Hypervolume-Based DIRECT for Multi-Objective Optimisation

Cheryl Wong Sze Yin
School of Computer Science and Engineering
Nanyang Technological University
Singapore 639798
cwong019@e.ntu.edu.sg

Abdullah Al-Dujaili
School of Computer Science and Engineering
Nanyang Technological University
Singapore 639798
aldujail001@e.ntu.edu.sg

S. Suresh
School of Computer Science and Engineering
Nanyang Technological University
Singapore 639798
ssundaram@ntu.edu.sg

ABSTRACT

DIRECT is a deterministic optimization algorithm based on the "divide-and-conquer" paradigm. It looks for optimal solutions by systematic sampling of the feasible decision space in the form of partitioning hyperrectangles. DIRECT has previously been extended to solve black-box multi-objective problems in its pure form in MO-DIRECT and in combination with multi-objective genetic algorithms in NS-DIRECT-GA. In this paper, we examine the two multi-objective variants of the algorithm in terms of diversification (well spread), exploration (global search) and exploitation (local search). Furthermore, motivated by the experimental success of indicator-based methods, we provide a hypervolume-based multi- objective DIRECT, which we refer to as MO-DIRECT-hv. Through the use of the hypervolume indicator, we seek to enhance the diversification of its search through selecting potentially optimal hyperrectangles with higher hypervolume contribution, whilst preserving the exploratory and exploitation balance of the algorithm. To validate the efficacy of the proposed strategy, we compare the performance of the three variants on 55 bi-objective problems from the COCO platform. The results indicate MO-DIRECT-hv performs better than other variants due to diversified sampling.

Categories and Subject Descriptors

G.1.6 [**Numerical Analysis**]: Optimization—*global optimization, unconstrained optimization*; F.2.1 [**Analysis of Algorithms and Problem Complexity**]: Numerical Algorithms and Problems

Keywords

Benchmarking, Black-box optimization, Bi-objective optimization

GECCO'16 Companion, July 20-24, 2016, Denver, CO, USA

DOI: http://dx.doi.org/10.1145/2908961.2931702

1. INTRODUCTION

Multi-objective optimization is a recurrent topic of interest as many engineering problems (e.g., [2,4]) can be modeled as Multi-objective Optimization Problems (MOPs). Due to the often conflicting objectives, MOPs have a set of optimal solutions instead of a single solution because one cannot conclude if solution 1 is better or worse off than solution 2, if solution 1 has a higher value in objective 1 and lower value in objective 2 compared to solution 2. The ultimate best solution is dependent on the preference of the user. Hence, many multi-objective optimization algorithms aim to find the set of optimal solutions, which is also known as the *Pareto* optimal set.

Population-based algorithms such as NSGA-II [7], MOPSO [6] are widely-used to solve MOPs because they are able to provide multiple solutions in a single run. These algorithms generally search for solutions based on the concept of *Pareto*-dominance and a measure of distance between solutions. However, these methods are not deterministic and highly dependent on the initial set of solutions.

DIRECT [13], on the other hand, is an established deterministic algorithm for single objective problems. It divides the feasible solution space in the form of hyperrectangles and searches for the optimal solution through: point-wise evaluation of the objective function at the centers of hyperrectangles and further decomposition of potentially optimal hyperrectangles. DIRECT is primarily exploratory in its search and studies have shown that it may behave as an exhaustive grid search [8].

In the recent years, several researchers attempted to adapt DIRECT in solving MOPs. Till today, to our knowledge, there are two different extensions of DIRECT, namely MO-DIRECT [1] and NS-DIRECT-GA [15]. MO-DIRECT is purely inspired by the basic DIRECT algorithm [13]. It involves decomposing the solution space, in the form of hyperrectangles. The objectives are evaluated at the center of hyperrectangles. The hyperrectangles are then further decomposed into smaller hyperrectangles based on values of the different objectives and the size of the hyperrectangle. NS-DIRECT-GA [15], on the other hand, is a combination of DIRECT and multi-objective genetic algorithm (MOGA). In this case, DIRECT is used to generate the population used in MOGA. After which, the MOGA is applied to find the optimal solution. After an optimal solution is found, it would be located in the solution space of DIRECT. DIRECT would then divide the hyperrectangles from its initial stage before MOGA is applied until the optimal solution is the center of a hyperrectangle, exploring the area around the optimal solution.

With the growing number of emerging algorithms for MOPs, a platform for comparison had to be established. Several quality indicators for performance assessment and comparsion have been proposed [16]; among these metrics is the hypervolume indicator, which—soon after its introduction—has been implemented as a selection operator in multi-objective solvers (e.g., [3]). Many evolutionary algorithms have been modified to use the hypervolume indicator as a selection tool for the next potentially optimal point to explore [9,12] and witnessed an experimental success. Studies have also shown hypervolume indicator based algorithms can achieve a good approximation for an exponentially large *Pareto* front [5].

In this paper, we propose using the hypervolume indicator as a criterion for `DIRECT` to enhance its ability in recognizing a well-spread set of *Pareto* optimal solutions maintaining a balance among the diversification (well spread), exploration (global search), and exploitation (local search) aspects of its search. The performance of `MO-DIRECT-hv` is evaluated on 55 bi-objective problems in COCO platform. The results are compared with `MO-DIRECT` of two strategies, `Rank` and `ND`.

The rest of the paper is organized as follows. Section 2 provides the background on `MO-DIRECT` and explains the theory behind the different strategies in selecting potentially optimal hyperrectangles. Section 3 describes the proposed algorithm `MO-DIRECT-hv`. Section 4 gives the CPU timing for the experiment on the biobjective test suite. Section 5 analyzes the results generated from the experiment. Section 6 provides a conclusion and some suggestions for future work.

