Dimension Selection in Axis-Parallel Brent-STEP Method for Black-Box Optimization of Separable Continuous Functions

Petr Pošík
Czech Technical University in Prague
Fac. of Electrical Eng., Dept. of Cybernetics
Technická 2, 16627 Prague 6, Czech Republic
petr.posik@fel.cvut.cz

Petr Baudiš
Czech Technical University in Prague
Fac. of Electrical Eng., Dept. of Cybernetics
Technická 2, 16627 Prague 6, Czech Republic
baudipet@fel.cvut.cz

ABSTRACT

The recently proposed Brent-STEP algorithm was generalized for separable functions by performing axis-parallel searches, interleaving the steps in individual dimensions in a round-robin fashion. This article explores the possibility to choose the dimension for the next step in a more "intelligent way", i.e. to optimize first along dimensions which are believed to bring the highest profit. We present here the results for the epsilon-greedy strategy, and for a method based on the internals of the Brent-STEP algorithm. Although the proposed methods work better than the round robin strategy in some situations, due to the marginal improvement they bring we suggest the round robin strategy to be used, thanks to its simplicity.

CCS Concepts

•Mathematics of computing → Solvers; Nonconvex optimization; •Computing methodologies → Continuous space search;

Keywords

Black-box optimization, Benchmarking, Line search, Separable functions

1. INTRODUCTION

Black-box optimization methods for bounded separable functions of the form

$$f(\mathbf{x}) = a_1 f_1(x_1) + \ldots + a_D f_D(x_D) \quad (1)$$

are not studied very often, partially due to the fact that the real-world problems are only seldom fully separable. Yet, researchers find it useful to make such methods part of their hybrid algorithm [9] or algorithmic portfolio [3], as a safeguard against separable problems.

Another reason why these methods are only seldom studied is a common belief that the solution of separable problems is simple: one can solve them by decomposing them to D univariate problems, optimizing one after another in a sequence. Yet, even in this setting, there are at least two choices left to be made which may hinder the performance of the resulting algorithm:

1. Which univariate solver shall one choose?
2. What stopping conditions shall one choose for the individual univariate solvers?

Regarding the issue no. 1, we may basically choose either a quickly converging local search algorithm, or a slower global search algorithm. Which solver is a better choice depends on the particular univariate function. The dilemma was recently solved to a great extent [2] by creating a hybrid of the Brent local search method [4] and a global search method called STEP [8]. In most cases, this hybrid called Brent-STEP takes the best of both worlds: it converges quickly on unimodal functions, and still finds the optimum of multimodal functions.

The issue no. 2 is also highly important: the stopping condition must be set separately for each dimension, i.e. for each univariate solver, and its optimal setting depends on the particular combination of the function being optimized and the chosen solver. However, the most important issue here is the fact that for the univariate solvers one cannot use a stopping condition based on the acceptable quality of the solution as a whole since until the last dimension is optimized, there will always be at least one dimension with a far-from-optimal value, making the whole solution look unfavourably. The practitioner usually has to use some fixed budget or stagnation stopping criteria for each dimension, which likely results in wasted resources on the one hand, or in the inability to locate the optimum precisely on the other hand.

An obvious, yet not often mentioned solution to this issue is to interleave the solvers in individual dimensions, i.e. to make an iteration (a single step) of the solver in dimension 1, then in dimension 2, etc., possibly updating the context solution if an improvement is found. This general principle was recently proposed for STEP algorithm [9]. The same general principle, albeit with a different implementation details, was used in [2] to improve the performance of the mul-

GECCO '15, July 11 - 15, 2015, Madrid, Spain

ISBN 978-1-4503-3488-4/15/07…$15.00
DOI: http://dx.doi.org/10.1145/2739482.2768469

tivariate Brent-STEP method. The dimensions were chosen uniformly in a round-robin (RR) fashion. However, RR strategy may not be the optimal choice when there are dimensions which are easier to optimize, and/or bring higher profit than others.

The goal of this work is to evaluate the possibility of choosing the dimension for the next iteration in a smart way, and determine whether the results are worth such an effort. In Sec. 2 we shortly present the optimization algorithms used in this article. Several methods used to choose the dimesion are introduced in Sec. 3 and experimentally compared in Sections 5, 6 and 7 (while the experiment settings are described in Sec. 4). Section 8 summarizes and concludes the paper.

2. BACKGROUND

A popular algorithm for univariate black-box bounded optimization is the method of Brent [4]. It is a variant of golden section search and quadratic interpolation. In principle, it maintains a triple of points which bracket a local optimum. They are interpolated by a quadratic function, and its minimum is considered as the next sampling point. If the minimum does not fulfill some basic conditions, a golden section step is done instead. From the description, it is clear that Brent's method is a local search algorithm.

