.2)	6.3 ^(1.2)	6.3 ^(1.2)	6.3 ^(1.2)	15/15
GLOBAL	10 ^(0.52)	11 ^(0.78)	11 ^(0.78)	11 ^(0.78)	11 ^(0.78)	11 ^(0.78)	15/15
DIRECT	180 ^(0.01)	224 ^(4.0)	226 ^(0.01)	226 ^(0.01)	226 ^(0.01)	226 ^(0.01)	5/5
MCS	1 ^(0.01) ↓ ⁴	1 ^(0.01) ↓ ⁴	1 ^(0.01) ↓ ⁴	1 ^(0.01) ↓ ⁴	1 ^(0.01) ↓ ⁴	1 ^(0.01) ↓ ⁴	15/15
NEWUOA	1.2 ^(0.10) ↓ ⁴	1.5 ^(0.38) ↓ ⁴	1.6 ^(0.48) ↓ ⁴	1.6 ^(0.48) ↓ ⁴	1.6 ^(0.48) ↓ ⁴	1.6 ^(0.48) ↓ ⁴	15/15

CMA-ES being slightly faster). This increase seems to be caused by solving f_3 , separable Rastrigin function, where GLOBAL, DIRECT and NEWUOA were able to solve only the problems with loose target levels, while BIPOP-CMA-ES and MCS were able to solve even the high precision problems. DIRECT, NEWUOA and GLOBAL eventually reached the level of 65 % of the solved problems. About 15 % of all the separable problems remained unsolved by any of the compared algorithms. To a large extent, these were the problems associated with f_4 , separable multimodal Rastrigin-Büche function.

For the 20-D functions and a low evaluation budget ($\#FEs < 10D$), NEWUOA followed by MCS are the best. Between 20D–100D, NEWUOA, MCS, and GLOBAL are equally good solving 40 % of the problems (again, these are the problems associated with f_1 and f_5). For $100D < \#FEs < 1000D$, MCS, and GLOBAL hold their superiority, but BIPOP-CMA-ES joins them, while NEWUOA slows down. For $\#FEs > 1000D$, BIPOP-CMA-ES wins, solving eventually about 65 % of the problems, followed by MCS, GLOBAL, and NEWUOA (62, 60 and 60) %. About 35 % of the problems remain unsolved, these are associated mainly with the multimodal functions f_3 and f_4 , which

Table 3: ERT on f_6-f_9 in 5-D over ERT_{best} obtained in BBOB-2009

6 Attractive sector							
Δft_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	114	214	281	580	1038	1332	15/15
BIPOP-C	2.3 (1.3)	2.1 (0.65)	2.2 (0.64)	1.7 (0.22)	1.3 (0.28)	1.3 (0.19)	15/15
GLOBAL	2.9 (0.21)	2.1 (0.60)	2.0 (0.50)	2.2 (1.6)	3.6 (3.6)	35 (37)	1/15
DIRECT	2.3 (1.6)	28 (21)	789 (712)	∞	∞	∞ 1e5	0/5
MCS	2.7 (3.9)	47 (117)	41 (93)	71 (86)	46 (56)	54 (64)	4/15
NEWUOA	1.7 (1.5)	2.4 (1.3)	3.6 (2.2)	3.3 (1.7)	2.7 (1.2)	2.9 (1.1)	15/15
7 Step-ellipsoid							
Δft_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	24	324	1171	1572	1572	1597	15/15
BIPOP-C	5.0 (5.0)	1.5 (1.4)	1 (1.2)	1 (0.88)	1 (0.88)	1 (0.86)	15/15
GLOBAL	12 (6.7)	5.7 (5.4)	10 (12)	∞	∞	∞ 1880	0/15
DIRECT	2.8 (1.6)	1.7 (1.1)	115 (146)	∞	∞	∞ 1e5	0/5
MCS	2.8 (4.5)	5.9 (4.3)	13 (21)	∞	∞	∞ 5e4	0/15
NEWUOA	10 (15)	13 (18)	60 (57)	∞	∞	∞ 3e4	0/15
8 Rosenbrock original							
Δft_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	73	273	336	391	410	422	15/15
BIPOP-C	3.2 (1.3)	3.7 (2.3)	4.5 (1.9)	4.8 (1.7)	5.1 (1.7)	5.4 (1.6)	15/15
GLOBAL	5.0 (0.34)	2.1 (1.3) \downarrow	2.1 (1.1) \downarrow 2	2.1 (0.86) \downarrow 3	2.1 (0.86) \downarrow 3	2.2 (0.81) \downarrow 4	15/15
DIRECT	4.1 (2.9)	5.7 (4.8)	22 (15)	100 (94)	195 (185)	293 (288)	3/5
MCS	1.5 (0.91) \downarrow 2	1.0 (1.2) \downarrow 3	1.0 (0.99) \downarrow 4	1.1 (0.89) \downarrow 4	1.1 (0.86) \downarrow 4	1.1 (0.84) \downarrow 4	15/15
NEWUOA	1 (0.95) \downarrow 3	1.1 (0.81) \downarrow 2	1.2 (0.49) \downarrow 3	1.2 (0.44) \downarrow 4	1.2 (0.41) \downarrow 4	1.2 (0.40) \downarrow 4	15/15
9 Rosenbrock rotated							
Δft_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	35	127	214	300	335	369	15/15
BIPOP-C	5.8 (1.8)	8.7 (3.4)	7.2 (2.3)	6.4 (1.5)	6.3 (1.3)	6.2 (1.2)	15/15
GLOBAL	11 (2.2)	4.6 (1.3)	3.2 (0.80) \downarrow 2	2.8 (0.74) \downarrow 2	2.7 (0.60) \downarrow 3	2.7 (1.2) \downarrow 3	13/15
DIRECT	3.2 (2.2)	4.2 (2.3)	48 (38)	134 (87)	151 (79)	306 (300)	3/5
MCS	1 (0.55) \downarrow 4	1 (0.23) \downarrow 4	1 (0.16) \downarrow 4	1 (0.16) \downarrow 4	1 (0.19) \downarrow 4	1 (0.12) \downarrow 4	15/15
NEWUOA	1.8 (0.74) \downarrow 4	3.6 (2.8)	2.5 (1.6) \downarrow 2	1.9 (1.1) \downarrow 3	1.9 (1.0) \downarrow 3	1.7 (0.92) \downarrow 4	15/15

present considerable problems to any algorithm that does not assume separability.

6.3 Unimodal functions with moderate conditioning f_6-f_9

The results for the unimodal functions with moderate conditioning f_6-f_9 are aggregated in the ECDF graphs of ERT for the 5-D and 20-D functions in Figs. 1 and 2, respectively, in the middle left part. The detailed results are presented in Table 3 for the 5-D functions, and in Table 4 for the 20-D functions.

For the 5-D problems, MCS and NEWUOA are the most successful algorithms until $100D$ evaluations. They gained their superiority by efficiently solving f_8 and f_9 , the original and rotated Rosenbrock functions. For $\#FEs < 50D$, DIRECT is third behind them, while GLOBAL solves only a very low proportion of the problems (due to the initial random phase). Around $100D$ evaluations, GLOBAL makes a huge progress (by executing the local search, Nelder-Mead in case of 5-D) and joins the leading group. For a short range of evaluation budgets, it becomes the best competitor. DIRECT slows down solving only about 20% of the problems where the others already solved around 60%. For $\#FEs > 500D$, BIPOP-CMA-ES clearly wins (solving eventually all problems, for which it needed about $1000D$ evaluations). MCS, GLOBAL, and NEWUOA solved about 85% of the problems. The problems they did not solve are associated with f_7 , step-ellipsoid function, where they were not able to find the target levels 10^{-3} and tighter. DIRECT was the slowest and least successful solving only 70% of the problems. Additionally to f_7 , it also did not solve the f_6 , attractive sector function.