2. BACKGROUND ON MO-DIRECT

Section 2.1 first explains the paritioning procedure of hyperrectangles used in `MO-DIRECT`. Then, Section 2.2 provides the formal algorithm of `MO-DIRECT` and two strategies in selecting potentially optimal hyperrectangles. Section 2.3 then reviews these strategies and proposes a new strategy using the hypervolume indicator.

2.1 Partitioning Procedure

`MO-DIRECT` starts with the whole solution space as a single hyperrectangle, sampling the point in the centre. Then, it follows a unique partitioning procedure to split the solution space into smaller hyperrectangles. `MO-DIRECT` first samples 2 points in each dimension of the problem such that the 3 points (including the initial center point) in each dimension is equally spread out. After all the points are sampled, the algorithm divides the solution space equally into one-thirds one dimension at a time, starting with the dimension with the lowest value of

$$w_j = \frac{1}{\min_{k\in\{1,-1\}} \|\mathbf{f}(\mathbf{c}_i + k\cdot\delta\cdot\mathbf{e}_j) - \mathbf{f}(\mathbf{c}_i)\|}\ , \qquad (1)$$

and continue to the dimension with the highest w_j.

In other words, a hyperrectangle is divided such that the biggest produced hyperrectangles contain the distant solutions from that of the hyperrectangle, increasing the likelihood of visiting unexplored regions of the function space.

2.2 MO-DIRECT Framework

The framework of `MO-DIRECT` follows the process of selecting potentially optimal hyperrectangles and dividing them into smaller hyperrectangles using the partitioning procedure in Section 2.1. In literature, two general strategies were followed in seeking potentially optimal hyperrectangles.

2.2.1 The Rank Strategy

The `Rank` strategy employs rank and size of the hyperrectangle to choose potentially optimal hyperrectangles. `NS-DIRECT-GA` [15] previously proposed using rank and crowding distance in choosing potentially optimal hyperrectangles. In this paper, for the sake of simplicity, we only use rank as an indicator and assumed the value of ϵ used in [15] to be 0, which lead to the following

$$\mathcal{I} = \mathtt{ND}(\{(rank_i, \sigma_i) : i \in \mathcal{H}\ , \sigma_i \geq \sigma_t\})\ , \qquad (2)$$

where $(rank_i, \sigma_i)$ is the 2-dimensional vector of hyperrectangle $i \in \mathcal{H}$ on the $rank_i$th front and of size σ_i. $\mathtt{ND}(\cdot)$ is an operator on a set of vectors A such that $\mathtt{ND}(A)$ is the set of non-dominated vectors in A. σ_t is the minimum size, a hyperrectangle can have, to be considered for potential optimality.

Hyperrectangles' ranks are obtained from the process of non-dominated sorting. A non-dominated solution is one that is at least equal in all objectives and better in at least 1 objective when compared to any other solution in the pool of solutions. After all the non-dominated solutions are found, they are removed from the pool of solutions and given a rank of 1. The new pool of solutions would go through the same process and the following non-dominated solutions would be given a rank of 2. This process would go on until all the solutions are ranked. This allows us quantify the quality of solutions using a ranking system.

$$\mathbf{rank} = nondominatedsort(\{\mathbf{f}(\mathbf{c}_i) : i \in \mathcal{H}\ \}) \qquad (3)$$

where $\mathbf{rank}$ is the $|\mathcal{H}|$-vector of the ranks of m-dimensional vectors, $\mathbf{f}(\mathbf{c}_i)$, $\forall i \in \mathcal{H}$ of center $\mathbf{c}_i$, and m is the number of objectives.

2.2.2 The ND Strategy

This `ND` strategy uses the combination of the non-dominated front and size of the hyperrectangle to choose potentially optimal hyperrectangles for division.

$$\mathcal{I} = \mathtt{ND}(\{(\mathbf{f}(\mathbf{c}_i), \sigma_i) : i \in \mathcal{H}\ , \sigma_i \geq \sigma_t\})\ , \qquad (4)$$

where $(\mathbf{f}(\mathbf{c}_i), \sigma_i)$ is the $(m+1)$-dimensional vector of hyperrectangle $i \in \mathcal{H}$ of center $\mathbf{c}_i$ and size σ_i.

2.3 Examination of MO-DIRECT Strategies

In this section, both of the strategies for selecting the potentially optimal hyperrectangles, namely `ND` and `Rank` are analyzed. Furthermore, motivated by the success of indicator-based search algorithms, we adapt the indicator-based approach as an alternative strategy to choose potentially optimal hyperrectangles and highlight its advantages over the former ones.

Figure 1 illustrates how the strategies `ND`, `Rank` and the new strategy `HV` would be choosing potentially optimal hyperectangles based on the given function space. From Figure 1, one can see a significant difference between `ND` and `Rank` in the number of potentially optimal hyperrectangles chosen (the white rectangles), with `ND` choosing 11 points and `Rank` choosing only 3 points. In particular, the number of chosen points from the non-dominated front: `Rank` chooses

Algorithm 1: `MO-DIRECT`

Input : vectorial function to be minimized $\mathbf{f}$,
search space $\mathcal{X}$,
evaluation budget v,
hyprrectangle threshold σ_t

Initialization: $\mathcal{H}_1 = \{\mathcal{X}\}$

Output : approximation set of $\min_{\mathbf{x}\in\mathcal{X}} \mathbf{f}(\mathbf{x})$, $\mathcal{Y}_*^v$

```
while evaluation budget v is not exhausted do
    Evaluate all the new hyperrectangles ∈ H_t.
    Choose potentially optimal hyperrectangles using
      Eq. (4) or Eq. (2).
    Partition the hyperrectangles in I_t according to the
      procedure outlined in Section 2.1 using Eq. (1).
    H_{t+1} ← H_t ∖ I_t ∪ {I_t's newly generated
      hyperrectangles}
    t ← t + 1
return ND({f(c_i)}_{i∈H_t})
```

only one hyperrectangle with the biggest size on the non-dominated front while `ND` chooses all the hyperrectangles (7 hyperrectangles) on the non-dominated front. Given a certain evaluation budget, choosing all the hyperrectangles on the non-dominated front demonstrates excessive sampling leading to inefficient use of function evaluations. On the other hand, choosing only one point on the non-dominated front with the biggest size could allow one to miss out on choosing the most potentially optimal hyperrectangle. This is illustrated in Figure 1, where the most potentially optimal hyperrectangle (4) is not chosen for division. Furthermore, the strategy `Rank` is likely to have a slow convergence to the optimal *Pareto* front.

