Preliminary study: Qualitative indicators in Multi-objective DIRECT framework

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

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

***Abstract*—DIRECT is known for balancing the exploration and exploitation of a search space. This paper seeks to explore the improvement of diversity among solutions through the use of qualitative indicators in multi-objective DIRECT framework. Three different indicators - Hypervolume (HV), Epsilon (EPS), R2 indicators are used in this study. The three variants of indicators are tested on the Black-box Multi-objective Optimization Benchmarking (BMOB) Platform. The results are presented and some insights in the choice of selection operator are provided. Overall, HV indicator performs the best followed by R2, then EPS. EPS indicator performs worse than HV and R2 in unimodal problems. Also, HV indicator achieves notably better results at high dimensions. R2 performs better than EPS in non-separable problems.**

I. Introduction

Multi-objective optimization problems (MOPs) are often modeled as black-box problems because the actual optimal solution in real life applications is almost always unknown. Most of the time in reality, we often settle for an approximate solution that is better than the rest of the solutions. Unlike single objective problems, where the best solution is the solution with minimum objective value (in the case of a minimization problem), the best solution in MOPs can not be determined as easily. Due to the conflicting objectives, MOPs have a set of optimal solutions instead of a single best 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. In a bi-objective problem, the set of solutions can be compared through a visualisation of their objective values in a 2D plot easily. However, in a many-objective problem, this can not be done as easily. In order to compare solutions, there is a need for us to measure the quality of solutions quantitatively to find out which solutions are better than the rest.

There are various qualitative indicators [1] that seeks to measure the quality of the solutions such as HV, EPS, generational distance, error ratio. Up to today, there is no clear indicator that is deemed superior to the others. On the other hand, there are studies [2], [3] that analyzes the correlation between these indicators, studying the similarities and differences. Six indicators, generational distance, EPS, spread, generalized spread, inverted generational distance and hypervolume were studied in [2]. The authors conclude that the six metrics show high consistencies when Pareto fronts are convex and certain contradictions on concave Pareto fronts.

With the surge of these indicators, the idea of using these indicators as selection operators too emerged. Popular metrics that were used include HV in HypE[4] and SMS-EMOA[5] and R2 in R2-EMOA[6] and R2-IBEA[7]. The metric EPS is also used in combination with HV in FV-EMOA [8]. HV returns a positive value of a non-dominated solution based on all the objectives and its neighboring non-dominated solutions, providing an accurate indicator on the diversity of all non-dominated solutions. Moreover, it has been previously verified that HV-based algorithms do perform better than classical methods experimentally [9].

Almost all, if not all, indicator-based algorithms were implemented in Evolutionary Algorithms (EAs). With the growing interest in mathematical-based algorithms, known for their mathematical tractability and convergence, we look to implement the use of qualitative indicators in mathematical-based algorithms such as `MO-DIRECT` [10]. Previously, the HV indicator has been implemented in the `MO-DIRECT` framework [11] and displayed better results than classical strategies. HV, despite performing well, requires a large computational time, which increases with the dimensions of the problem. Therefore, in this study, we extend the use of indicators in the `MO-DIRECT` framework [10] to the EPS and R2 indicator. These variants would be tested on the BMOB platform [12] and the results would be analyzed.

The rest of the paper is organized as follows. Section II briefly introduces the three indicators (HV, EPS and R2). Section III provides the background on `MO-DIRECT` framework. Section IV-A provides details on the experimental setup and the test suite. Section IV-B analyzes the results generated from the experiment. Section V provides a conclusion and some suggestions for future work.

II. Quantitative Indicators

This section briefly introduces the three indicators that has been chosen as selection operators for `MO-DIRECT` and how they are implemented. Section II-A, II-B and II-C will cover the HV, EPS and R2 indicators respectively.