Table 4: ERT on f_6 - f_9 in 20-D over ERT_{best} obtained in BBOB-2009

6 Attractive sector							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	1296	2343	3413	5220	6728	8409	15/15
BIPOP-C	1.5 (0.38)	1.3 (0.25)	1.2 (0.23)	1.1 (0.18)	1.2 (0.12)	1.2 (0.10)	15/15
GLOBAL	3.6 (1.00)	3.6 (0.74)	6.1 (3.0)	∞	∞	∞ 4e4	0/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	∞	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	1 (0.33)	1 (0.37)	1 (0.49)	1.1 (0.54)	1.3 (0.83)	1.3 (0.66)	15/15
7 Step-ellipsoid							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	1351	4274	9503	16524	16524	16969	15/15
BIPOP-C	1 (0.47)	4.9 (2.5)	3.5 (0.60)	2.2 (0.26)	2.2 (0.26)	2.1 (0.25)	15/15
GLOBAL	∞	∞	∞	∞	∞	∞ 1e4	0/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	∞	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	∞	∞	∞	∞	∞	∞ 5e5	0/15
8 Rosenbrock original							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	2039	3871	4040	4219	4371	4484	15/15
BIPOP-C	4.0(1.0)	4.0(0.68)	4.3(0.62)	4.5(0.60)	4.6(0.56)	4.6(0.56)	15/15
GLOBAL	1.6 (0.32) \downarrow^4	1.2 (0.16) \downarrow^4	1.2 (0.16) \downarrow^4	1.2 (0.15) \downarrow^4	1.2 (0.14) \downarrow^4	1.2 (0.14) \downarrow^4	15/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	1.5 (0.78) \downarrow^4	1.7 (1.4) \downarrow^3	1.8 (1.4) \downarrow^3	1.8 (1.4) \downarrow^3	1.8 (1.3) \downarrow^3	1.8 (1.3) \downarrow^3	15/15
NEWUOA	1 (0.26) \downarrow^4	1 (0.59) \downarrow^4	1 (0.56) \downarrow^4	1 (0.54) \downarrow^4	1 (0.52) \downarrow^4	1 (0.50) \downarrow^4	15/15
9 Rosenbrock rotated							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	1716	3102	3277	3455	3594	3727	15/15
BIPOP-C	4.7(1.7)	5.7(1.1)	6.0(1.1)	6.1(1.0)	6.1(1.00)	6.1(0.93)	15/15
GLOBAL	1.7 (0.28) \downarrow^4	1.7 (0.89) \downarrow^4	1.6 (0.84) \downarrow^4	1.6 (0.79) \downarrow^4	1.6 (0.77) \downarrow^4	1.5 (0.74) \downarrow^4	15/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	1 (0.48) \downarrow^4	1.3 (0.68) \downarrow^4	1.5 (0.70) \downarrow^4	1.6 (0.73) \downarrow^4	1.7 (0.71) \downarrow^4	1.6 (0.68) \downarrow^4	15/15
NEWUOA	1.0 (0.16) \downarrow^4	1 (0.60) \downarrow^4	1 (0.57) \downarrow^4	1 (0.54) \downarrow^4	1 (0.52) \downarrow^4	1 (0.50) \downarrow^4	15/15

In the 20-D space, for low evaluation budgets ($\#FEs < 100D$), NEWUOA starts successfully by solving about 10 % of the problems, followed by BIPOP-CMA-ES, MCS, and GLOBAL (all between 5 and 10 %). DIRECT solved virtually no problem so far. For $100D < \#FEs < 1000D$, NEWUOA stays in the lead, solving more than 75 % of the problems, with GLOBAL and MCS behind (60 % and over 50 %, respectively). For larger budgets ($\#FEs > 1000D$), NEWUOA, GLOBAL, and MCS did not improve their results significantly. Around $1000D$ evaluations, the proportion of the problems solved by BIPOP-CMA-ES rises steeply and reaches 100 % after $2000D$ evaluations. DIRECT solved less than 10 % of the problems. All algorithms (with the exception of DIRECT) are pretty efficient at solving the Rosenbrock functions, f_8 and f_9 , thanks to the local search routines they use. The step-ellipsoid function, f_7 , is even harder in 20-D, since only BIPOP-CMA-ES was able to solve the function to all reported target levels. The attractive sector function, f_6 , was successfully solved by BIPOP-CMA-ES and NEWUOA only, GLOBAL was able to find target levels until 10^{-1} , while DIRECT and MCS did not find any reported target level.

6.4 Unimodal ill-conditioned functions f_{10} - f_{14}

The results for the unimodal ill-conditioned functions f_{10} - f_{14} are aggregated in the ECDF graphs of ERT for the 5-D and 20-D functions in Figs. 1 and 2, respectively, in the middle right part. The detailed results are presented in Table 5 for the 5-D functions, and in Table 6 for the 20-D functions.

Table 5: ERT on $f_{10}-f_{14}$ in 5-D over ERT_{best} obtained in BBOB-2009

10 Ellipsoid							
Δft_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	349	500	574	626	829	880	15/15
BIPOP-C	3.5 _(0.82)	2.9 _(0.43)	2.7 _(0.37)	2.8 _(0.25)	2.3 _(0.18)	2.4 _(0.14)	15/15
GLOBAL	1.9 _(0.70) ^{↓3}	1.6 _(0.49) ^{↓3}	1.8 _(0.71)	2.0 _(1.5)	1.7 _(1.1)	1.7 _(1.1)	15/15
DIRECT	110 ₍₁₅₈₎	141 ₍₁₈₁₎	281 ₍₃₅₇₎	∞	∞	∞ <i>1e5</i>	0/5
MCS	277 ₍₃₂₀₎	∞	∞	∞	∞	∞ <i>5e4</i>	0/15
NEWUOA	3.1 _(3.3)	5.5 _(4.4)	8.1 _(6.7)	14 _(8.5)	16 _(7.5)	21 _(7.5)	15/15
11 Discus							
Δft_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	143	202	763	1177	1467	1673	15/15
BIPOP-C	8.4 _(3.0)	7.2 _(1.6)	2.2 _(0.30)	1.6 _(0.20)	1.4 _(0.11)	1.3 _(0.11)	15/15
GLOBAL	4.0 _(1.5) ^{↓3}	5.5 _(2.6)	3.5 _(3.2)	5.0 _(6.1)	5.0 _(6.5)	8.5 _(7.9)	8/15
DIRECT	87 ₍₁₆₂₎	2228 ₍₂₄₇₈₎	∞	∞	∞	∞ <i>1e5</i>	0/5
MCS	82 ₍₁₀₆₎	461 ₍₅₀₁₎	∞	∞	∞	∞ <i>5e4</i>	0/15
NEWUOA	3.5 _(1.8) ^{↓3}	4.7 _(2.2) [↓]	1.8 _(0.63)	1.8 _(0.44)	2.0 _(0.35)	2.2 _(0.41)	15/15
12 Bent cigar							
Δft_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	108	268	371	461	1303	1494	15/15
BIPOP-C	11 ₍₁₂₎	7.4 _(7.8)	7.4 _(5.7)	7.7 _(5.4)	3.3 _(2.2)	3.3 _(2.1)	15/15
GLOBAL	4.6 _(1.2)	2.7 _(0.61) [↓]	2.4 _(0.82) ^{↓3}	5.0 _(6.7) [↓]	3.1 _(4.0) [↓]	3.4 _(4.0)	6/15
DIRECT	8.5 _(1.1)	8.7 _(5.7)	19 ₍₁₇₎	108 ₍₆₂₎	377 ₍₃₈₄₎	∞ <i>1e5</i>	0/5
MCS	1 _(0.20) ^{↓4}	18 ₍₄₇₎	17 ₍₃₃₎	22 ₍₃₂₎	26 ₍₃₈₎	56 ₍₆₉₎	6/15
NEWUOA	3.5 _(3.3) [↓]	2.6 _(2.6) [↓]	2.5 _(2.3) ^{↓2}	2.6 _(2.3) ^{↓2}	1.1 _(1.0) ^{↓2}	1.1 _(0.99) [↓]	15/15
13 Sharp ridge							
Δft_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	132	195	250	1310	1752	2255	15/15
BIPOP-C	3.9 _(3.3)	5.4 _(3.5)	5.9 _(2.6)	1.6 _(0.26)	1.5 _(0.22)	1.7 _(0.83)	15/15
GLOBAL	4.2 _(2.5)	6.1 _(4.8)	11 ₍₁₁₎	∞	∞	∞ <i>1262</i>	0/15
DIRECT	7.0 _(5.2)	21 ₍₁₅₎	34 ₍₁₅₎	42 ₍₄₆₎	119 ₍₁₁₅₎	∞ <i>1e5</i>	0/5
MCS	41 ₍₅₀₎	214 ₍₂₁₈₎	462 ₍₅₇₀₎	550 ₍₆₁₁₎	∞	∞ <i>5e4</i>	0/15
NEWUOA	3.1 _(3.3)	9.3 ₍₁₂₎	35 ₍₃₆₎	54 ₍₅₄₎	335 ₍₃₆₇₎	∞ <i>4e4</i>	0/15
14 Sum of different powers							
Δft_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	10	41	58	139	251	476	15/15
BIPOP-C	1.1 _(0.97)	2.8 _(1.3)	3.7 _(0.91)	4.6 _(0.73)	5.4 _(0.55)	4.5 _(0.34)	15/15
GLOBAL	2.2 _(1.9)	7.7 _(0.22)	5.9 _(0.28)	3.3 _(0.40) ^{↓2}	3.6 _(2.5) ^{↓3}	∞ <i>1302</i>	0/15
DIRECT	1 _(1.4)	3.7 _(3.1)	4.8 _(3.5)	23 ₍₂₅₎	1892 ₍₁₉₉₀₎	∞ <i>1e5</i>	0/5
MCS	1.4 _(2.3)	2.8 _(3.3)	2.7 _(2.3)	2.8 _(1.1) ^{↓3}	225 ₍₂₉₉₎	∞ <i>5e4</i>	0/15
NEWUOA	1.7 _(0.56)	1 _(0.35) ^{↓4}	1 _(0.27) ^{↓4}	1.2 _(0.31) ^{↓4}	5.5 _(1.9)	2525 ₍₂₉₆₁₎	0/15