The STEP method [8] iteratively splits the domain into intervals. In each iteration, it chooses a single interval with the lowest *difficulty* and divides it into halves. The difficulty measure shall express how difficult it should be to improve the current best-so-far (BSF) solution by sampling from the respective interval. The measure used in STEP is defined as the lowest curvature the function would have to have on the interval such that it passes through both the boundary points and somewhere on the interval reaches a level $f_{\mathrm{BSF}} - \epsilon$, i.e. improves BSF by a non-trivial amount ϵ. It is thus based on quadratic interpolation similarly to Brent's method, but it is used in a completely different way. The method is global: it eventually performs a complete search of the whole (discretized) domain; the minimal interval length is a parameter of the algorithm.

A recently proposed hybrid of these two algorithms, Brent-STEP [2], searches through the already sampled points (interval boundaries) for such triples of consecutive points which bracket a local optimum. To each such triple, it fits a parabola in Brent's way and estimates its optimum. The triple with the best estimated optimum is then chosen as a candidate for Brent's method; a single Brent's iteration is then applied to that triple only if the estimated optimum improves the BSF solution by ϵ, at least. If the estimated optimum is not good enough, the algorithm falls back to the STEP method, i.e. it chooses the interval with the lowest difficulty and halves it. The algorithm usually behaves like a local search first (executing Brent's method often), then it switches to STEP, until it finds another promising basin of attraction, where it starts using Brent's method again. It shall eventually end up in a similar state as the STEP method.

Multidimensional versions of Brent's method and STEP constructed in the naive way (optimizing one dimension after another, not interleaved) were compared in [10]. As expected, Brent's method was fast on unimodal separable problems, but failed on multimodal ones, while STEP found the optimum of both the uni- and multimodal separable functions reliably, but was an order of magnitude slower. Both methods had in common, that until the last dimension started to be optimized, the algorithms showed virtually no progress at all. The Brent-STEP method and its multidimensional generalization [2] mitigated both these issues in the same time: the method works for unimodal functions quickly, for multimodal functions reliably, and shows a progress much sooner during the optimization.

3. DIMENSION SELECTION METHODS

The multivariate Brent-STEP method for separable functions [2] uses the round-robin (RR) strategy to choose the dimension for the next iteration. We denote this algorithm as **BSrr**. There are, however, other possibilities, choosing the dimension unevenly, hopefully identifying those dimensions which bring higher profit sooner. However, they may not be easy to design, and one can make a rather huge mistake by choosing a bad one.

First, consider the mere STEP method (not Brent-STEP). In the univariate version, the easiest interval is chosen for the next split. It seems natural to extend it for the multivariate version such that the easiest interval across all dimensions is chosen. The results of this strategy (not shown in this paper) are, however, very disappointing. It turns out that the interval difficulties are not comparable across dimensions. The easiest intervals tend to be in the dimensions which do not have much effect on the overall fitness value; the strategy thus chooses unpromising dimensions.

In the Brent-STEP algorithm, the estimated minima (arising from quadratic interpolation of triples of points bracketing the optimum) are comparable across dimensions and can be used to choose the dimension. This method thus makes a Brent iteration in the dimension with the most improving estimate of the minimum; if no such dimension exists, the method falls back to STEP with the round-robin strategy. We denote this method as **BSqi**.

Instead of trying to predict the quality of the sampling outcome (as done in BSqi), we can track the historical data and choose the dimension which brought most improvements in the past. We decided to track the *improvement frequency* (IF) in each dimension, i.e. the frequency of the event that the newly sampled data point improved the BSF solution. We estimate IF using the exponentially weighted moving average (EWMA) estimator with damping factor 0.9. The resulting methods are denoted as **Sif** and **BSif**.

One can also view the set of univariate solvers as a kind of portfolio [1] of cooperating algorithms working towards a common goal, and in principle we can use here the methods for selecting an algorithm from a portfolio. We choose here only a single example of these methods, namely the ϵ-greedy strategy, which chooses a random dimension with probability ϵ, and the most promising dimension with probability $1 - \epsilon$. We combine it with the IF strategy, giving rise to **Sifeg** and **BSifeg**.

4. EXPERIMENT SETTINGS

We compare the following multivariate algorithms: Srr[1],

[1]Note, that the Srr algorithm is (up to some implementation details) equivalent to the HCMA algorithm [9] with its NEWUOA and BIPOP-CMA-ES components switched off. Also note, that the Srr algorithm is the same as NDstep from [2].