A. Hypervolume (HV) indicator

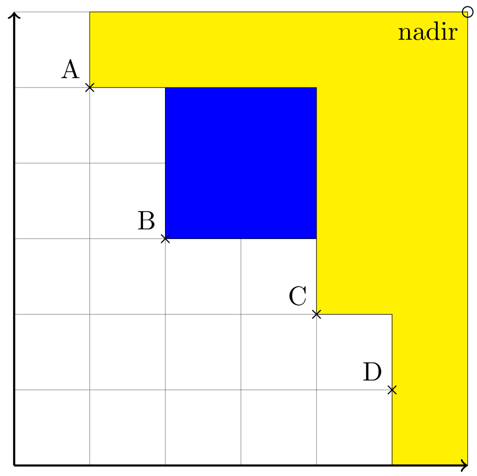

Fig. 1. Illustration of HV calculation

The HV indicator utilizes the nadir point of the optimal Pareto front as a reference point and calculates the amount of area that the set of solutions cover as illustrated in Figure 1. The bigger the area, the better the set of solutions. Thus, the bigger the value of the HV indicator, the better the set of solutions. In Figure 1, the value of hypervolume of all the solutions (A,B,C,D) is given by the combination of the shaded area (both blue and yellow).

However, in order to use the HV indicator as a selection operator, we require the HV contribution of each solution. We illustrate an example in Figure 1 as follows. To obtain the HV contribution of a solution (B), the HV of the set of solution excluding the solution (A,C,D) would be calculated, giving us the yellow area. This area would then be subtracted from the value of HV of the whole set (blue and yellow), giving us the HV contribution of the solution B in blue. Applying this concept on all the solutions would give only non-dominated solutions positive values of HV. All dominated solutions would be given a value of zero.

As the nadir point is unknown in a black-box optimization, it is estimated using the maximum value of the approximate set of solutions on the Pareto front, where its value is 110% of the value of the maximum value of each objective.

B. Epsilon (EPS) indicator

The EPS indicator used here is the unary additive EPS indicator, which requires a reference set to compare the approximate set of solutions. In a set of solutions, the value of EPS indicator is calculated as follows

$$eps = \max_{i=1}^{B} \min_{j=1}^{A} \max_{k=1}^{nObjs} (\mathbf{approx}(j,k) - \mathbf{ref}(i,k)) \quad (1)$$

where eps is the scalar EPS indicator, B is the number of points in the reference set, $\mathbf{ref}$, A is the number of solutions in the approximate set, $\mathbf{approx}$ and nObjs is the number of objectives.

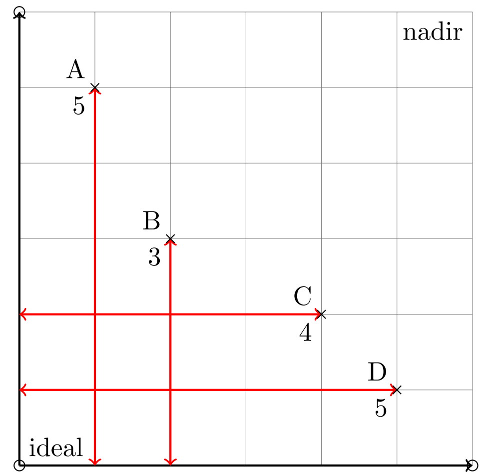

Fig. 2. Graphical illustration of EPS calculation of each solution in the approximate set

In order to calculate the EPS value of each solution, the $\mathbf{approx}$ is replaced with the solution instead. Figure 2 illustrates a simple example of EPS calculation graphically. In this 2-objective problem, the reference set is made up of 3 points (indicated by the circles) derived from the ideal and nadir points. The approximate set is made of the solutions A,B,C,D. Taking into consideration one solution (of the approximate set) at a time, the EPS value takes the maximum distance from the solution to the reference set, considering each objective separately. This in turn returns us the EPS value of 5,3,4,5 to solutions A,B,C,D respectively. The solution with the smallest EPS value is closest to the reference set, hence the best solution in the approximate set.