Compared to the results of functions with moderate conditioning, on the ill-conditioned functions in the 5-D space the situation is quite similar with the exceptions that MCS and DIRECT have worse performance, while NEWUOA improved its results. For budgets lower than $100D$ evaluations, NEWUOA and MCS are in the lead having solved about 20 % of the problems. Around $100D$ evaluations, the performance of MCS starts to decline, while the performance of GLOBAL and BIPOP-CMA-ES rises steeply. For a short range of evaluation budgets, GLOBAL is the best competitor. It reaches the level of 50 % twice as fast as NEWUOA or BIPOP-CMA-ES. For $\#FEs > 500D$, BIPOP-CMA-ES takes over and solves all the problems needing slightly above $1000D$ evaluations (behavior very similar to functions with moderate conditioning). NEWUOA reaches almost 95 % of the solved problems, GLOBAL 85 %, DIRECT and MCS around 55 %. DIRECT and MCS fail significantly on f_{10} , ellipsoid, and f_{11} , discus, functions. Note that on f_{10} , DIRECT can find the target level 10^{-1} , while MCS finds only 10^1 , because the local search procedure employed by MCS cannot solve this function due to the problems with high-conditioning (see also Sec. 7.2). GLOBAL, on the other hand,

Table 6: ERT on f_{10} - f_{14} in 20-D over ERT_{best} obtained in BBOB-2009

10 Ellipsoid							
Δ ftarget ERT _{best}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
BIPOP-C	1.9 (0.21)	1.8 (0.17)	1.6 (0.13)	1.2 (0.03)	1.1 (0.03)	1.1 (0.03)	15/15
GLOBAL	1 (0.22) ^{↓4}	1.1 (0.15)	1.1 (0.53)	2.0 (1.7)	5.9 (6.2)	∞ 4e4	0/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	∞	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	1.7 (0.50)	2.6 (0.78)	3.3 (1.1)	4.0 (0.83)	4.7 (0.76)	5.8 (1.0)	15/15
11 Discus							
Δ ftarget ERT _{best}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
BIPOP-C	10 (0.54)	5.1 (0.25)	1.9 (0.08)	1.4 (0.05)	1.2 (0.04)	1.0 (0.03)	15/15
GLOBAL	1.2 (0.42) ^{↓4}	1.0 (0.60) ^{↓4}	1 (0.84)	∞	∞	∞ 3e4	0/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	∞	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	15 (2.5)	13 (2.0)	5.8 (0.55)	6.1 (0.47)	6.6 (0.32)	6.5 (0.29)	15/15
12 Bent cigar							
Δ ftarget ERT _{best}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
BIPOP-C	3.0 (1.8)	4.0 (3.2)	4.5 (3.3)	4.5 (2.1)	1.9 (0.69)	2.0 (0.66)	15/15
GLOBAL	1 (0.85) ^{↓2}	1 (0.88) [↓]	1 (0.70) ^{↓2}	1 (0.49) ^{↓3}	1.1 (1.1) ^{↓3}	3.4 (3.2)	0/15
DIRECT	421(481)	235(258)	∞	∞	∞	∞ 1e5	0/5
MCS	1.1 (0.78) [↓]	8.4(18)	12(15)	43(43)	94(101)	∞ 8e4	0/15
NEWUOA	3.0 (2.9)	3.0 (2.4)	3.0 (1.7)	2.5 (1.2)	1 (0.42) ^{↓2}	1 (0.37) ^{↓3}	15/15
13 Sharp ridge							
Δ ftarget ERT _{best}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
BIPOP-C	4.3 (5.6)	2.7 (2.1)	5.1 (5.6)	1.5 (0.77)	2.3 (2.3)	3.0 (1.9)	15/15
GLOBAL	2.0 (0.34) ^{↓2}	1.1 (0.08)	1.1 (0.04)	4.5 (5.1)	∞	∞ 2e4	0/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	34(46)	37(30)	61(60)	∞	∞	∞ 8e4	0/15
NEWUOA	1 (1.4) ^{↓2}	3.0 (4.9)	9.3 (12)	19 (21)	∞	∞ 2e5	0/15
14 Sum of different powers							
Δ ftarget ERT _{best}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
BIPOP-C	3.9 (1.2)	2.9 (0.44)	3.7 (0.42)	4.1 (0.32)	6.2 (0.49)	1.2 (0.09)	15/15
GLOBAL	5.0 (0.28)	2.2 (0.22) ^{↓3}	2.1 (0.21) ^{↓4}	1.1 (0.08) ^{↓4}	1 (0.04) ^{↓4}	∞ 8538	0/15
DIRECT	8.4(6.0)	153(213)	290(380)	∞	∞	∞ 1e5	0/5
MCS	1 (0.54) ^{↓4}	2.1 (0.78) [↓]	3.4 (0.87)	3.2 (0.87) [↓]	∞	∞ 8e4	0/15
NEWUOA	1.5 (0.75) ^{↓3}	1 (0.32) ^{↓4}	1 (0.28) ^{↓4}	1 (0.18) ^{↓4}	9.1 (0.95)	43 (32)	0/15

has the biggest problems with f_{13} , sharp ridge, due to the inability of the GLOBAL's local search procedure (Nelder-Mead in case of 5-D) to make progress on this function. The tiny proportion of the problems not solved by NEWUOA consists of the problems with tight target levels ($< 10^{-5}$) associated mainly with f_{13} , sharp ridge.

In all previous cases, MCS belongs among 2 best algorithms with respect to the initial stage of the search. For the 20-D ill-conditioned functions, the initial performance of MCS is not that good. For $\#FEs < 40D$, the NEWUOA is the only algorithm having solved about 10 % of the problems, while the others solved only a few percent so far. For $40D < \#FEs < 400D$ range, GLOBAL is the most successful algorithm having solved about 60 % of the problems, followed by NEWUOA, BIPOP-CMA-ES, and MCS (all between 20 and 30 %). For budgets above 1000D evaluations, BIPOP-CMA-ES is again the best algorithm reaching 100 % of the solved problems (again, it has behavior similar to behavior on moderate functions). NEWUOA solved more than 90 % of the problems, beating GLOBAL by 20 %. MCS (37 %) and DIRECT (12 %) are not successful solvers for this class of the problems. NEWUOA has slight difficulties solving f_{13} , sharp ridge,

to the target levels tighter than 10^{-3} . On the other hand, among all BBOB competitors it is the fastest algorithm that solves f_{12} , bent cigar, for the tightest target levels below 10^{-3} , and for the middle target levels between 10^0 and 10^{-5} for f_{14} , sum of different powers. GLOBAL fails for the tightest target levels for the functions f_{10} , f_{11} , f_{13} , and f_{14} , but is the fastest for the loose target levels of f_{12} . Overall, it belongs to the fastest algorithms for this class of functions; if it can find the desired target level, it finds it quickly. This can be attributed to its local search procedure, BFGS, which is a fast local search method in itself. Based on the results of the multistart BFGS (Ros, 2009a), which are not shown here, it can be stated that on the moderate and ill-conditioned functions BFGS itself is equally good or better than the GLOBAL algorithm. In both methods, however, some parameters are set differently.