In order to preserve the exploration and diversification nature of using the `ND` strategy and at the same time reduce inefficient sampling, there has to be a reduction of potentially optimal hyperrectangles chosen on the non-dominated front. From the perspective of `Rank`, more potentially optimal hyperrectangles need to be chosen for a faster convergence to the optimal *Pareto* front. In order to find a balance between the two extremes from choosing one and all the hyperrectangles on the non-dominated front, we propose the use of the hypervolume indicator to selectively choose the potentially optimal hyperrectangles. The hypervolume indicator is able to provide the hypervolume contribution of each point in the objective space. If a point is in an unexplored region on the non-dominated front, it has a high hypervolume contribution. However, if a point is not on the non-dominated front, it has zero hypervolume. In other words, the hypervolume indicator is able to identify relatively more potentially optimal points based on the concept of diversification on the non-dominated front. From Figure 1, we can observe the process of choosing potentially optimal hyperrectangles using the hypervolume indicator. Hyperrectangles on the non-dominated front are carefully selected while only one hyperrectangle with the largest size is chosen on the dominated fronts. Hence, this new hypervolume strategy helps to perform diversified sampling on the non-dominated front while sacrificing some exploration on the dominated fronts.

3. HYPERVOLUME-BASED MO-DIRECT

Motivated by the aforementioned discussion, in this section, we introduce the use of hypervolume indicator to help in a more diversified sampling.

3.1 From Non-dominated Front to Hypervolume Indicator

The equation below illustrates how potentially optimal hyperrectangles are identified using the hypervolume indicator.

$$\mathcal{I} = \mathtt{ND}(\{(-hv_i, \sigma_i) : i \in \mathcal{P}\}) , \qquad (5)$$

where $(-hv_i, \sigma_i)$ replaces $(\mathbf{f}(\mathbf{c}_i), \sigma_i)$ in Equation (4) of the `ND` strategy as the 2-dimensional vector of hyperrectangle $i \in \mathcal{P}$ with hypervolume contribution hv_i and size σ_i. $\mathtt{ND}(\cdot)$ is as previously defined in Section 2.2.1. To be consistent with the minimization settings of $\mathtt{ND}(\cdot)$, a negative sign is added in front of the hypervolume indicator to identify the hyperrectangles with the highest hypervolume.

3.2 Generating the Hypervolume contribution

The algorithm first identifies the non-dominated front of the set of hyperrectangles that are considered for potential optimality.

$$\mathcal{P} = \mathtt{ND}(\{\mathbf{f}(\mathbf{c}_i) : i \in \mathcal{H} , \sigma_i \geq \sigma_t\}) , \qquad (6)$$

where $\mathcal{P}$ is the set of hyperrectangles on the non-dominated front. The rest of the notations are as in Section 2.2.1. After the non-dominated front is found, the hypervolume contribution of every point on the non-dominated front is calculated.

$$\mathbf{hv} = hypervolume(\{\mathbf{f}(\mathbf{c}_i) : i \in \mathcal{P} \}) \qquad (7)$$

where $\mathbf{hv}$ is the $|\mathcal{P}|$-vector consisting of the individual hypervolume contribution of each point i on the non-dominated front, $hypervolume(\cdot)$ is an operator on a set of vectors A such that $hypervolume(A)$ returns the hypervolume contribution of each point in A. Only hyperrectangles in the non-dominated front $\mathcal{P}$ is considered because hyperrectangles in the dominated front would return a value of zero.

3.3 Visiting Unexplored Area in the Solution Space

The hypervolume indicator focuses on diversifying the selection of hyperrectangles on the non-dominated front, while selecting only a single point on the dominated fronts. This in turn reduces the exploratory nature of `MO-DIRECT`, leaving some areas in the solution space unexplored. In MOPs, the points on the *Pareto* front in objective space does not necessarily occur in the same area but often as clusters in different areas of the solution space. Therefore, a global search to identify potentially optimal hyperrectangles is required when such a situation occur. In other words, a global search is necessary when the algorithm is stuck in a *local* optimal *Pareto* set or a part of the *global Pareto* front.

To illustrate this idea, let us go back to Figure 1. Figure 1 shows the approximate *Pareto* front being divided into three groups, points 1,2,3, point 4 and points 5,6,7. From this, we may deduce that these three groups of points belong to different areas in the solution space. Let us assume that we have found only the points 1, 2, 3, 5, 6, and 7; using the hypervolume indicator would keep us selecting points 3 and 5 as potentially optimal hyperrectangles. This is because the hypervolume indicator aims to diversify the solutions on the *Pareto* front, hoping to find point 4. However, we might