Alg.	Dim. 2	3	5	10	20	40
Srr	1.5	1.3	1.4	1.7	1.5	1.9
BSrr	5.2	5.0	4.2	4.2	5.2	9.0
BSqi	11.0	13.0	14.0	20.0	24.0	65.0

Table 1: The time (in miliseconds) per function evaluation for various dimensions and algorithms.

BSrr[2], Sif, BSif, Sifeg, BSifeg, and BSqi. The capital letters, S or BS, denote STEP or Brent-STEP method, respectively. The lowercase letters denote the dimension selection strategy: 'rr' for round-robin, 'if' for the EWMA estimate of the improvement frequency, 'ifeg' for 'if' combined with ϵ-greedy strategy, and 'qi' for dimension selection based on the prediction of the quadratic interpolation in Brent-STEP.

The value of ϵ which represents a non-trivial amount of improvement for STEP and Brent-STEP was set to 10^{-8}. The damping factor in the EWMA estimate of improvement frequency was set to 0.9. For the ϵ-greedy strategy, the probability of choosing random dimension is 0.5. The first $4D$ evaluations are set as a burn-in phase, where the round-robin strategy is always applied. Methods are restarted if an improvement cannot be found for 2000 iterations. The overall budget was set to $10^4 D$ evaluations in each run.

The experiments are performed according to [6] on the benchmark functions given in [5, 7]. The **expected running time (ERT)**, used as the performance measure, depends on a given target function value, $f_t = f_{opt} + \Delta f$, and is computed over all relevant trials as the number of function evaluations executed during each trial while the best function value did not reach f_t, summed over all trials and divided by the number of trials that actually reached f_t [6, 11].

5. CPU TIMING

Table 1 provides an indication of the CPU timing of our implementations of some of the benchmarked algorithms. Srr (STEP with round-robin strategy) is the simplest one and shall be the fastest. BSrr (Brent-STEP with round-robin strategy) shall reveal the costs of the hybridization of STEP with Brent. BSqi (Brent-STEP with the dimension selected according to the prediction of the quadratic interpolation) was the most time-consuming method in our experiments, and shall indicate the costs of using a more involved dimension selection method.

We have run the algorithms on function f_8 without restarts for at least 30 seconds. The algorithms were implemented and run using Python 2.7 on a Windows 7 machine with Intel(R) Core(TM) i7-2670QM CPU @ 2.20GHz with 1 processor and 4 cores. The time per function evaluation in miliseconds is shown in Table 1.

6. RESULTS FOR UNLIMITED BUDGET

The results for the scenario with an unlimited budget are presented in Figures 1 and 2. The results for non-separable functions f6-f24 (with the exception of f20 and f21) are omited from these figures, because (unsurprisingly) the algorithms are not able to reach the target precision 10^{-8} and the figures do not display anything interesting.

For the separable functions 1, 2, and 4, we can see 2 groups of methods: the lower (better) values of ERT correspond to the Brent-STEP variants, while the higher (worse) ERT values belong to STEP variants. This suggests that on these functions, the hybridization with Brent really improves STEP. On functions 3 and 5, the two groups of methods have similar performance. Functions 3 and 4 are separable and multimodal, and the results confirm that the multivariate STEP and Brent-STEP methods are perfect match for such functions. For functions 1, 2, and 5, which are unimodal with an easy structure, even better solvers exist, but especially the Brent-STEP algorithm does not have a large gap from them.

7. RESULTS FOR EXPENSIVE SCENARIO

The results for the expensive scenario (low budget) are presented in Figures 3, 4, and 5.

The results for separable functions are similar to the unlimited budget scenario, although the differences are less pronounced. The only substantial difference is function f2, where the presented methods actually improve the results of the best 2009 algorithm (which was not the case for unlimited budget).

Although the presented multivariate STEP and Brent-STEP methods were not designed for optimization of non-separable functions, Fig. 3 reveals that there are non-separable functions (8, 12, 14, 16, 20, 23, 24) for which (some of) the presented methods provide performance which is not very far from the methods that take the interactions among variables into account. There are at least 2 factors contributing to the explanation of this observation. (1) Many of the multimodal functions look like unimodal, when levels much worse than the optimum are considered, so that any form of local search helps in the initial phases. (2) During the first n evaluations, more complex methods do not have enough information required by them to work efficiently.