In a black-box optimization, the values of reference set, ideal and nadir points are unknown, thus they have to be estimated. The nadir point is estimated using the maximum value of the approximate set of solutions, where the nadir point is 110% of the value of the maximum value of each objective. The ideal point, on the other hand, is estimated using the minimum value of the approximate set of solutions, where the ideal point is 90% of the value of the minimum value of each objective. The reference set is then built based on the nadir and ideal point, where the ideal point is one of the points on the reference set and the other points are created through switching the value of one objective of ideal point with the nadir point. For example in the problem illustrated in Figure 2, the ideal point is (0,0) and the nadir point is (6,6). The reference set would consists of (0,0) (ideal point), (0,6) and (6,0) (through the switching of points).

C. R2 indicator

The R2 indicator uses a set of weights that reflects the relative importance of each objective. The weights can be adjusted based on the preference of the user. In this case, we vary the weights from 0 to 1 for each objective with equal increments, consisting of different combinations.

R2 first calculates the distance of each solution from the ideal point, then utilizes the different combinations of weights to generate a set of different values that measures the minimum distance from the set of solutions to the ideal point. From this set of values, the average is returned as the R2 value. This can be formally represented as

$$r2 = \frac{\sum_{a=1}^{N}(\min_{j=1}^{A} \max_{k=1}^{nObjs}(\mathbf{distance} * \mathbf{weight}))}{N} \quad (2)$$

where r2 is the value of the R2 indicator, N is the number of combinations of different weights in the vector **weight**, A is the number of solutions in the approximate set, nObjs is the number of objectives and **distance** is a vector consisting the distance between each solution in the approximate set and the ideal point.

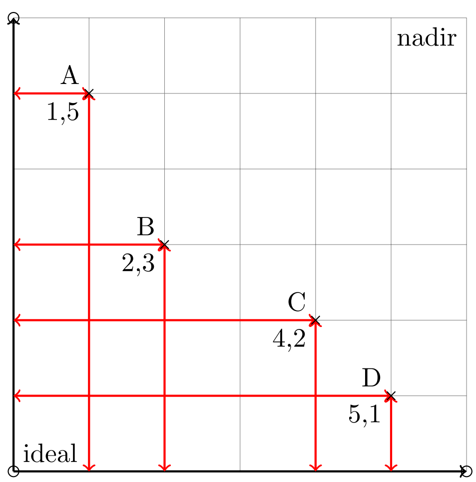

Fig. 3. R2 calculation

For example, given the weights for objective 1 and 2 in Figure 3 is 0.5 each, the value for this set of weights would be 1.5 ($0.5 * 3$ coming from solution B).

R2 contributions of each solution, however is computed differently. The R2 contribution of a solution is essentially, the sum of the set of values that measures the minimum distance from the set of solutions excluding the solution to the ideal point. This is formally represented as

$$r2_p = \sum_{a=1}^{N}(\min_{j=1}^{C} \max_{k=1}^{nObjs}(\mathbf{distance} * \mathbf{weight})) \quad (3)$$

where $r2_p$ is the value of the R2 contribution of solution p, N is the number of combinations of different weights in the vector **weight**, C is the number of solutions in the approximate set excluding solution p, nObjs is the number of objectives and **distance** is a vector consisting the distance between each solution in the approximate set (excluding solution p) and the ideal point. Therefore, the greater the value of R2 contribution, $r2_p$, the better the solution p is.

Similarly, the ideal point is estimated using the minimum value of the approximate set of solutions, where the ideal point is 90% of the value of the minimum value of each objective.

III. MO-DIRECT FRAMEWORK

Section III-A first explains the paritioning procedure of hyperrectangles used in `MO-DIRECT`. Then, Section III-B provides the framework of `MO-DIRECT`. Finally, Section III-C proposes a formal algorithm to implement the different indicators.