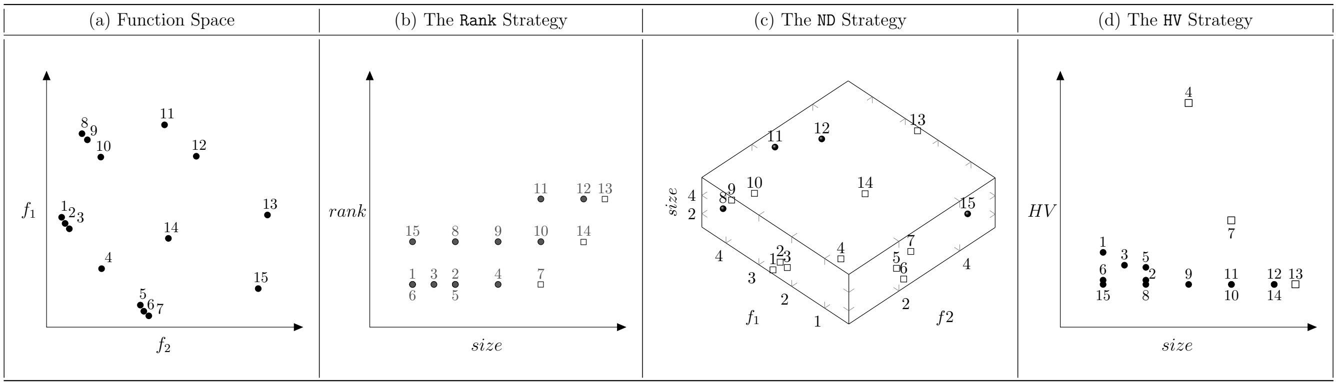

Figure 1: Visualization of sampling strategies in MO-DIRECT for the points whose objective vectors are shown (a). In each of the strategies, the selected points are marked with the symbol □ compared to other points (marked with ●). Whilst conservative (identifying 3 potentially optimal points), the Rank strategy (b) fails to select the most potentially optimal point (4) on the contrary of the ND strategy (c), which on the other hand, indifferently selects all the points on the non-dominated front (11 out of the 15 points). The ND strategy preserves its exploration and diversification balance at the expense of exploitative search, whereas the Rank strategy preserves its exploration and exploitation balance at the expense of diversification search. With this concern, indicator-based search can prove to be effective in achieving a balance among the three search components.

not be able to find solution 4 as it could be in a different area in the solution space. Thus, in such scenarios, we seek to increase the exploratory aspect of the algorithm to look into other areas in the solution space.

To this end, we use intermittently and whenever necessary the strategy of Rank to increase the global search aspect of the proposed technique. Being stuck in a local optimum, we want to avoid choosing hyperrectangles on the non-dominated front as much as possible; and the Rank strategy serves as a suitable means as only one hyperrectangle on the non-dominated front would be chosen.

With the implementation of this global search feature using Rank coupled with the hypervolume indicator, we are able to achieve a balance among the three search components of diversification, exploration and exploitation. The next section summarizes the algorithm, which we refer to as MO-DIRECT-hv.

3.4 The MO-DIRECT-hv Algorithm

Algorithm 2 provides a formal description of MO-DIRECT-hv. It implements the use of hypervolume indicator to select potentially optimal hyperrectangles. This is only activated when the number of hyperrectangles on the non-dominated front is more than two, because it helps to choose potentially optimal hyperrectangles on the non-dominated front wisely. However, due to possible exploitative behaviour—as discussed in Section 3.3; selecting only one point out of all the dominated fronts, the algorithm could be stuck in a local optimum. In order to overcome this issue, the algorithm would conduct a global search using the Rank strategy to explore the other areas when the hypervolume contributions of all the solutions do not change over time given that the hypervolume contribution of any points is not significantly large (Line 8 in Algorithm 2).

Algorithm 2: MO-DIRECT-hv

Input : vectorial function to be minimized $\mathbf{f}$, search space $\mathcal{X}$, evaluation budget v, hyprrectangle threshold σ_t
Initialization: $\mathcal{H}_1 = \{\mathcal{X}\}$
Output : approximation set of $\min_{\mathbf{x}\in\mathcal{X}} \mathbf{f}(\mathbf{x})$, $\mathcal{Y}_*^v$

1 **while** *evaluation budget v is not exhausted* **do**
2 Evaluate all the new hyperrectangles $\in \mathcal{H}_t$.
3 $\mathcal{P} \leftarrow \texttt{ND}(\{\mathbf{f}(\mathbf{c}_i) : i \in \mathcal{H}_t, \sigma_i \geq \sigma_t\})$.
4 **if** *size* $(\mathcal{P}) > 2$ **then**
5 $hv_j = hypervolume(\{\mathbf{f}(\mathbf{c}_j) : j \in \mathcal{P}\})$.
6 $P^t_{hv} = \sum(hv_j)$.
7 $\mathcal{I}_t \leftarrow \texttt{ND}(\{(-hv_j, \sigma_j) : j \in \mathcal{P}\})$.
8 **if** $P^t_{hv} - P^{t-1}_{hv} < 0.0001 \& \max(\mathbf{hv}) > 0.001 * P^t_{hv}$ **then**
9 $rank_i = nondominatedsort(\{\mathbf{f}(\mathbf{c}_i) : i \in \mathcal{H}_t, \sigma_i \geq \sigma_t\})$.
10 $\mathcal{I}_t = \texttt{ND}(\{(rank_i), \sigma_i) : i \in \mathcal{H}, \sigma_i \geq \sigma_t\})$.
11 **else**
12 $\mathcal{I}_t = \mathcal{P}$
13 **return** $\mathcal{I}_t$.
14 Partition the hyperrectangles in $\mathcal{I}_t$ according to the procedure outlined in Section 2.1 using Eq. (1)
15 $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t \setminus \mathcal{I}_t \cup \{\mathcal{I}_t$'s newly generated hyperrectangles$\}$
16 $t \leftarrow t + 1$
17 **return** $\texttt{ND}(\{\mathbf{f}(\mathbf{c}_i)\}_{i\in\mathcal{H}_t})$

4. CPU TIMING

In order to evaluate the CPU timing of the algorithm, we have run MO-DIRECT with strategies, Rank, ND as well as the proposed approach without restarts on the entire bbob-biobj test suite with solution space dimension $D \in \{2, 3, 5, 10, 20\}$. The Matlab code was run on a Windows machine: Intel(R)

Xeon(R) CPU E5-1650@ 3.20GHz with 1 processor and 6 cores. The time per function evaluation for `Rank` in dimensions 2, 3, 5, 10, 20 equals 0.4672041, 0.5645781, 0.6768276, 0.9111276, 1.324321 millisecond, respectively. The time per function evaluation for `ND` in dimensions 2, 3, 5, 10, 20 equals 0.1016983, 0.08452395, 0.07248416, 0.05746423, 0.05250416 milliseconds, respectively. The time per function evaluation for `MO-DIRECT-hv` in dimensions 2, 3, 5, 10, 20 equals 0.8100738, 0.4218101, 0.4089053, 0.4361579, 0.3410024 milliseconds respectively.