Looking at the ECDF graphs in Figs. 4 and 5, one can say that the sweet spot of applying the STEP and Brent-STEP methods is between $5D$ and $20D$ evaluations. The performance for these budgets is close to or better than the virtual best algorithm of BBOB 2009, and these budgets are also sufficient to find the optimum (target level 10^{-8}) of unimodal separable functions (see Fig. 2). However, to reach the global optima on multimodal separable functions, the (Brent-)STEP algorithms need larger budgets, ca. $400D$ evaluations.

It can be also stated about the addition of Brent's component into STEP, that while it helps on separable functions, it is usually waste of resources on non-separable ones, where the STEP variants are not slower than Brent-STEP, but usually reach higher levels of the proportion of solved problems.

Regarding the individual dimension-selection strategies, on seperable functions we did not find any significant differences among them. It may be, however, caused by the BBOB test problems. If they are constructed by using the same function for all dimensions, it is likely that the individual line searches will have similar improvement frequency; moreover, if the functions in all dimensions have similar range of values, it is also likely, that the optimum estimates from quadratic interpolation will have similar function val-

[2]BSrr algorithm is the same as the NDsqistep from [2].

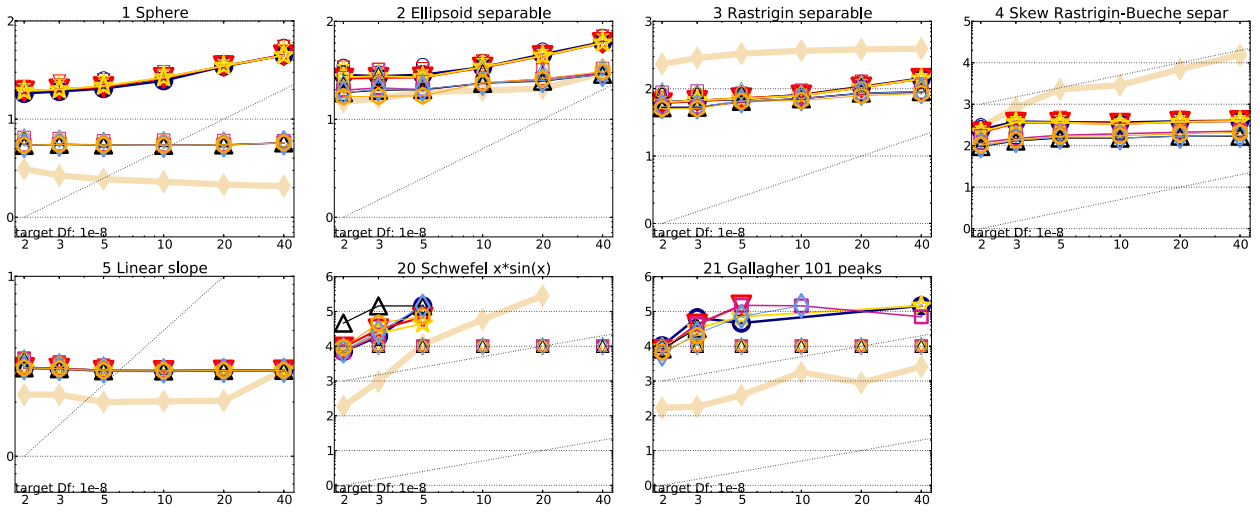

Figure 1: Unconstrained budget scenario: Expected running time (ERT in number of f-evaluations as $\log_{10}$ value), divided by dimension for target function value 10^{-8} versus dimension. Slanted grid lines indicate quadratic scaling with the dimension. Different symbols correspond to different algorithms given in the legend. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Black stars indicate a statistically better result compared to all other algorithms with $p < 0.01$ and Bonferroni correction number of dimensions (six). Different symbols correspond to different algorithms given in the legend. Legend: ○:Srr, ▽:Sif, ⋆:Sifeg, □:BSrr, △:BSif, ◇:BSifeg, ⬡:BSqi