A. 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

$$w_j = \frac{1}{\min_{k\in\{1,-1\}} ||\mathbf{f}(\mathbf{c}_i + k\cdot\delta\cdot\mathbf{e}_j) - \mathbf{f}(\mathbf{c}_i)||}\ , \quad (4)$$

where w_j represents the inverse of spread/diversity between the newly sample solution and the original solution. The measure of diversity is measured using the minimum distance between the objective values of a newly sampled solution $\mathbf{f}(\mathbf{c}_i + \delta\cdot\mathbf{e}_j)$ and the original solution $\mathbf{f}(\mathbf{c}_i)$. The partitioning then continues 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.

B. `MO-DIRECT` Framework

The framework of `MO-DIRECT` provided in Algorithm 1 essentially consists of two parts: the process of selecting potentially optimal hyperrectangles and dividing them into smaller hyperrectangles using the partitioning procedure in Section III-A. In this paper, we focus on the process of selecting potentially optimal hyperrectangles using different indicators. Besides the indicators, the size of the hyperrectangles, represented by σ_i, too plays a role in the selection to ensure that there is some exploration of the search space. σ_t is a parameter that is set by the user, representing the smallest size the hyperrectangle can get.

Algorithm 1 MO-DIRECT framework adapted from [10]

Input: vectorial function to be minimized $\mathbf{f}$,
search space $\mathcal{X}$,
evaluation budget v,
hyperrectangle threshold σ_t
Output: approximation set of $\min_{\mathbf{x}\in\mathcal{X}} \mathbf{f}(\mathbf{x})$, $\mathcal{Y}_*^v$
Initialisation : $\mathcal{H}_1 = \{\mathcal{X}\}$
while evaluation budget v is not exhausted **do**
  Evaluate all the new hyperrectangles $\in \mathcal{H}_t$.
  Choose potentially optimal hyperrectangles using indicators introduced Section II.
  Partition the hyperrectangles in $\mathcal{I}_t$ according to the procedure outlined in Section III-A using Eq. (4).
  $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t \setminus \mathcal{I}_t \cup \{\mathcal{I}_t$'s newly generated hyperrectangles$\}$
  $t \leftarrow t+1$
end while
return $nondominated\big(\{\mathbf{f}(\mathbf{c}_i)\}_{i\in\mathcal{H}_t}\big)$

Algorithm 2 Indicator-based MO-DIRECT

Input: vectorial function to be minimized $\mathbf{f}$,
search space $\mathcal{X}$,
evaluation budget v,
hyprrectangle threshold σ_t
Output: approximation set of $\min_{\mathbf{x}\in\mathcal{X}} \mathbf{f}(\mathbf{x})$, $\mathcal{Y}_*^v$
Initialisation : $\mathcal{H}_1 = \{\mathcal{X}\}$
1: **while** evaluation budget v is not exhausted **do**
2:   Evaluate all the new hyperrectangles $\in \mathcal{H}_t$.
  $\mathcal{P} \leftarrow \mathbf{f}(\mathbf{c}_i) : i \in \mathcal{H}_t\ , \sigma_i \geq \sigma_t$.
3:   **if** size $(\mathcal{P}) > 2$ **then**
4:     Calculate $indicator_j : j \in \mathcal{P}$ and $indicator_{\mathcal{P}}$.
    $\mathcal{I}_t \leftarrow nondominated(\{(indicator_j, \sigma_j) : j \in \mathcal{P}\ \})$.
5:     **if** small change in $indicator_{\mathcal{P},t}$ and $indicator_{\mathcal{P},t-1}$ **then**
6:       $rank_i = nondominatedsort(\{\mathbf{f}(\mathbf{c}_i) : i \in \mathcal{H}_t\ , \sigma_i \geq \sigma_t\})$.
      $\mathcal{I}_t = nondominated(\{(rank_i), \sigma_i) : i \in \mathcal{H}\ , \sigma_i \geq \sigma_t\})$.
7:     **end if**
8:   **else**
9:     $\mathcal{I}_t = \mathcal{P}$
10:   **end if**
11:   **return** $\mathcal{I}_t$.
  Partition the hyperrectangles in $\mathcal{I}_t$ according to the procedure outlined in Section III-A using Eq. (4)
  $\mathcal{H}_{t+1} \leftarrow \mathcal{H}_t \setminus \mathcal{I}_t \cup \{\mathcal{I}_t$'s newly generated hyperrectangles$\}$
  $t \leftarrow t+1$
12: **end while**
13: **return** $nondominated\big(\{\mathbf{f}(\mathbf{c}_i)\}_{i\in\mathcal{H}_t}\big)$