5. RESULTS

Results of all three strategies from experiments according to [10,11] on the benchmark functions given in [14] are presented in Figures 2, 3 and Table 1. An evaluation budget of $1000 \cdot D$ is used. Table 1 displays average runtime (aRT) to reach given targets, measured in number of function evaluations on selected functions ($f1 - f28$). The experiments were performed with COCO [10], version 1.0, the plots were produced with version 1.1.

5.1 Overall Performance

From Table 1, one can see `MO-DIRECT-hv` is able to hit more hypervolume target values as compared to `Rank` and `ND` with lesser number of evaluations in most of the functions (indicated in bold). This in turn affirms the motivation behind our proposed idea that using the hypervolume indicator is more effective in finding a good approximation of the *Pareto* front in the `MO-DIRECT` framework.

Moreover, from Figure 2, one can see that `MO-DIRECT-hv` performs better than the other two strategies, `Rank` and `HV` used in `MO-DIRECT`, in all dimensions of the benchmark functions, in terms of hitting the targets and the number of evaluations required to hit the targets. This again reaffirms the motivation behind using hypervolume indicator. The improvement in using `MO-DIRECT-hv` as compared to strategies `Rank` and `HV` becomes more significant in higher dimensions such as $5D$, $10D$ and $20D$. However, at higher dimensions, the number of targets found on the *Pareto* front decreased significantly for all strategies. This is because of the sampling procedure of `MO-DIRECT`, which uses an increasing number of function evaluations in each iteration as the number of dimensions increases. When `MO-DIRECT` picks a potentially optimal hyperrectangle to divide, it would sample up to $2 \cdot D$ points where D is the number of dimensions. Therefore, in higher dimensions, such as $20 - D$, 40 points would be sampled without any consideration if the 40 points chosen are potentially optimal or not. This in turn leads to inefficient sampling when compared to evolutionary algorithms.

5.2 Performance on Different Types of Functions

Figure 3 allows us to take a closer look at different categories of functions and how different strategies perform. On closer analysis, `MO-DIRECT-hv` shows larger improvement in separable, moderate, ill-conditioned functions compared to multi-modal and weakly-structured functions. In the category of weakly-structured functions, at least one of the two objective functions is a highly multi-modal function.

This in turn led us to reason that `HV` might not be able to find a significantly larger number of solutions than `ND` and `Rank` in problems with multi-modal function. Multi-modal functions consist of multiple minima points, which usually indicates that the optimal solutions on different parts of the Pareto front would lie in different areas in the solution space. Therefore, `MO-DIRECT-hv`, being implemented as a relatively more exploitative measure would face difficulty in the search of optimal solutions in such functions. However, it is still able to identify the optimal solutions at a faster rate than `Rank` and `ND`.

6. CONCLUSION

This paper introduced `MO-DIRECT-hv`, an algorithm that employs the hypervolume indicator with dividing rectangles to identify solutions on the *Pareto* front. `MO-DIRECT-hv` is benchmarked against two strategies, namely strategy `Rank` and `ND` in the bi-objective test suite of 55 problems on the COCO platform. `MO-DIRECT-hv` is able to find better approximation sets of the *Pareto* front and shows faster convergence to quality targets compared to the other two strategies. Further studies can be done to see how effective this method is when used in problems with higher number of objectives. However, in higerh dimensions, due to the basic partitioning procedure of `MO-DIRECT`, many points are sampled at once, leading to inefficient sampling. To overcome this problem, one would have to explore other methods of partitioning the space, such as adaptive grid partitioning.

ACKNOWLEDGMENT

The authors wish to extend their thanks to the ATMRI:2014-R8, Singapore, for providing financial support to conduct this study.

7. REFERENCES

[1] A. Al-Dujaili and S. Suresh. Dividing rectangles attack multi-objective optimization. In *IEEE Congress on Evolutionary Computation (CEC), 2016*. IEEE, 2016.

[2] E. Asadollahi-Yazdi, A. Hassan, A. Siadat, J. Y. Dantan, A. Azadeh, and A. Keramati. Multi-objective optimization for inspection planning using nsga-ii. In *Industrial Engineering and Engineering Management (IEEM), 2015 IEEE International Conference on*, pages 1422–1426, Dec. 2015.

[3] N. Beume, B. Naujoks, and M. T. M. Emmerich. SMS-EMOA: multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.

[4] K. Bi, B. Liu, W. Zhang, and R. Nie. Optimization of aircraft flight control system based on an improved mopso algorithm. In *Electrical and Control Engineering (ICECE), 2011 International Conference on*, pages 4842–4845, Sept. 2011.

[5] D. Brockhoff, T. Friedrich, and F. Neumann. Analyzing hypervolume indicator based algorithms. In *Parallel Problem Solving from Nature - PPSN X*. Springer, Jan. 2008.

[6] C. A. C. Coello and M. S. Lechuga. Mopso: a proposal for multiple objective particle swarm optimization. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, volume 2, pages 1051–1056, 2002.