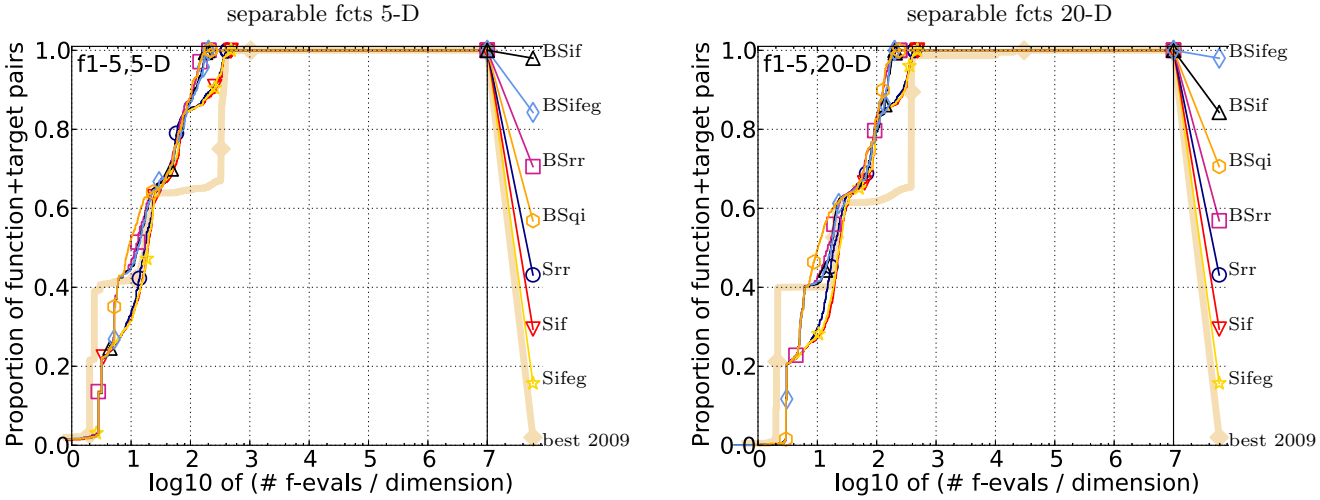

Figure 2: Unconstrained budget scenario: Empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 50 targets in $10^{[-8..2]}$ for separable functions in 5-D and 20-D. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each single target.