6.5 Multimodal functions f_{15} – f_{19}

The results for the multimodal functions f_{15} – f_{19} are aggregated in the ECDF graphs of ERT for the 5-D and 20-D functions in Figs. 1 and 2, respectively, in the bottom left part. The detailed results are presented in Table 7 for the 5-D functions, and in Table 8 for the 20-D functions.

In the 5-D space, the performance of all algorithms in the beginning of the search is quite similar. For $100D < \#FEs < 1000D$, BIPOP-CMA-ES and DIRECT solve the highest proportion of the problems (around 30%). The performance of the other algorithms starts to decline. Eventually, BIPOP-CMA-ES solves all the problems (needing about 10^5D evaluations). DIRECT solved almost 70% of the problems, MCS and GLOBAL about 40%, and NEWUOA below 30%.

For the 20-D functions, the performance of all algorithms dropped significantly with the exception of BIPOP-CMA-ES. For low evaluation budgets ($\#FEs < 100D$), MCS provides the best results (about 10%) closely followed by DIRECT. Around $100D$ evaluations, BIPOP-CMA-ES starts to dominate, eventually solving all the functions, needing $4 \cdot 10^5D$ evaluations. The other algorithms are with a huge gap behind: DIRECT solved about 20%, MCS 15%, NEWUOA 12%, and GLOBAL around 10%.

Regarding the performance of the algorithms for individual functions in this class, the relations are not so obvious. Leaving out the BIPOP-CMA-ES, which solved all the functions, the performance of DIRECT in 5-D is quite surprising. Launching a local search procedure for low-dimensional versions of these functions seems to be only a waste of resources for this class of the problems (see Sec. 7.2 for a supporting argument). For 20-D, DIRECT, MCS, GLOBAL, and NEWUOA are only able to find the loose target levels ($> 10^{-1}$) which are obviously not very hard to find. The functions in this class have some global structure which can be exploited if the algorithm uses some smoothing mechanism. This is the case for BIPOP-CMA-ES and we conjecture it is the reason for its superior behavior.

6.6 Multimodal functions with weak structure f_{20} – f_{24}

The results for the multimodal functions f_{20} – f_{24} are aggregated in the ECDF graphs of ERT for the 5-D and 20-D functions in Figs. 1 and 2, respectively, in the bottom right part. The detailed results are presented in Table 9 for the 5-D functions, and in Tables 10 for the 20-D functions.

For the 5-D functions, until $100D$ evaluations NEWUOA and MCS are in the lead, DIRECT and BIPOP-CMA-ES have a similar performance, and GLOBAL is the least successful because of its initial random phase. After $100D$ evaluations, all algorithms solved approx. 20% of the problems. For $100D < \#FEs < 3 \cdot 10^3D$, GLOBAL is the

Table 7: ERT on $f_{15}-f_{19}$ in 5-D over ERT_{best} obtained in BBOB-2009

15 Rastrigin							
$\Delta ftarget$	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	511	9310	19369	20073	20769	21359	14/15
BIPOP-C	1.6 (2.2)	1.5 (1.5)	1.2 (0.68)	1.2 (0.67)	1.2 (0.66)	1.2 (0.65)	15/15
GLOBAL	6.0(7.0)	∞	∞	∞	∞	∞ 2659	0/15
DIRECT	5.4 (4.1)	9.4 (12)	∞	∞	∞	∞ 1e5	0/5
MCS	4.0 (4.1)	25 (27)	38 (40)	37 (40)	36 (39)	∞ 5e4	0/15
NEWUOA	5.8(5.7)	41(44)	∞	∞	∞	∞ 3e4	0/15
16 Weierstrass							
$\Delta ftarget$	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	120	612	2663	10449	11644	12095	15/15
BIPOP-C	3.0(2.6)	3.6 (2.7)	2.6 (1.4)	1.3 (1.5)	1.4 (1.6)	1.4 (1.7)	15/15
GLOBAL	1.4 (1.3)	1 (0.53) \downarrow	1 (1.2)	3.5 (4.1)	6.8 (7.9)	6.6 (7.4)	0/15
DIRECT	1.2 (0.65)	1.6 (0.66)	3.4 (3.5)	5.9 (7.3)	19 (17)	40 (37)	1/5
MCS	1.9 (2.6)	18(23)	131(149)	∞	∞	∞ 5e4	0/15
NEWUOA	2.1 (1.8)	29(23)	∞	∞	∞	∞ 4e4	0/15
17 Schaffer F7, condition 10							
$\Delta ftarget$	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	5.2	215	899	3669	6351	7934	15/15
BIPOP-C	3.4(3.1)	1 (0.22)	1 (2.0)	1 (0.71)	1 (0.51)	1.2 (0.46)	15/15
GLOBAL	3.5(3.1)	5.0 (4.0)	∞	∞	∞	∞ 3140	0/15
DIRECT	1 (1.3)	1.4 (1.0)	1.7 (0.71)	4.4 (5.1)	9.5 (10)	10 (7.6)	1/5
MCS	1.9 (1.8)	24(24)	63(71)	∞	∞	∞ 5e4	0/15
NEWUOA	2.3 (1.5)	40(47)	617(668)	∞	∞	∞ 3e4	0/15
18 Schaffer F7, condition 1000							
$\Delta ftarget$	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	103	378	3968	9280	10905	12469	15/15
BIPOP-C	1 (0.69)	3.4 (5.3)	1 (1.2)	1 (0.35)	1.2 (0.75)	1.3 (0.63)	15/15
GLOBAL	3.9 (1.7)	15 (15)	14 (13)	∞	∞	∞ 2633	0/15
DIRECT	1.4 (0.82)	2.9 (2.9)	1.9 (1.7)	6.5 (6.4)	∞	∞ 1e5	0/5
MCS	19(25)	154(151)	∞	∞	∞	∞ 5e4	0/15
NEWUOA	31(28)	1351(1802)	∞	∞	∞	∞ 9e4	0/15
19 Griewank-Rosenbrock F8F2							
$\Delta ftarget$	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	1	1	242	1.20e5	1.21e5	1.22e5	15/15
BIPOP-C	20(16)	2801 (5071)	161 (175)	1 (0.69)	1 (0.69)	1 (0.68)	15/15
GLOBAL	46(44)	7329(8221)	∞	∞	∞	∞ 4311	0/15
DIRECT	1 (0.0) \downarrow	1 (0.0) \downarrow	1.1 (0.71) \downarrow	3.7 (4.2)	∞	∞ 1e5	0/5
MCS	1 (0.0) \downarrow^4	1 (0.0) \downarrow^4	1 (0.44) \downarrow^4	∞	∞	∞ 5e4	0/15
NEWUOA	14 (2.5)	2.7e4 (2.4e4)	1415(1927)	∞	∞	∞ 5e5	0/15

most successful algorithm reaching about 50 % of the solved problems and is very close to the best BBOB-2009 competitor. For budgets larger than 10^4D , BIPOP-CMA-ES leads and eventually reaches 100 % needing at least 10^6D evaluations. MCS placed second with 70 % of the solved problems. DIRECT, NEWUOA, and GLOBAL solved 60, 56, and 53 %, respectively.

Both Gallagher's functions, f_{21} and f_{22} , were solved by all algorithms. For f_{23} , Katsuura's, and f_{24} , Lunacek bi-Rastrigin, all the algorithms (except BIPOP-CMA-ES) were only able to find loose target levels $> 10^{-3}$. MCS beats GLOBAL, DIRECT, and NEWUOA thanks to f_{20} , Schwefel's function, which is only solved by BIPOP-CMA-ES and MCS.