C. Indicator-based Algorithm

Previously in [11], it has been experimentally verified that using HV indicator would reduce the exploratory nature of MO-DIRECT, hence causing it to be stuck in the local optimum. To avoid this situation, we propose to use the MO-DIRECT-hv algorithm [11] and adapt it for the use of other indicators. The generic algorithm is formally presented in Algorithm 2. More details on the rank strategy (Line 6) can be found in [11]. In order to determine when to use the rank strategy, the indicator value of the whole set is required in each iteration (t), which is represented by $indicator_{\mathcal{P},t}$. In relation to the previous examples in Section II, $HV_{\mathcal{P},t}$ would be the total area (blue and yellow) in Figure 1, $eps_{\mathcal{P},t}$ the value returned by Equation 1 and $r2_{\mathcal{P},t}$ is the value returned by Equation 2. The indicator value of each solution is represented by $indicator_j$. Similarly, HV_B would be the area in blue in Figure 1, eps_j the values illustrated in Figure 2 and $r2_j$ returned by Equation 3.

IV. Numerical Experiments

A. Setup

The numerical experiments are set up according to [12], where each algorithm is run on 100 multi-objective problems categorized over seven groups: low-dimensional, high-dimensional, separable, non-separable, uni-modal, multi-modal, and mixed categories.

The procedure for assessing the solution quality of an algorithm is based on recording its *runtime*: the number of function evaluations required by the algorithm for its solution to reach a specific (target) quality value. The recorded runtimes are then expressed in terms of data profiles, which capture various aspects of the algorithms' convergence behavior. For more details, one can refer to [12].

B. Results

The results in Figure 4 indicates that the HV indicator performs the best overall, followed by R2 and EPS indicator. The main difference in the calculation of individual indicator values between HV, EPS and R2 is that HV and R2 considers the whole set of approximate solutions when calculating indicator values of each solution while EPS only takes into account the solution that is being evaluated with respect to the reference set. By not taking into consideration its neighbouring solutions, the indicator does not take into account the spread of the solution set. This may in turn lead selecting many solutions and wasting evaluations. This is reflected in the overall results where EPS does not perform as well as HV and R2.

The results are then analyzed following the classification of problems in terms of

- Modality:
  Looking at the uni-modal plots in Figure 6, EPS reflected the worst performance out of the three indicators. Going back to how EPS indicator is formulated in Section II-B, it can be agreed that there is a bias in the selection of the best solution towards the middle of the Pareto front. This in turn reduces the probability of finding solutions at the ends of the Pareto front as EPS would always indicate the solutions in the middle of Pareto front as better solutions. When it reaches the real Pareto front

(when change in EPS is small), the algorithm samples new solutions using the rank strategy instead, leading to less efficient sampling compared to the other indicators. On the other hand, a similar data profile is observed for all indicators in the multi-modal plots of Figure 6. In multi-modal problems, the spread of solutions tend to be in clusters, requiring more global search, in other words, the use of the rank strategy. This in turn reduces the effect of the comparably less efficient sampling of EPS in search of solutions at the ends of the Pareto front. Hence, allowing EPS to perform as well as HV and R2.

- Dimensionality:
  In the high-dimensional plots of Figure 6, it can be observed that HV generally outperforms R2 and EPS indicators. R2 and EPS are distance-based indicators, which only takes into consideration the distance that has the maximum distance from the ideal point or reference set in that particular objective. On the other hand, HV calculates a more accurate measure, returning the hypervolume contribution of each solution, which takes into account the distance in every objective in the problem which in turn provides a more complete representation of the quality of each solution. The benefits of this increased accuracy is clearly reflected in high-dimensional problems.
- Separability:
  Another observation that can be made in Figure 6 is that R2 does better than EPS in non-separable problems. Non-separable problems are problems that are not separable in their objectives. R2 takes varying weights between the objectives, hence better able to handle non-separable problems compared to EPS.