[7] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm:

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f1							
Rank	**1**(0)	2011(1627)	∞	∞	∞	∞ *5007*	0/5
ND	**1**(0)	1625(730)	∞	∞	∞	∞ *5001*	0/5
HV-Rank	**1**(0)	**179**(76)	**2092**(1577)	∞	∞	∞ *5001*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f2							
Rank	68(42)	2304(4216)	∞	∞	∞	∞ *5003*	0/5
ND	56(78)	3927(3398)	∞	∞	∞	∞ *5005*	0/5
HV-Rank	**23**(27)	**299**(92)	**5199**(2856)	∞	∞	∞ *5005*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f3							
Rank	**1**(0)	3263(4166)	∞	∞	∞	∞ *5005*	0/5
ND	**1**(0)	4270(2140)	∞	∞	∞	∞ *5003*	0/5
HV-Rank	**1**(0)	**665**(345)	**3165**(2829)	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f4							
Rank	**1**(0)	1411(591)	∞	∞	∞	∞ *5001*	0/5
ND	**1**(0)	1725(732)	2.4e4(5e4)	∞	∞	∞ *5003*	0/5
HV-Rank	**1**(0)	**512**(705)	**1565**(1356)	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f5							
Rank	**1**(0)	2180(118)	∞	∞	∞	∞ *5005*	0/5
ND	**1**(0)	2398(1187)	∞	∞	∞	∞ *5003*	0/5
HV-Rank	**1**(0)	**465**(411)	**4820**(3683)	∞	∞	∞ *5001*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f6							
Rank	**1**(0)	1142(836)	∞	∞	∞	∞ *5003*	0/5
ND	**1**(0)	1769(544)	∞	∞	∞	∞ *5001*	0/5
HV-Rank	**1**(0)	**192**(52)	**2277**(1298)	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f7							
Rank	**1**(0)	3552(2754)	∞	∞	∞	∞ *5003*	0/5
ND	**1**(0)	3613(2274)	∞	∞	∞	∞ *5001*	0/5
HV-Rank	**1**(0)	**1765**(1863)	∞	∞	∞	∞ *5007*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f8							
Rank	64(79)	4760(2016)	∞	∞	∞	∞ *5003*	0/5
ND	114(142)	6258(4280)	∞	∞	∞	∞ *5005*	0/5
HV-Rank	**16**(22)	**3222**(2274)	∞	∞	∞	∞ *5001*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f9							
Rank	**1**(0)	1112(700)	∞	∞	∞	∞ *5005*	0/5
ND	**1**(0)	1383(661)	2.3e4(3e4)	∞	∞	∞ *5005*	0/5
HV-Rank	**1**(0)	**310**(282)	**1725**(1266)	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f10							
Rank	**1**(0)	1788(480)	∞	∞	∞	∞ *5007*	0/5
ND	**1**(0)	1967(951)	∞	∞	∞	∞ *5003*	0/5
HV-Rank	**1**(0)	**434**(552)	**2.3e4**(3e4)	∞	∞	∞ *5005*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f11							
Rank	**1**(0)	2.2e4(3e4)	∞	∞	∞	∞ *5005*	0/5
ND	**1**(0)	2418(4161)	∞	∞	∞	∞ *5003*	0/5
HV-Rank	**1**(0)	**818**(161)	∞	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f12							
Rank	**1**(0)	1866(1826)	4533(9226)	1.1e4(1e4)	∞	∞ *5001*	0/5
ND	**1**(0)	2098(6058)	4985(5346)	2.4e4(3e4)	∞	∞ *5005*	0/5
HV-Rank	**1**(0)	**1611**(2835)	**3484**(3772)	**1.0e4**(5234)	∞	∞ *5007*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f13							
Rank	14(32)	550(718)	5335(5004)	∞	∞	∞ *5005*	0/5
ND	33(80)	1528(2179)	1.0e4(2809)	∞	∞	∞ *5007*	0/5
HV-Rank	**8.8**(20)	**77**(47)	**229**(89)	**5623**(5836)	∞	∞ *5005*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f14							
Rank	**1**(0)	1.1e4(2e4)	∞	∞	∞	∞ *5003*	0/5
ND	**1**(0)	2.3e4(2e4)	∞	∞	∞	∞ *5003*	0/5
HV-Rank	**1**(0)	**1267**(852)	∞	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f15							
Rank	108(134)	2984(3494)	∞	∞	∞	∞ *5001*	0/5
ND	600(1429)	9064(8877)	∞	∞	∞	∞ *5001*	0/5
HV-Rank	**34**(0)	**765**(1028)	**2.3e4**(2e4)	∞	∞	∞ *5001*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f16							
Rank	**1**(0)	3000(3323)	∞	∞	∞	∞ *5003*	0/5
ND	**1**(0)	9285(2e4)	∞	∞	∞	∞ *5005*	0/5
HV-Rank	**1**(0)	**1470**(2583)	∞	∞	∞	∞ *5005*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f17							
Rank	1286(36)	4748(9869)	∞	∞	∞	∞ *5005*	0/5
ND	1331(2545)	9510(1e4)	∞	∞	∞	∞ *5005*	0/5
HV-Rank	**52**(88)	**2806**(2394)	∞	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f18							
Rank	**1**(0)	1307(570)	∞	∞	∞	∞ *5001*	0/5
ND	**1**(0)	803(454)	2.1e4(4e4)	∞	∞	∞ *5001*	0/5
HV-Rank	**1**(0)	**149**(20)	**2234**(2070)	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f19							
Rank	59(48)	1383(530)	∞	∞	∞	∞ *5001*	0/5
ND	74(61)	2378(694)	∞	∞	∞	∞ *5001*	0/5
HV-Rank	**25**(20)	**1231**(1672)	**1.2e4**(5788)	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f20							
Rank	71(174)	3282(2746)	2.3e4(1e4)	∞	∞	∞ *5003*	0/5
ND	12(0)	3542(4397)	∞	∞	∞	∞ *5003*	0/5
HV-Rank	**8.0**(18)	**1525**(15)	**5018**(4390)	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f21							
Rank	**1**(0)	3922(4453)	2.2e4(4e4)	∞	∞	∞ *5005*	0/5
ND	**1**(0)	4733(7891)	2.3e4(6e4)	∞	∞	∞ *5003*	0/5
HV-Rank	**1**(0)	**2324**(3902)	**8737**(1e4)	**2.2e4**(2e4)	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f22							
Rank	**1**(0)	**2812**(698)	∞	∞	∞	∞ *5003*	0/5
ND	**1**(0)	6134(4480)	∞	∞	∞	∞ *5003*	0/5
HV-Rank	**1**(0)	2878(2674)	∞	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f23							
Rank	**1**(0)	2534(2099)	2.3e4(2e4)	∞	∞	∞ *5001*	0/5
ND	**1**(0)	1854(1284)	2.5e4(3e4)	∞	∞	∞ *5003*	0/5
HV-Rank	**1**(0)	**691**(728)	**6126**(7604)	∞	∞	∞ *5005*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f24							
Rank	6.6(7)	2.5e4(3e4)	∞	∞	∞	∞ *5003*	0/5
ND	**4.6**(9)	1.1e4(1e4)	∞	∞	∞	∞ *5005*	0/5
HV-Rank	**4.6**(9)	**1778**(1290)	∞	∞	∞	∞ *5005*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f25							
Rank	7.4(16)	5392(7502)	∞	∞	∞	∞ *5001*	0/5
ND	**4.2**(0)	5947(1e4)	∞	∞	∞	∞ *5003*	0/5
HV-Rank	**4.2**(8)	**2482**(2220)	∞	∞	∞	∞ *5005*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f26							
Rank	173(215)	1891(1017)	2.3e4(2e4)	∞	∞	∞ *5007*	0/5
ND	90(222)	2772(1120)	2.5e4(4e4)	∞	∞	∞ *5001*	0/5
HV-Rank	**24**(56)	**819**(994)	**2951**(3495)	∞	∞	∞ *5003*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f27							
Rank	**1**(0)	**2331**(1168)	∞	∞	∞	∞ *5005*	0/5
ND	**1**(0)	2633(431)	∞	∞	∞	∞ *5007*	0/5
HV-Rank	**1**(0)	3727(5095)	∞	∞	∞	∞ *5005*	0/5