In the 20-D space, around $10D$ evaluations, NEWUOA is the most successful algorithm with 10 % already solved, while the other algorithms are below 5 %. For $100D < \#FEs < 2 \cdot 10^4D$, GLOBAL solves the largest proportion of the problems reaching about 50 % and is the best BBOB-2009 algorithm for budgets between $100D$ and $2000D$ evaluations. NEWUOA closely follows GLOBAL. BIPOP-CMA-ES, MCS,

Table 8: ERT on $f_{15}-f_{19}$ in 20-D over ERT_{best} obtained in BBOB-2009

15 Rastrigin							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT _{best}	30378	1.47e5	3.12e5	3.20e5	4.49e5	4.59e5	15/15
BIPOP-C	1 _(0.42)	2.0 _(0.76)	1.4 _(0.49)	1.4 _(0.48)	1 _(0.35)	1 _(0.34)	15/15
GLOBAL	∞	∞	∞	∞	∞	∞ 2e4	0/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	∞	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	∞	∞	∞	∞	∞	∞ 1e5	0/15
16 Weierstrass							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT _{best}	1384	27265	77015	1.88e5	1.98e5	2.20e5	15/15
BIPOP-C	1.7 _(0.42)	1.0 _(0.70)	1.2 _(0.74)	1 _(0.72)	1 _(0.70)	1 _(0.70)	15/15
GLOBAL	1 _(0.72)	∞	∞	∞	∞	∞ 1e4	0/15
DIRECT	8.3 ₍₁₇₎	7.0 _(7.6)	∞	∞	∞	∞ 1e5	0/5
MCS	11 ₍₁₅₎	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	16 ₍₁₇₎	∞	∞	∞	∞	∞ 2e5	0/15
17 Schaffer F7, condition 10							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT _{best}	63	1030	4005	30677	56288	80472	15/15
BIPOP-C	2.2 _(1.7)	1 _(0.31)	1 _(1.5)	1.2 _(1.1)	1.3 _(0.58)	1.4 _(0.72)	15/15
GLOBAL	6.2 _(1.2)	∞	∞	∞	∞	∞ 7e4	0/15
DIRECT	1.8 _(2.2)	55 ₍₅₈₎	∞	∞	∞	∞ 1e5	0/5
MCS	1 _(1.2)	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	16 _(4.0)	∞	∞	∞	∞	∞ 2e6	0/15
18 Schaffer F7, condition 1000							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT _{best}	621	3972	19561	67569	1.31e5	1.47e5	15/15
BIPOP-C	1.0 _(0.36)	2.4 _(2.4)	1.2 _(0.89)	1.1 _(0.56)	1.7 _(0.67)	1.6 _(0.62)	15/15
GLOBAL	∞	∞	∞	∞	∞	∞ 7e4	0/15
DIRECT	9.1 _(8.4)	113 ₍₁₂₆₎	∞	∞	∞	∞ 1e5	0/5
MCS	∞	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	1.2e4 _(1.3e4)	∞	∞	∞	∞	∞ 2e6	0/15
19 Griewank-Rosenbrock F8F2							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT _{best}	1	1	3.43e5	6.22e6	6.69e6	6.74e6	15/15
BIPOP-C	169 ₍₇₄₎	2.4e4 _(1.2e4)	1.2 _(0.60)	1 _(0.32)	1 _(0.28)	1 _(0.28)	15/15
GLOBAL	5601 ₍₃₅₃₁₎	∞	∞	∞	∞	∞ 5e4	0/15
DIRECT	1 _(0.0) ↓	1 _(0.0) ↓	∞	∞	∞	∞ 1e5	0/5
MCS	1 _(0.0) ↓4	1 _(0.0) ↓4	1 _(1.2)	∞	∞	∞ 8e4	0/15
NEWUOA	76 ₍₅₁₎ ↓2	4.3e6 _(5.2e6)	∞	∞	∞	∞ 2e6	0/15

and DIRECT have a gap of about 20 to 40 %. For #FEs > 2 · 10⁴D, BIPOP-CMA-ES wins solving 100 % of the problems, NEWUOA beats GLOBAL by a few percent both solving around 50 % of the problems. MCS with 40 % is a way in front of DIRECT with only 20 % of success.

DIRECT can solve only the loose target level problems for f_{21} , f_{22} , and f_{23} . MCS solved f_{21} , but for f_{22} and f_{23} it only reached the targets 10⁻³ and above. GLOBAL and NEWUOA solved very quickly both Gallagher’s functions, f_{21} and f_{22} . (The shape of the peaks in these functions is very suitable for the local search engine of GLOBAL (BFGS) and for NEWUOA.) They were only partially successful in solving the rest of the functions.

7 Discussion by Algorithm

In this section, we look at the results from the point of view of individual algorithms. A global view of the algorithm results is presented in Table 11. The table items describe the number of functions for which we are able to compute a finite ERT, i.e. the number

Table 9: ERT on $f_{20}-f_{24}$ in 5-D over ERT_{best} obtained in BBOB-2009

20 Schwefel $x*\sin(x)$							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	16	851	38111	54470	54861	55313	14/15
BIPOP-C	3.3 (2.7)	8.2(10)	2.8 (1.0)	2.1 (0.81)	2.2 (0.82)	2.2 (0.82)	15/15
GLOBAL	17(4.9)	18(19)	∞	∞	∞	∞ 2276	0/15
DIRECT	3.8(0.03)	1.5 (5.9e-4)	∞	∞	∞	∞ 1e5	0/5
MCS	2.7 (1.6)	1 (1.8) \downarrow^2	9.1 (10)	6.4 (6.9)	6.4 (7.2)	6.3 (7.1)	2/15
NEWUOA	1(0.25)	3.3 (3.6)	∞	∞	∞	∞ 3e4	0/15
21 Gallagher 101 peaks							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	41	1157	1674	1705	1729	1757	14/15
BIPOP-C	2.3 (2.1)	14(8.8)	24(35)	25(36)	25(36)	25(36)	15/15
GLOBAL	2.3 (2.2)	1.1 (0.87)	1 (0.85)	1 (0.83)	1 (0.82)	1 (0.81)	14/15
DIRECT	1(0.50)	1 (1.4)	1.1 (1.6)	2.1 (2.0)	19(29)	19(29)	4/5
MCS	1.0 (0.89)	3.9(6.6)	5.1(6.1)	5.0(6.0)	5.0 (5.9)	5.0 (5.8)	14/15
NEWUOA	1.1 (0.62)	2.2 (2.5)	1.8 (2.1)	1.8 (2.1)	1.8 (2.1)	1.9 (2.0)	15/15
22 Gallagher 21 peaks							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	71	386	938	1008	1040	1068	14/15
BIPOP-C	6.9(11)	20(14)	45(94)	42(88)	41(85)	40(83)	15/15
GLOBAL	3.6(1.7)	1.3 (0.90)	1 (1.2)	1 (1.1) \downarrow^2	1 (1.1) \downarrow^2	1 (1.0) \downarrow^2	14/15
DIRECT	1(0.63)	1 (0.55)	12 (11)	22(13)	128(125)	400(469)	1/5
MCS	1.0 (0.63)	1.1 (1.3)	12(14)	11 (13)	11 (13)	15 (16)	14/15
NEWUOA	2.1 (3.0)	2.1 (2.2)	2.0 (3.1) \downarrow	2.1 (2.8) \downarrow	2.3 (2.8)	2.4 (2.7)	15/15
23 Katsuuras							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	3.0	518	14249	31654	33030	34256	15/15
BIPOP-C	1.7 (1.8)	13(15)	3.7 (4.2)	1.8 (1.9)	1.8 (1.9)	1.8 (1.8)	15/15
GLOBAL	1.6 (2.0)	1.0 (0.48) \downarrow^4	4.8 (5.6)	∞	∞	∞ 4859	0/15
DIRECT	1.5 (1.5)	3.5(4.2)	5.7 (7.0)	6.0 (6.3)	∞	∞ 1e5	0/5
MCS	3.4(3.5)	2.4 (2.8)	51(57)	∞	∞	∞ 5e4	0/15
NEWUOA	6.2(4.2)	2.4 (2.5)	7.1(8.1)	∞	∞	∞ 3e4	0/15
24 Lunacek bi-Rastrigin							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	1622	2.16e5	6.36e6	9.62e6	1.28e7	1.28e7	3/15
BIPOP-C	2.1 (1.7)	1.6 (2.7)	1 (0.92)	1 (1.1)	1 (1.1)	1 (1.1)	3/15
GLOBAL	4.2 (4.9)	∞	∞	∞	∞	∞ 6417	0/15
DIRECT	7.5(14)	1.9 (2.3)	∞	∞	∞	∞ 1e5	0/5
MCS	7.0(8.0)	3.5(3.7)	∞	∞	∞	∞ 5e4	0/15
NEWUOA	2.9 (2.2)	2.1 (2.2)	∞	∞	∞	∞ 3e4	0/15

of functions for which at least 1 instance was solved to the ultimate precision 10^{-8} by the respective algorithm. Despite the fact that the table shows only one very rough particular view of the results, it can be seen that the BIPOP-CMA-ES scales the best with the dimension. For NEWUOA, we can observe a big drop in the number of solved functions after $D = 3$, but then its performance stays rather constant. Another point to note is almost the same profile of MCS and GLOBAL. This is a pure coincidence—there is a substantial proportion of functions where both methods behave quite differently. The DIRECT is relatively successful for low-dimensional functions ($D \leq 3$), but its success rate then drops significantly.