From Figure 6, we also observe that there is a sudden jump in the proportion of targets met between 10^2 and 10^3 in all plots except for the multi-modal plots. This observation could account for the change in strategy for selection of potentially optimal points when the improvement in indicator values for the set of solutions is small. This, however, is not reflected in the multi-modal plots as it is more likely for `MO-DIRECT` to switch strategy more frequently in multi-modal problems, allowing a smoother increase in hitting target values.

C. Empirical Runtime Evaluation

In order to evaluate the complexity of the algorithms (measured in runtime), the algorithms are run on a representative set of the problems. The empirical complexity of an algorithm is then computed as the running time (in seconds) of the algorithm summed over all the problems given divided by the total number of function evaluations used. The results for three different evaluation budgets are shown in Figure 5. All indicators are implemented on the MATLAB platform. HV is implemented using an efficient method that is provided in [8]. However, HV still requires a longer runtime due to its relatively higher complexity in computation, which increases with respect to the number of objectives.

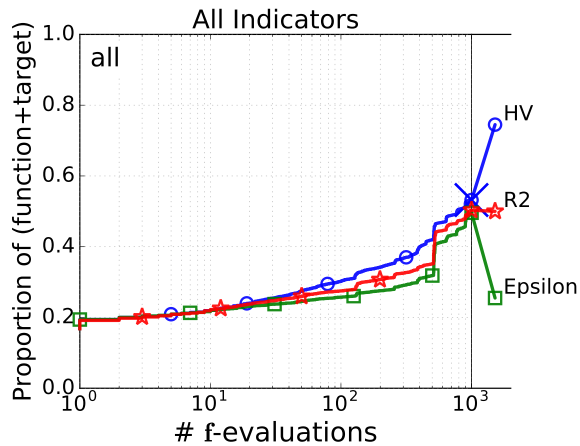

Fig. 4. Empirical cumulative distribution function of the observed number of function evaluations in which the y-axis tells how many of 70 targets - over the set of problems and quality indicators, have been reached by each algorithm for a given evaluation budget (on x-axis). Data profiles aggregated over all the problems across all the quality indicators computed for each of the compared algorithms. The symbol $\times$ indicates the maximum number of function evaluations.

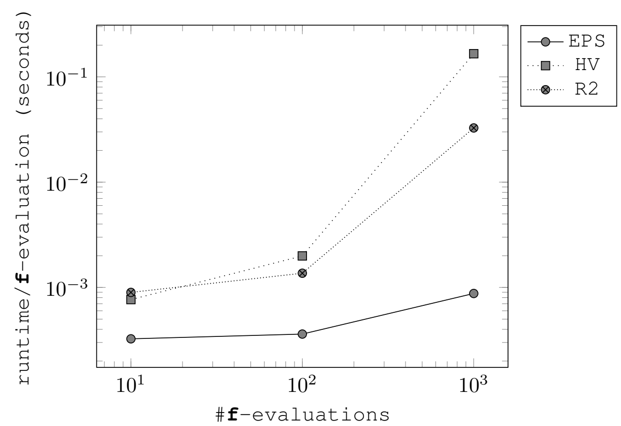

Fig. 5. A *semi*-log plot visualizing the runtime per one function evaluation (in seconds) of the compared algorithms. All the algorithms were run on a selected set of problems over a set of evaluation budgets, namely BK1, DPAM1, L3ZDT1, DTLZ3, and FES3; with an evaluation budget $\in \{10, 100, 1000\}$ per problem on a PC with: 64-bit Windows 7, Intel(R) Xeon(R) CPU E5-1650@ 3.20GHz with 1 processor and 6 cores

V. Conclusion

Different indicators are implemented as selection operators in a deterministic algorithm, using the `MO-DIRECT` framework. This allows a fair comparison between the use of selection operators.