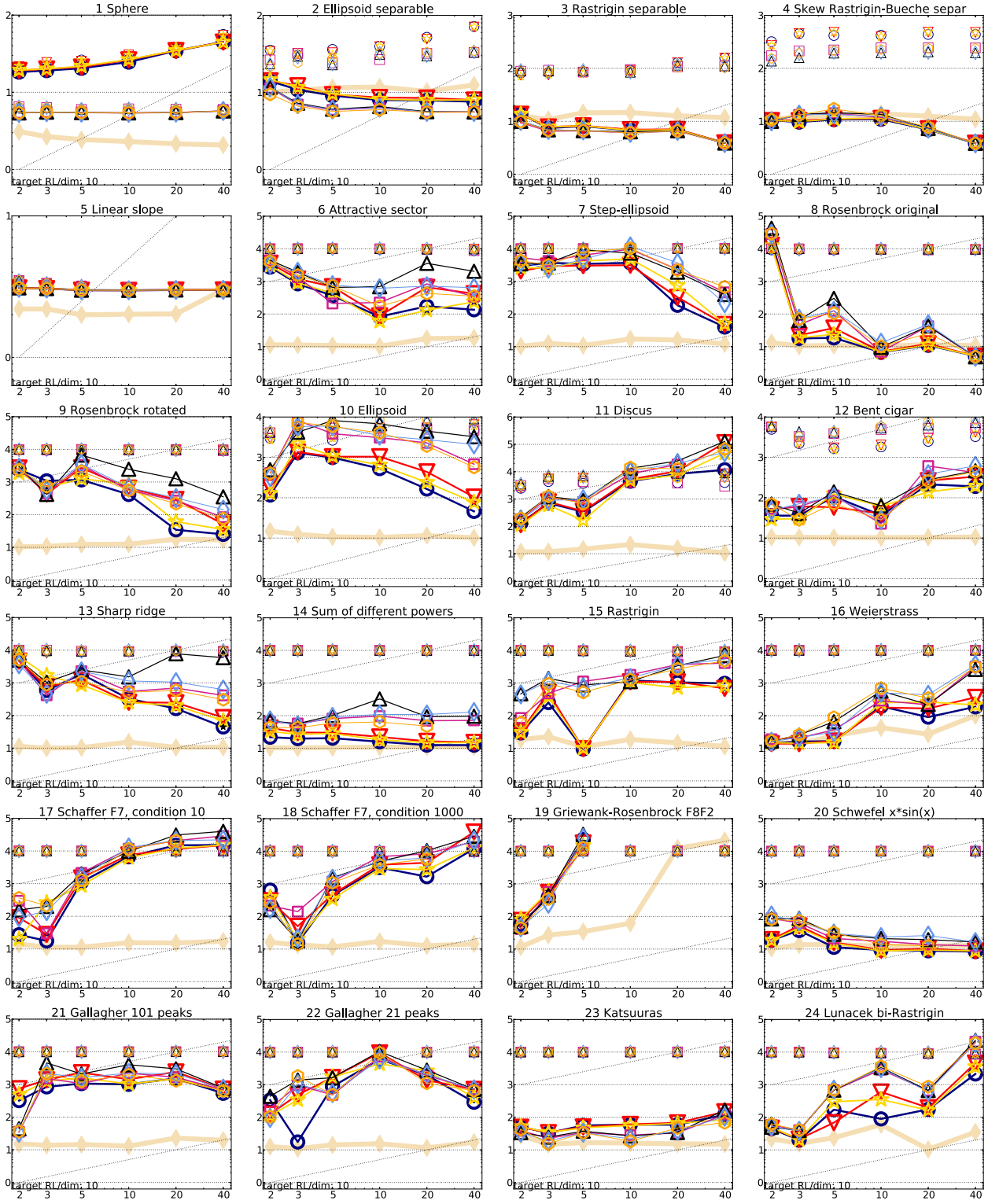

Figure 3: Expensive scenario: Expected running time (ERT **in number of** f**-evaluations as** $\log_{10}$ **value) divided by dimension versus dimension. The target function value is chosen such that the best**GECCO2009 **artificial algorithm just failed to achieve an** ERT **of** $10 \times \mathrm{DIM}$**. Different symbols correspond to different algorithms given in the legend. Light symbols give the maximum number of function evaluations from the longest trial divided by dimension. Black stars indicate a statistically better result compared to all other algorithms with** $p < 0.01$ **and Bonferroni correction number of dimensions (six). Legend:** ○**:Srr,** ▽**:Sif,** ⋆**:Sifeg,** □**:BSrr,** △**:BSif,** ◇**:BSifeg,** ⬡**:BSqi**