The following subsections present a detailed discussion of the results broken down by the three main algorithms under the study. The NEWUOA and BIPOP-CMA-ES algorithms are discussed only in the summary discussion of strengths and weaknesses of the individual algorithms in Sec. 7.4.

Table 10: ERT on $f_{20}-f_{24}$ in 20-D over ERT_{best} obtained in BBOB-2009

20 Schwefel $x*\sin(x)$							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	82	46150	3.10e6	5.54e6	5.59e6	5.64e6	14/15
BIPOP-C	4.3 (1.0)	9.2 (4.1)	1 (0.54)	1 (0.34)	1 (0.34)	1 (0.34)	14/15
GLOBAL	5.2(0.38)	1.6 (1.7) \downarrow^3	∞	∞	∞	∞ 8e4	0/15
DIRECT	31(6.1e-3)	∞	∞	∞	∞	∞ 1e5	0/5
MCS	4.7 (0.51)	12 (13)	∞	∞	∞	∞ 8e4	0/15
NEWUOA	1 (0.45) \downarrow^4	15(18)	∞	∞	∞	∞ 4e5	0/15
21 Gallagher 101 peaks							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	561	6541	14103	14643	15567	17589	15/15
BIPOP-C	3.2 (5.5)	55(48)	48(92)	46(89)	43(85)	39(75)	13/15
GLOBAL	1 (0.26)	1 (1.3)	1 (1.2)	1 (1.1)	1 (1.1)	2.1 (2.5)	0/15
DIRECT	3.3(2.2)	27 (33)	∞	∞	∞	∞ 1e5	0/5
MCS	26(36)	32 (32)	26 (28)	25 (28)	23 (25)	32 (36)	2/15
NEWUOA	1.7 (2.5)	2.2 (2.1)	1.2 (1.8)	1.2 (1.8)	1.1 (1.7)	1 (1.5)	15/15
22 Gallagher 21 peaks							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	467	5580	23491	24948	26847	1.35e5	12/15
BIPOP-C	6.8 (13)	13 (21)	215(264)	202(247)	188 (231)	37 (45)	5/15
GLOBAL	1.1 (0.54)	1 (1.5)	1 (1.2)	1 (1.1)	1 (0.99)	1.3 (1.5)	0/15
DIRECT	10(10)	16(20)	∞	∞	∞	∞ 1e5	0/5
MCS	17(43)	20(25)	50 (53)	47 (51)	∞	∞ 8e4	0/15
NEWUOA	1 (1.2) \downarrow	4.9 (6.4)	6.8 (7.9)	6.4 (7.5)	6.0 (6.9)	1.2 (1.4)	7/15
23 Katsuuras							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	3.2	1614	67457	4.89e5	8.11e5	8.38e5	15/15
BIPOP-C	4.3(4.5)	32 (33)	1 (0.83)	2.0 (1.5)	1.2 (0.88)	1.2 (0.85)	15/15
GLOBAL	2.8 (2.7)	1 (0.93) \downarrow^4	∞	∞	∞	∞ 9259	0/15
DIRECT	4.1 (6.7)	52(73)	∞	∞	∞	∞ 1e5	0/5
MCS	1.3 (1.4)	124(131)	∞	∞	∞	∞ 8e4	0/15
NEWUOA	12(8.3)	3.5 (3.3) \downarrow^3	32 (36)	∞	∞	∞ 2e5	0/15
24 Lunacek bi-Rastrigin							
Δf_{target}	1e1	1e0	1e-1	1e-3	1e-5	1e-7	#succ
ERT_{best}	1.34e6	7.48e6	5.19e7	5.20e7	5.20e7	5.20e7	3/15
BIPOP-C	1 (0.90)	1 (0.95)	1 (1.0)	1 (0.98)	1 (1.0)	1 (1.0)	3/15
GLOBAL	∞	∞	∞	∞	∞	∞ 3e4	0/15
DIRECT	∞	∞	∞	∞	∞	∞ 1e5	0/5
MCS	∞	∞	∞	∞	∞	∞ 8e4	0/15
NEWUOA	∞	∞	∞	∞	∞	∞ 2e5	0/15

7.1 Discussion on DIRECT

DIRECT’s approach to the search is a systematic sampling of points lying in a grid combined with the selection of promising regions to sample. The method, however, uses no dedicated local search algorithm to improve the quality of sampled points. Thus, DIRECT’s emphasis on the global search is much larger than that of the other methods.

Due to the absence of the local search algorithm, DIRECT is generally the slowest of the compared methods and within its evaluations budget it often is not able to find the tight target levels. As expected, in higher dimensions both the effectiveness and the efficiency of DIRECT is worse than for small D , since the resulting grid is less dense.

DIRECT works on problems where the main trends provide high enough signal to DIRECT’s decision process. These are primarily easy functions like f_1 , sphere, and f_5 , linear slope. DIRECT can solve these functions even to tight target levels, but its efficiency is not good (it is several orders of magnitude slower than the best BBOB-2009 algorithm).

Table 11: The number of functions (out of 24) for which the algorithm found the ultimate precision of 10^{-8} for at least 1 run (out of 15) on the function.

D	2	3	5	10	20	40
BIPOP-CMA-ES	24	24	23	23	22	20
GLOBAL	18	16	11	8	5	—
DIRECT	24	19	9	3	2	1
MCS	19	16	11	8	5	—
NEWUOA	21	18	11	11	11	9

DIRECT can compete with other algorithms (with the exception of BIPOP-CMA-ES) on multimodal functions which (1) provide a strong enough signal (have a global structure), and (2) do not offer an advantage to algorithms which use local search to improve the sampled points. In the multimodal functions subgroup (f_{15} – f_{19}), DIRECT placed second after BIPOP-CMA-ES, leaving MCS and GLOBAL at most 30 % of solved problems behind (for 3-D and 5-D). In 20-D, its gap to the third algorithm is much lower, but it is still second after BIPOP-CMA-ES and about 5 % better than the next competitor. For the subgroup of multimodal functions with weak structure, DIRECT solves a lower number of problems. These functions do not provide so strong signal to DIRECT’s decision process. Moreover, the Gallagher’s functions f_{21} and f_{22} , which were not solved by DIRECT for $D \geq 10$, are known to be efficiently solvable by a restarted local optimizer, so that the other methods (GLOBAL, MCS, NEWUOA) gained their lead over DIRECT mainly on these functions.

7.2 Discussion on MCS

MCS1 reached the lowest target function value $f_t^* := f_{\text{opt}} + 10^{-8}$ for all instances of the linear slope function f_5 after $2D + 1$ function calls, i.e., after the initialization procedure, since this function has its minimizer in a corner and the initialization list MCS1 contains the boundary values of all variables. The other problems where all instances, dimensions and precision levels were solved are f_1 (sphere function, quadratic and separable), f_8 (Rosenbrock function, original), and f_9 (Rosenbrock function, rotated). The good performance of MCS on f_8 and f_9 is mainly due to the local search algorithm. For all instances and dimensions of f_9 the value f_t^* was already achieved by the first call to the local search algorithm and the same is true for 84 % of the instances and dimensions of f_8 ; in the remaining cases f_t^* was reached by the second call to the local search.