With the results, we are not able to clearly conclude which indicator is best suited as a selection operator. However, we are able to gain several insights on the `MO-DIRECT` and the selection operators. Firstly, indicators (HV, R2) that takes into consideration the whole approximate set of solutions tend to perform better. EPS indicator performs worse than HV and R2 in uni-modal problems. HV indicator performs better than others in high dimensions. Also, R2 performs better than EPS in non-separable problems. Last but not least, the results too illustrate how the proposed indicator-based MO-DIRECT searches.

Through running these variants of indicator-based MO-DIRECT on BMOBench, some general trends were found. However, in order to gain deeper insights and validate the current insights, a study that allows the visualisation of Pareto front, similar to [13], has to be performed. Additionally, in order to get a better idea which indicator is a better selection operator, the experiment has to be run on a greater evaluation budget.

ACKNOWLEDGMENT

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

REFERENCES

[1] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

[2] S. Jiang, Y. S. Ong, J. Zhang, and L. Feng, "Consistencies and contradictions of performance metrics in multiobjective optimization," *IEEE Transactions on Cybernetics*, vol. 44, no. 12, pp. 2391–2404, Dec. 2014.

[3] A. Liefooghe and B. Derbel, "A correlation analysis of set quality indicator values in multiobjective optimization," in *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, ser. GECCO '16. New York, NY, USA: ACM, 2016, pp. 581–588. [Online]. Available: http://doi.acm.org/10.1145/2908812.2908906

[4] J. Bader and E. Zitzler, "Hype: An algorithm for fast hypervolume-based many-objective optimization," *Evol. Comput.*, vol. 19, no. 1, pp. 45–76, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1162/EVCO_a_00009

[5] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653–1669, sep 2007. [Online]. Available: http://dx.doi.org/10.1016/j.ejor.2006.08.008

[6] D. Brockhoff, T. Wagner, and H. Trautmann, "2 indicator-based multiobjective search," *Evolutionary Computation*, vol. 23, no. 3, pp. 369–395, Sep. 2015.

[7] D. H. Phan and J. Suzuki, "R2-ibea: R2 indicator based evolutionary algorithm for multiobjective optimization," in *2013 IEEE Congress on Evolutionary Computation*, Jun. 2013, pp. 1836–1845.

[8] 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*, vol. 45, no. 10, pp. 2202–2213, Oct. 2015.

[9] D. Brockhoff, T. Friedrich, and F. Neumann, "Analyzing hypervolume indicator based algorithms," *Parallel Problem Solving from Nature PPSN X*, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87700-4_65

[10] A. Al-Dujaili and S. Suresh., "Dividing rectangles attack multi-objective optimization." in *IEEE Congress on Evolutionary Computation (CEC)*, Vancouver, Canada, July 2016.

[11] S. S. Cheryl Wong Sze Yin, Abdullah Al-Dujaili, "Hypervolume-based direct for multi-objective optimisation," in *GECCO 2016 (Companion)*. ACM, 2016.

[12] A. Al-Dujaili and S. Suresh, "BMOBench: Black-box multi-objective optimization benchmarking platform," *ArXiv e-prints*, vol. arXiv:1605.07009, 2016.

[13] C. A. Rodríguez Villalobos and C. A. Coello Coello, "A new multi-objective evolutionary algorithm based on a performance assessment indicator," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '12. New York, NY, USA: ACM, 2012, pp. 505–512. [Online]. Available: http://doi.acm.org/10.1145/2330163.2330235