Δf_{opt}	1e0	1e-1	1e-2	1e-3	1e-4	1e-5	#succ
f28							
Rank	**1**(0)	452(456)	6024(7582)	∞	∞	∞ *5003*	0/5
ND	**1**(0)	475(628)	**2097**(877)	∞	∞	∞ *5003*	0/5
HV-Rank	**1**(0)	**86**(33)	2237(568)	**7439**(2504)	∞	∞ *5005*	0/5

Table 1: Functions f_1 to f_{28}. Average runtime (aRT) to reach given targets, measured in number of function evaluations, in dimension 5. For each function, the aRT and, in braces as dispersion measure, the half difference between 10 and 90%-tile of (bootstrapped) runtimes is shown for the different target Δf-values as shown in the top row. #succ is the number of trials that reached the last target $\mathrm{HV_{ref}} + 10^{-5}$. The median number of conducted function evaluations is additionally given in *italics*, if the target in the last column was never reached. Entries, succeeded by a star, are statistically significantly better (according to the rank-sum test) when compared to all other algorithms of the table, with $p = 0.05$ or $p = 10^{-k}$ when the number k following the star is larger than 1, with Bonferroni correction by the number of instances. Best results are printed in bold.

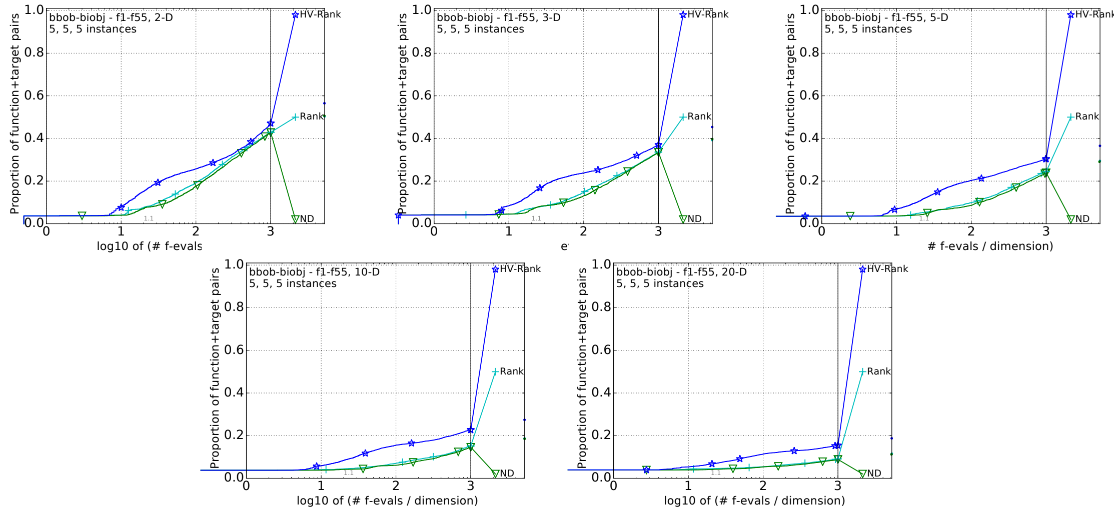

Figure 2: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 58 targets with target precision in $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ over all the problems in D$\in$ $\{2, 3, 5, 10, 20\}$.

Nsga-ii. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, Apr. 2002.

[8] D. Finkel and C. Kelley. Additive scaling and the direct algorithm. *Journal of Global Optimization*, 36(4):597–608, 2006.