MCS performs well on the separable functions (f_1 – f_5) and the functions with low or moderate condition (f_6 – f_9) at least for small dimensions, and the same holds for the ill-conditioned function f_{12} and the multimodal functions f_{15} and f_{19} – f_{22} . The value f_t^* was reached for at least 12 (out of 15) instances in 2-D for f_4 and f_6 , up to 3-D for f_3 , f_7 , f_{15} , f_{19} , and f_{20} , and up to 5-D for f_2 , f_{21} , and f_{22} ; in addition, it was reached for 8 instances of f_{22} for 10-D. For all other dimensions f_t^* was only reached in a minority of the instances for these functions. The only function apart from the easy functions where reaching f_t^* occurred for at least 2 instances of each dimension is Gallagher’s Gaussian 101-me peaks function f_{21} (101 local optima, moderate conditioning around the global optimum). In 63 % of the “successful” cases for f_2 – f_4 , f_6 , f_7 , f_{12} , f_{15} , and f_{19} – f_{22} , the value f_t^* was already achieved by MCS1, in 15 % by MCS2, and in 11 % each by both MCS3 and later calls to MCS.

The functions for which MCS did not reach f_t^* even in low dimensions in a majority

of the instances belong to two classes. The functions f_{10} , f_{11} , f_{13} , and f_{14} are unimodal with high conditioning, which means that approaching the minimizer becomes more difficult close to the minimizer, and the problem becomes worse in higher dimensions. In low dimensions we may still achieve the precision level 10^{-5} or even 10^{-7} , but in large dimensions Δf_{best} becomes much larger. This is a problem of the local optimization part of MCS; in most cases the local optimization algorithm becomes so slow that the limit $n_{f_{\text{max}}}$ on the number of function calls is reached before a stopping criterion for the local search algorithm is satisfied, and in some cases the local optimization stops due to not improving the function value any more before reaching a local minimizer.

The second class of those functions are the multimodal functions f_{16} (highly rugged), f_{17} (highly multimodal), f_{18} (moderately ill-conditioned counterpart of f_{17}), f_{23} (highly rugged, more than 10^D global optima), and f_{24} (highly multimodal, funnel of one non-global minimizer has roughly 70% of the search space volume); f_{24} in 2-D is the only exception since f_t^* was reached for 8 instances. In many of the runs where the final precision was not achieved (particularly the ones with higher dimensions, but also some of the unsuccessful runs for f_{24} for 2-D), the algorithm was repeatedly caught in the same non-global minimum, and the final f_{best} was larger in higher dimensions. The deterministic nature of MCS does not allow to recover from getting stuck when the algorithm happened to rate the region containing the global minimum as unpromising.

In order to study the effect of the local searches on the performance of the algorithm, we also applied MCS1 with $n_{s_{\text{loc}}} = 0$ (no local search), $s_{\text{max}} = 25D$ (a higher value than before to compensate for the lack of a local optimizer), and $n_{f_{\text{max}}} = 100000$ (the budget used for DIRECT) to the test functions, i.e., no restarts were made and the whole (larger) function evaluation budget was used for a single call to MCS. Since no random element is contained, each problem was only treated once. Most of the results for the “easy” functions f_1 – f_4 , f_8 , f_9 , f_{12} , and f_{19} – f_{22} are far inferior, which suggests that the local searches are to a great extent responsible for the good results. For the functions f_{16} ($D \leq 10$), f_{17} ($D = 5, 10$), and f_{18} ($D = 2, 3, 5$), however, tighter precision levels were reached, and the improvement is largest for f_{16} for $D = 2, 3, 5$. The results support the conjecture made in Sec. 6.5 that launching a local search procedure is only a waste of resources for the 5-D problems f_{15} – f_{19} for three of these problems.

In Huyer and Neumaier (1999) it was already observed that the performance of MCS was not satisfactory for most of the problems with dimensions $D \geq 10$ and for hard multimodal test problems, but except for the two-dimensional Rosenbrock function, the test problems considered there were different from the BBOB test set. Moreover, only a single target function value was considered in that paper. Failures of MCS due to shortcomings of the local search algorithm were not encountered by the authors before.

7.3 Discussion on GLOBAL

GLOBAL obtains the highest number of successful trials for functions from the separable, moderate, ill-conditioned and weak-structure subgroups, specifically for f_1 , f_2 , f_5 , f_6 , f_8 , f_9 , f_{10} , f_{11} , f_{12} , f_{21} and f_{22} in dimensions 2, 3, and 5. For f_1 , f_2 , f_5 , f_8 and f_9 , GLOBAL obtained successful trials for all dimensions.

Considering the different function subgroups, the best behavior of GLOBAL can be observed on the separable (except f_3 and f_4), moderately conditioned (except f_7) and on ill-conditioned functions subgroups. These functions are unimodal (or have a quite large attraction region of the global optimum, f_8 and f_9), which is the main reason for the good performance of GLOBAL. The results on the attractive sector function f_6 in

20-D can be improved by increasing the function evaluation limit of the BFGS method. Finding the best solution for f_8 and f_9 is mainly due to the BFGS local search and partly to the property of these functions presented previously. For the discus function f_{11} , GLOBAL cannot find a target precision value better than 10^{-1} in 20-D due to a problem of the BFGS local search. In this case the method stops too early because it cannot decrease the objective function along the current search direction. GLOBAL also performs well on the Gallagher's multimodal functions f_{21} and f_{22} with weak global structure, thanks to restarting the local search method.

The hardest problems, for which the method did not reach the solution in higher dimensions, are the multimodal Rastrigin functions f_3, f_4, f_{15} and f_{24} . In case of the f_{24} function, even in 2-D we cannot find a better target precision value than 10^{-2} , while in the case of the f_3, f_4, f_{15} functions the Δf_{best} value is not better than 10 in 5-D and 10^2 in 20-D, respectively. The common property of these functions is that they have more than 10^D local optima. Therefore the algorithm cannot discover the overall function structure. Moreover, the size of the region of attraction of the global optimum is small for these problems, and hence the algorithm fails to satisfactorily sample in these regions. GLOBAL also fails to reach precision levels better than 1 on the f_{17} and f_{19} multimodal functions with adequate global structure in 5-D and 10 in 20-D. The reason is the same as presented above.

7.4 Summary of the Strengths and Weaknesses

The DIRECT algorithm is competitive only for the low-dimensional functions and the problems with loose target levels. From the ECDF graphs of ERT, it is obvious that the results for DIRECT are virtually never part of the Pareto front, i.e. DIRECT is almost always dominated by some other algorithm(s).

The most obvious strength of the MCS algorithm is its initial phase. In all tested dimensions ($D \leq 20$), for budgets lower than $100D$, it belongs to the 2 best performing algorithms across all function groups. MCS also exhibits competitive performance on separable functions. The weakness of this algorithm is the employed local search procedure which does not perform well in certain situations, especially for high-conditioned and rugged functions. Moreover, getting stuck in a nonglobal minimizer is a problem occurring for difficult multimodal functions, and, like DIRECT, the splitting procedure handles high dimensions less satisfactorily.

The strength of the GLOBAL algorithm is its ability to reduce the number of calls to the local search procedure. Otherwise, it builds on the strengths of its local search procedures, the Nelder-Mead simplex search and the quasi-Newton method with the BFGS update. Across all function groups and all tested dimensions ($D \leq 20$), it exhibits the fastest progress between $100D$ and $1000D$ function evaluations. It works especially well for the class of unimodal ill-conditioned functions. The weakness of GLOBAL is the fact that the first local search is executed only after 300 function evaluations. This deteriorates the performance of GLOBAL in the initial phase. Another weakness of GLOBAL is the sampling phase especially in the case of functions with small basins of attraction, where the algorithm fails to satisfactorily sample in these regions.

The NEWUOA algorithm competes with MCS for the fastest progress in the initial stages of the search ($\#FEs < 100D$). With increasing dimension, the effectiveness of NEWUOA drops more slowly than for DIRECT, MCS, and GLOBAL, so that it eventually solves the second highest proportion of functions after BIPOP-CMA-ES. The weakness of NEWUOA is its slower rate (compared e.g. to BFGS) with which it updates its quadratic model in the middle stage of the search. This can be the reason for the failure

of the method on multimodal functions with an adequate structure.

The most striking advantage of the BIPOP-CMA-ES algorithm is its ability to solve a very large proportion of problems. Its weakness would then be a slower progress in the initial stages of the search. But even in these cases, to solve the same proportion of problems, BIPOP-CMA-ES is usually at most 10 times slower than the fastest of the other algorithms. After about $1000D$ evaluations, BIPOP-CMA-ES becomes the most successful algorithm among those compared.

8 Summary and Conclusions

In this article, we experimentally compared 5 optimization algorithms and discussed their results, strengths and weaknesses. Their evaluation is related to the particular instances of the individual algorithms in terms of parameter settings and choice of individual components. It is possible that with a different parametrization of the algorithms we could get better results. However, the algorithms were tuned to provide acceptable performance across the test suite, and we believe that the parameter settings are good enough to show the most important aspects of the algorithms.