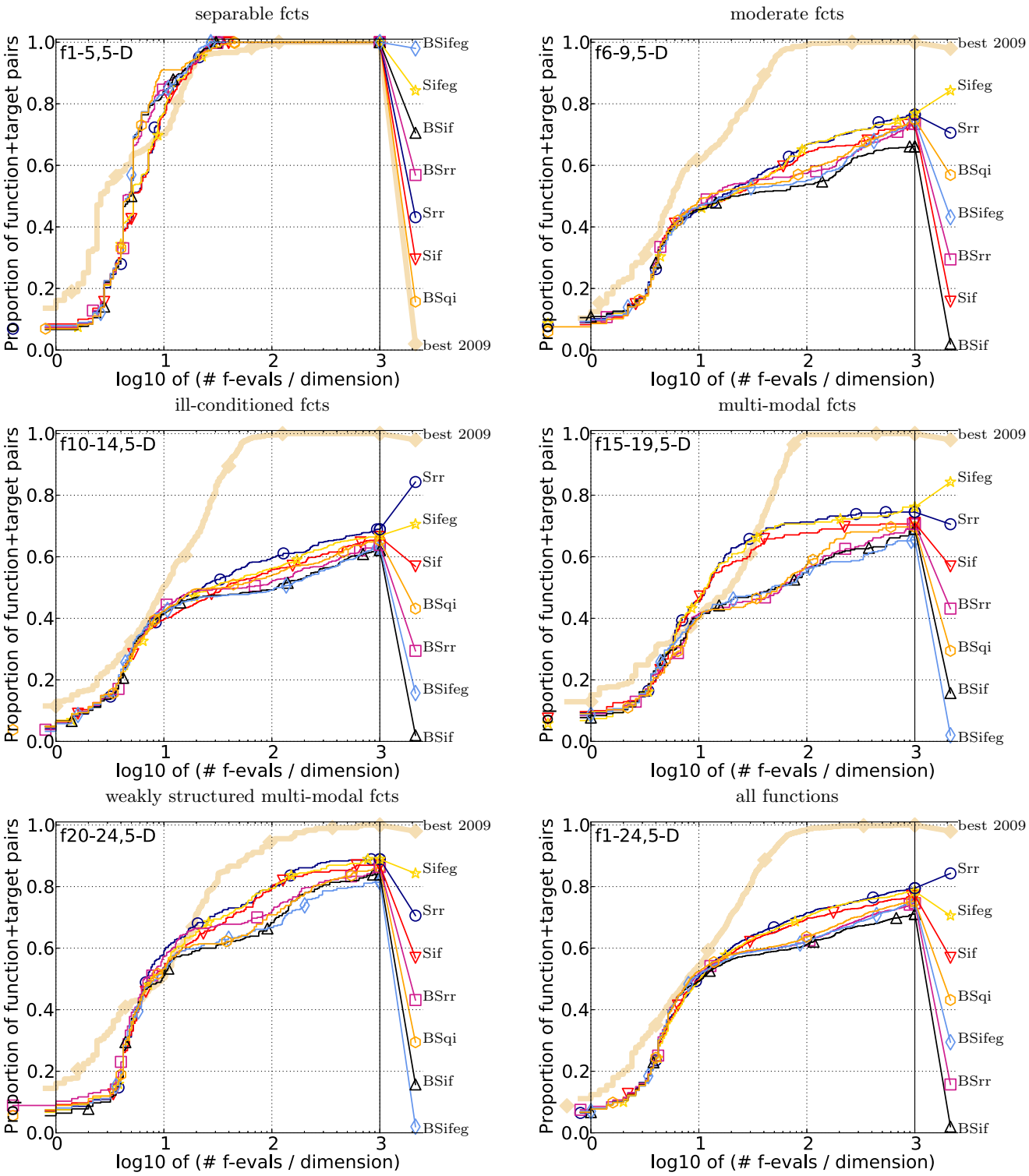

Figure 4: Expensive scenario: Empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for all functions and subgroups in 5-D. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just not reached them within a given budget of $k \times$ DIM, with $k \in \{0.5, 1.2, 3, 10, 50\}$. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each selected target.