[9] I. C. García, C. A. C. Coello, and A. Arias-Montaño. Mopsohv: A new hypervolume-based multi-objective particle swarm optimizer. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 266–273, July 2014.

[10] N. Hansen, A. Auger, O. Mersmann, T. Tusar, and D. Brockhoff. COCO: A Platform for Comparing Continuous Optimizers in a Black-Box Setting. *arXiv preprint arXiv:1603.08785*, 2016.

[11] N. Hansen, T. Tusar, A. Auger, D. Brockhoff, and O. Mersmann. COCO: Experimental Procedure. http://numbbo.github.io/coco-doc/experimental-setup/, 2016.

[12] S. Jiang, J. Zhang, Y.-S. Ong, A. N. Zhang, and P. S. Tan. A simple and fast hypervolume indicator-based multiobjective evolutionary algorithm. *IEEE Transactions on Cybernetics*, 45(10):2202–2213, Oct. 2015.

[13] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157, Oct. 1993.

[14] T. Tusar, D. Brockhoff, N. Hansen, and A. Auger. COCO: The Bi-objective Black Box Optimization Benchmarking (bbob-biobj) Test Suite. ArXiv e-prints, arXiv:1604.00359, Apr. 2016.

[15] L. Wang, H. Ishida, T. Hiroyasu, and M. Miki. Examination of multi-objective optimization method for global search using direct and ga. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 2446–2451, June 2008.

[16] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evolutionary Computation*, 7(2):117–132, 2003.

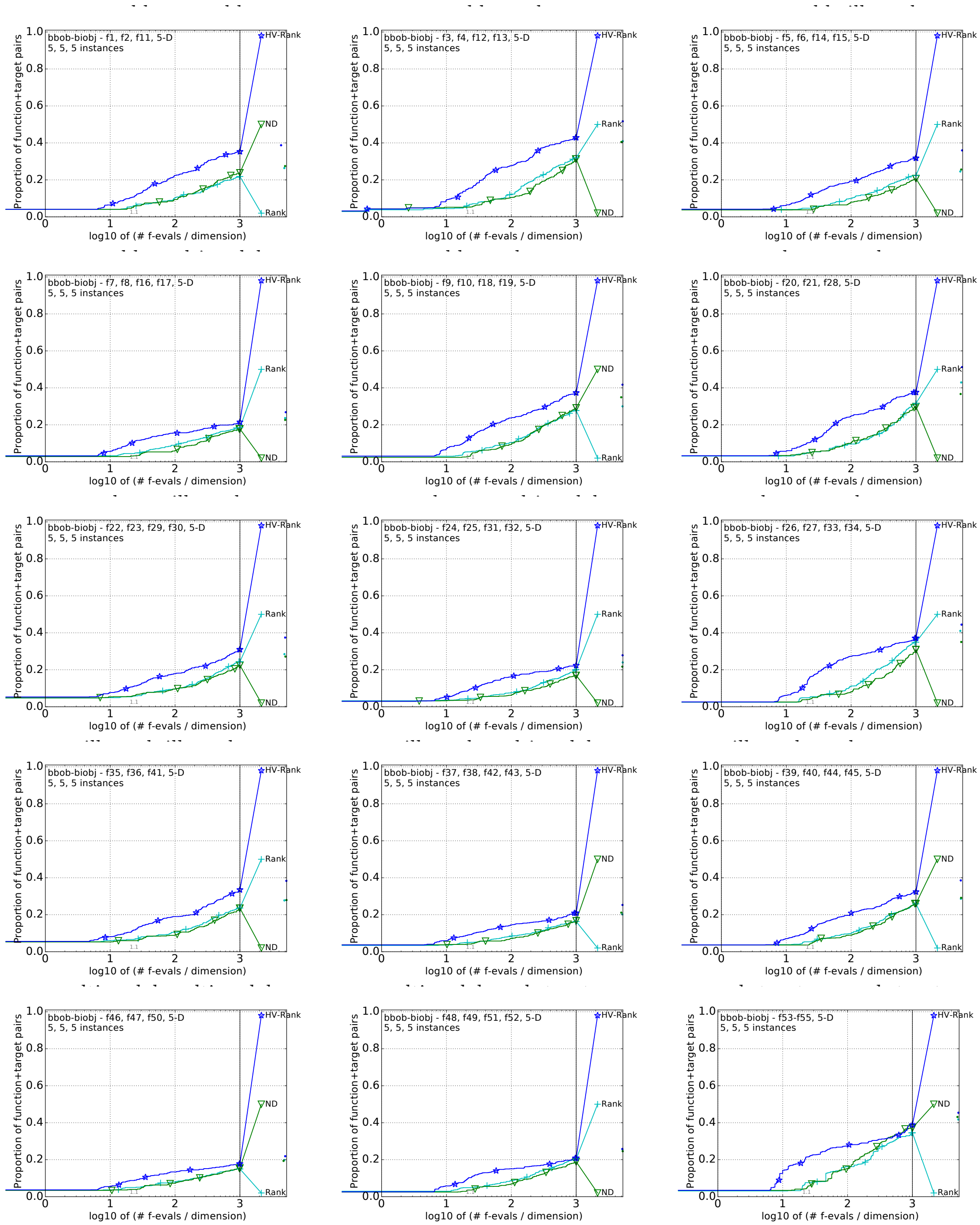

Figure 3: Bootstrapped empirical cumulative distribution of the number of objective function evaluations divided by dimension (FEvals/DIM) for 58 targets with target precision in $\{-10^{-4}, -10^{-4.2}, -10^{-4.4}, -10^{-4.6}, -10^{-4.8}, -10^{-5}, 0, 10^{-5}, 10^{-4.9}, 10^{-4.8}, \ldots, 10^{-0.1}, 10^{0}\}$ for all function subgroups in 5-D.