8.1 Which Algorithm Should I Choose?

The results confirmed that there is no best algorithm. Despite of that, we can give the following suggestions for the black-box scenario. (If the experimenter knows some characteristics of her problem, we refer her to the respective part of Sec. 6). The suggestions hold across the tested dimensions (with a few exceptions for 2-D and 3-D).

Low evaluations budget, $\#FEs < 100D$. The mathematical programming algorithms were designed to provide quick progress in the initial stages of the search, and they fulfill this goal. In the initial stages, MCS and NEWUOA usually provide the best results. There is also a substantial proportion of functions which can be solved quickly to a high precision by these solvers.

Intermediate evaluations budget, $100D < \#FEs < 1000D$. In the middle stage of the search, restarting a good local optimizer can be very rewarding. At these stages, GLOBAL and restarted NEWUOA often belong to the best algorithms.

There are also problems which are very hard for these algorithms and which were solved only by BIPOP-CMA-ES. The other methods are not able to solve these problems in a reasonable time; they need a high number of function evaluations and are stopped too early to solve them. Some form of aggregation or averaging is often needed to solve some of the multimodal benchmark problems, and BIPOP-CMA-ES is the only algorithm in this comparison that performs such aggregation.

The final proportion of solved problems. BIPOP-CMA-ES is a clear winner from this point of view (it usually overtakes the other tested algorithms for budgets larger than $\#FEs > 1000D$). With NEWUOA and GLOBAL, which placed second and third, it belongs to the class of algorithms which restart an efficient local optimizer (possibly varying some algorithm parameters). On the contrary, MCS and DIRECT—representatives of the “divide and conquer” approach—are worse and slower in the later stages of the search.

Despite the fact that the mathematical programming methods often provide some important theoretical properties (convergence properties for DIRECT and MCS; finite number of local searches for GLOBAL), a good stochastic optimizer with origins in evolutionary computation, BIPOP-CMA-ES, is able to (1) find the optimum of a broader class of functions, (2) solve problems with a higher precision, and (3) solve some problems faster.

8.2 Suggestions for Hybridization

The evolutionary community can find inspiration in many features of the mathematical programming methods. We can name at least a few.

The successful sampling process of MCS or NEWUOA in the beginning of the search can be used for the population initialization. It may bias the algorithm on the one hand, but could provide an initial boost on the other hand.

To prevent premature convergence, many evolutionary algorithms (EAs) occasionally incorporate randomly generated individuals into the population (or reinitialize a part of the population). The systematic sampling procedure of DIRECT can be used as a generator of the new population members. It would ensure that the search space is covered, yet the new members would not be mere random guesses, but samples of promising regions.

Many memetic algorithms can be viewed as multistart methods with an EA serving as the generator of the starting points for the local search method. We can use the clustering method of GLOBAL to filter the starting point candidates generated by an EA to reduce the chance of finding an already known local optimum. A different method with the same goal was introduced in the context of genetic algorithms as Clustered Genetic Search (Schaefer, 2007).

It is also possible to use some particular principles of the MP methods to improve the behavior of EAs. One attempt in this direction was already presented in the bachelor thesis of Körner (2011). The principle of DIRECT to select the potentially optimal boxes was used to pre-select the suitable pairs of parents for arithmetic crossover in an otherwise canonical EA. The unpromising pairs are filtered out and more effort is put to crossing over the pairs of parents with high chance of producing a good offspring. Such a modification significantly improved the convergence speed in the beginning of the search.

Acknowledgements. The authors would like to thank to N. Hansen, R. Ros, and A. Auger for the tremendous work during the development of the COCO framework. Many thanks go to A. Neumaier for his patient discussions on global optimization. The first author was supported by the Ministry of Education, Youth and Sport of the Czech Republic with the grant No. MSM6840770012 entitled “Transdisciplinary Research in Biomedical Engineering II”. The third author was supported by the grant TÁMOP-4.2.2/08/1/2008-0008 of the Hungarian National Development Agency. The authors wish to thank the anonymous reviewers for their useful constructive comments.

References

- Boender, C. G. E., Rinnooy Kan, A. H. G., Timmer, G. T., and Stougie, L. (1982). A stochastic method for global optimization. *Mathematical Programming*, 22:125–140.
- Csendes, T. (1988). Nonlinear parameter estimation by global optimization—efficiency and reliability. *Acta Cybernetica*, 8(4):361–370.
- Csendes, T., Pál, L., Sendin, J. O. H., and Banga, J. R. (2008). The GLOBAL Optimization Method Revisited. *Optimization Letters*, 2:445–454.
- Finck, S., Hansen, N., Ros, R., and Auger, A. (2009a). Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE.
- Finck, S., Hansen, N., Ros, R., and Auger, A. (2009b). Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE. Updated February 2010.

- Finkel, D. E. (2003). *DIRECT Optimization Algorithm User Guide*.
- Hansen, N. (2009). Benchmarking a BI-population CMA-ES on the BBOB-2009 function testbed. In Rothlauf (2009), pages 2389–2396.
- Hansen, N., Auger, A., Finck, S., and Ros, R. (2009a). Real-parameter black-box optimization benchmarking 2009: Experimental setup. Technical Report RR-6828, INRIA.
- Hansen, N., Auger, A., Ros, R., Finck, S., and Pošík, P. (2010). Comparing results of 31 algorithms from the black-box optimization benchmarking BBOB-2009. In *Proceedings of the 12th annual conference companion on Genetic and evolutionary computation, GECCO '10*, pages 1689–1696, New York, NY, USA. ACM.
- Hansen, N., Finck, S., Ros, R., and Auger, A. (2009b). Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA.
- Hoos, H. and Stützle, T. (1998). Evaluating Las Vegas algorithms—pitfalls and remedies. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 238–245.
- Huyer, W. and Neumaier, A. (1999). Global optimization by multilevel coordinate search. *Journal of Global Optimization*, 14(4):331–355–355.
- Huyer, W. and Neumaier, A. (2009). Benchmarking of MCS on the noiseless function testbed. <http://www.mat.univie.ac.at/~neum/papers.html>. P. 989.
- Jones, D. R., Perttunen, C. D., and Stuckman, B. E. (1993). Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181.
- Kelley, C. T. (1999). *Iterative Methods for Optimization*. SIAM in Philadelphia.
- Körner, P. (2011). Evolutionary algorithms with crossover success estimation. Bachelor thesis, Czech Technical University in Prague.
- Nelder, J. and Mead, R. (1965). A simplex method for function minimization. *The Computer Journal*, 7(4):308–313.
- Neumaier, A. (2004). Complete search in continuous global optimization and constraint satisfaction. *Acta Numerica*, 13:271–369.
- Pál, L., Csendes, T., Markót, M. C., and Neumaier, A. (2009). BBO-benchmarking of the GLOBAL method for the noiseless function testbed. <http://www.mat.univie.ac.at/~neum/papers.html>. P. 986.
- Pošík, P. (2009). BBOB-benchmarking the DIRECT global optimization algorithm. In Rothlauf (2009), pages 2315–2320.
- Powell, M. J. D. (2006). The NEWUOA software for unconstrained optimization without derivatives. *Large Scale Nonlinear Optimization*, pages 255–297.
- Price, K. (1997). Differential evolution vs. the functions of the second ICEO. In *Proceedings of the IEEE International Congress on Evolutionary Computation*, pages 153–157.
- Rinnooy Kan, A. H. G. and Timmer, G. T. (1987). Stochastic global optimization methods Part I: Clustering methods. *Mathematical Programming*, 39:27–56.
- Ros, R. (2009a). Benchmarking the BFGS algorithm on the BBOB-2009 function testbed. In Rothlauf (2009), pages 2409–2414.
- Ros, R. (2009b). Benchmarking the NEWUOA on the BBOB-2009 function testbed. In Rothlauf (2009), pages 2421–2428.
- Rothlauf, F., editor (2009). *Genetic and Evolutionary Computation Conference, GECCO 2009, Proceedings, Montreal, Québec, Canada, July 8-12, 2009, Companion Material*. ACM.
- Schaefer, R. (2007). *Foundations of Global Genetic Optimization*, volume 74 of *Studies in Computational Intelligence*. Springer.