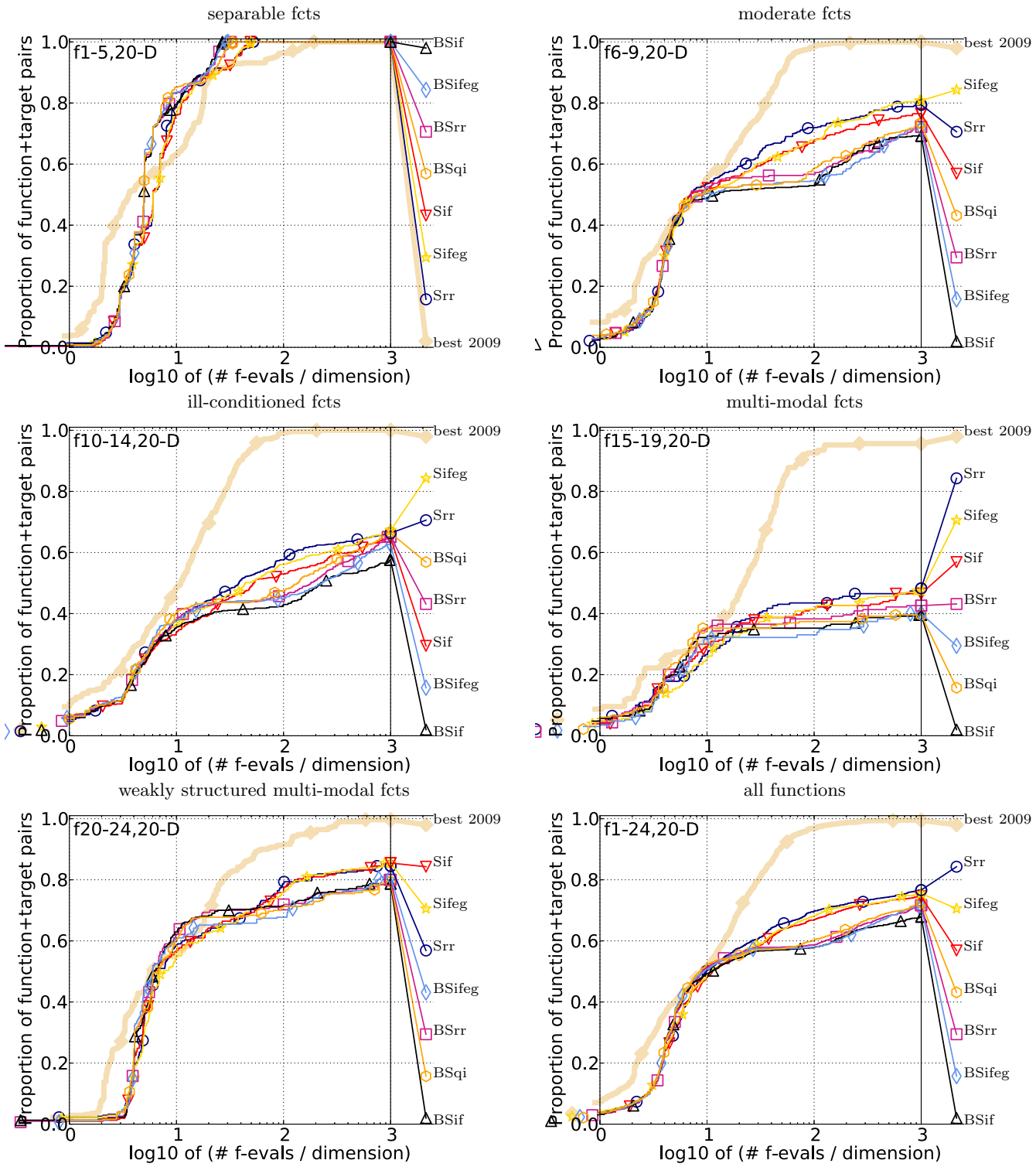

Figure 5: Expensive scenario: Empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for all functions and subgroups in 20-D. The targets are chosen from $10^{[-8..2]}$ such that the bestGECCO2009 artificial algorithm just not reached them within a given budget of $k \times$ DIM, with $k \in \{0.5, 1.2, 3, 10, 50\}$. The "best 2009" line corresponds to the best ERT observed during BBOB 2009 for each selected target.

ues; it is thus possible that all the presented methods behave very often as a round-robin strategy.

The differences among dimension selection strategies are more visible on non-separable functions in higher dimensions. Nevertheless, the non-separable functions are not the primary target area for the proposed algorithms, and thus the differences are largely irrelevant here.

8. SUMMARY AND CONCLUSIONS

The recently proposed Brent-STEP method for optimization of separable functions is reviewed in this article. Several dimension selection strategies are proposed as alternatives to the default round-robin strategy, and experimentally compared.

We confirmed that the hybridization with Brent's method is benefitial for STEP: Brent-STEP is faster on separable functions than STEP. This does not hold for non-separable problems where STEP is usually better. Nevertheless, there are better solvers for non-separable functions, thus we stay with our recommendation to use Brent-STEP method. We further recommend to use the round-robin strategy for dimension selection: we have not found any significant advantage in using a more complicated dimension selection method, but we have found that some of these more complicated methods may worsen the results in certain cases.

In the initial phases of the search, the proposed interleaved axis-parallel methods may be suitable even for the non-separable functions. This can be used e.g. as a part of an initialization procedure of another subsequent solver.

9. ACKNOWLEDGMENTS

This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS14/194/OHK3/3T/13.

Access to computing and storage facilities owned by parties and projects contributing to the National Grid Infrastructure MetaCentrum, provided under the programme „Projects of Large Infrastructure for Research, Development, and Innovations“ (LM2010005), is greatly appreciated.

10. REFERENCES

[1] P. Baudiš and P. Pošík. Online black-box algorithm portfolios for continuous optimization. In T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, editors, *Parallel Problem Solving from Nature – PPSN XIII*, volume 8672 of *Lecture Notes in Computer Science*, pages 40–49. Springer International Publishing, 2014.

[2] P. Baudiš and P. Pošík. Global line search algorithm hybridized with quadratic interpolation and its extension to separable functions. In *Proceedings of the 2015 Conference on Genetic and Evolutionary Computation*, New York, NY, USA, 2015. ACM.

[3] B. Bischl, O. Mersmann, H. Trautmann, and M. Preuss. Algorithm selection based on exploratory landscape analysis and cost-sensitive learning. In *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, pages 313–320, New York, NY, USA, 2012. ACM.

[4] R. P. Brent. *Algorithms for Minimisation Without Derivatives*. Prentice Hall, 1973.

[5] S. Finck, N. Hansen, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Presentation of the noiseless functions. Technical Report 2009/20, Research Center PPE, 2009. Updated February 2010.

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

[7] N. Hansen, S. Finck, R. Ros, and A. Auger. Real-parameter black-box optimization benchmarking 2009: Noiseless functions definitions. Technical Report RR-6829, INRIA, 2009. Updated February 2010.

[8] S. Langerman, G. Seront, and H. Bersini. S.T.E.P.: The Easiest Way to Optimize a Function. In *IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on Evolutionary Computation*, pages 519–524 vol.1, 1994.

[9] I. Loshchilov, M. Schoenauer, and M. Sèbag. Bi-population CMA-ES agorithms with surrogate models and line searches. In *Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation*, GECCO '13 Companion, pages 1177–1184, New York, NY, USA, 2013. ACM.

[10] P. Pošík. BBOB-benchmarking two variants of the line-search algorithm. In *GECCO '09: Proceedings of the 11th annual conference companion on Genetic and evolutionary computation conference*, pages 2329–2336, New York, NY, USA, 2009. ACM.

[11] K. Price. Differential evolution vs. the functions of the second ICEO. In *Proceedings of the IEEE International Congress on Evolutionary Computation*, pages 153–157, 1997.