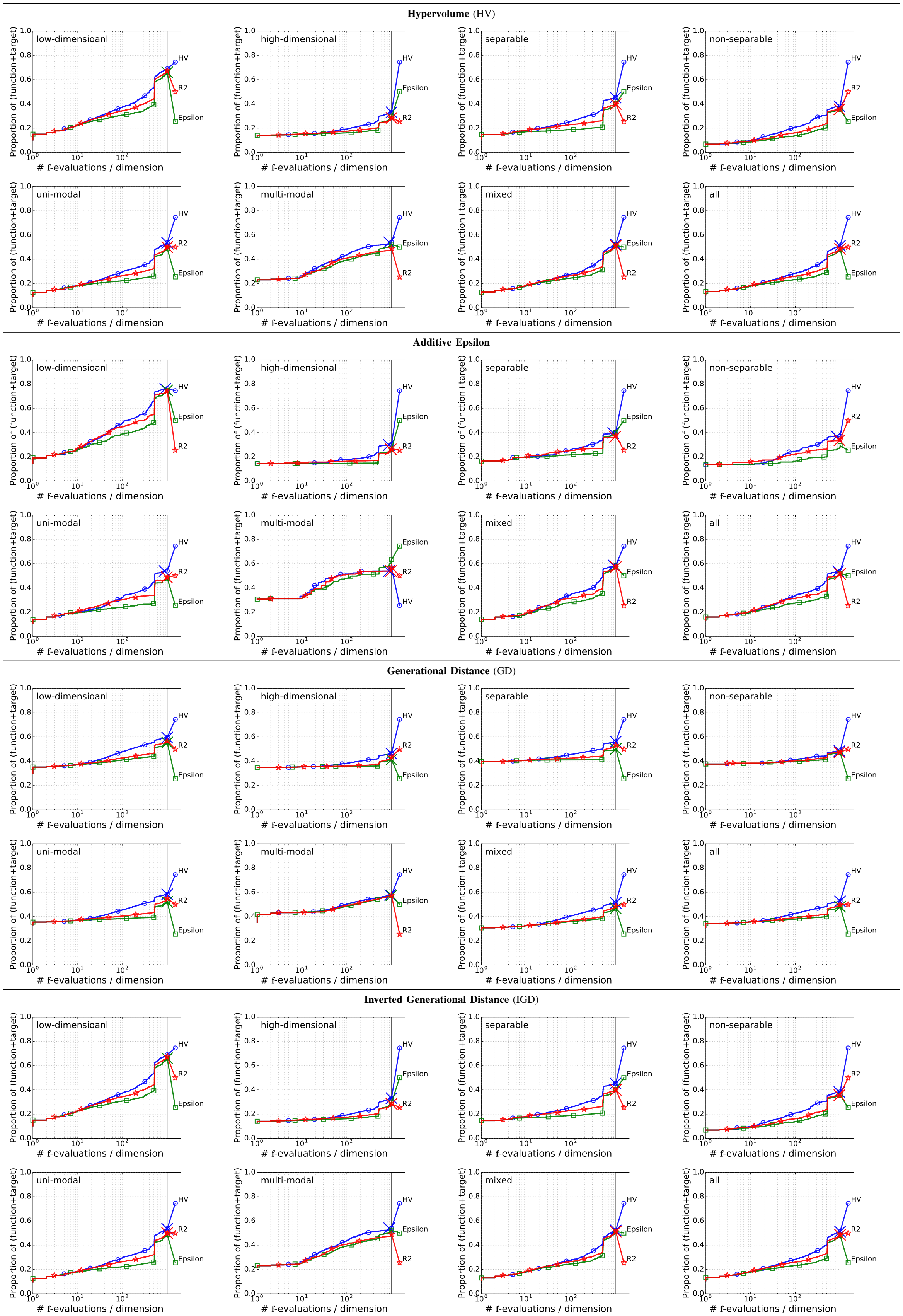

Fig. 6. Empirical cumulative distribution function of the observed number of function evaluations in which the y-axis tells how many of 70 targets - over the set of problems and quality indicators, have been reached by each algorithm for a given evaluation budget (on x-axis). Data profiles aggregated over problem categories for each of the quality indicators computed. The symbol $\times$ indicates the maximum number of function evaluations.