A Critical Review of "A Practical Guide to Select Quality Indicators for Assessing Pareto-Based Search Algorithms in Search-Based Software Engineering": Essay on Quality Indicator Selection for SBSE

Miqing Li
CERCIA, School of Computer Science, University of Birmingham, UK
m.li.8@cs.bham.ac.uk

Tao Chen
Department of Computing and Technology, Nottingham Trent University, UK;
CERCIA, School of Computer Science, University of Birmingham, UK
t.chen@cs.bham.ac.uk

Xin Yao
Department of Computer Science and Engineering, Southern University of Science and Technology, China;
CERCIA, School of Computer Science, University of Birmingham, UK
x.yao@cs.bham.ac.uk

ABSTRACT

This paper presents a critical review of the work published at ICSE'2016 on a practical guide of quality indicator selection for assessing multiobjective solution sets in search-based software engineering (SBSE). This review has two goals. First, we aim at explaining why we disagree with the work at ICSE'2016 and why the reasons behind this disagreement are important to the SBSE community. Second, we aim at providing a more clarified guide of quality indicator selection, serving as a new direction on this particular topic for the SBSE community. In particular, we argue that it does matter which quality indicator to select, whatever in the same quality category or across different categories. This claim is based upon the fundamental goal of multiobjective optimisation — supplying the decision-maker a set of solutions which are the most consistent with their preferences.

CCS CONCEPTS

• **Software and its engineering → Search-based software engineering**;

KEYWORDS

Multiobjective optimisation, search-based software engineering, quality assessment, quality indicator selection

ACM Reference Format:
Miqing Li, Tao Chen, and Xin Yao. 2018. A Critical Review of "A Practical Guide to Select Quality Indicators for Assessing Pareto-Based Search Algorithms in Search-Based Software Engineering": Essay on Quality Indicator Selection for SBSE. In *ICSE-NIER'18: 40th International Conference on Software Engineering: New Ideas and Emerging Results Track, May 27-June 3, 2018, Gothenburg, Sweden.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3183399.3183405

ICSE-NIER'18, May 27-June 3, 2018, Gothenburg, Sweden

ACM ISBN 978-1-4503-5662-6/18/05…$15.00
https://doi.org/10.1145/3183399.3183405

1 INTRODUCTION

The growing interest in simultaneously dealing with multiple objectives in software engineering results in a significant number of search methods, e.g., various heuristics and metaheuristics, for a broad range of problems [2, 10, 11]. These multiobjective search methods typically aim to generate a set of representative solutions to the whole Pareto optimal front from which the decision-maker (DM) can choose their favourite one. This raises an important research topic — how to evaluate and compare the quality of solution sets generated by different search methods; i.e., which kind of solution set is more likely to be preferred by the DM.

Over the last two decades, a large number of quality indicators have been emerging in fields of evolutionary computation [20], mathematical optimisation [12], and operations research [1]. Among them, many have already been frequently used in Search-Based Software Engineering (SBSE), such as Hypervolume (HV) [19] and ϵ-indicator [20]. However, the variety of quality indicators may overwhelm the user in the SBSE community, as every indicator works solely on a specific quality aspect of solution sets. This leads to a practical issue of how to select quality indicators to properly evaluate solution sets in SBSE.

The work of Wang et al. [15] (in ICSE'2016) is a notable endeavour to address this important issue. They attempted to provide a practical guide for the SBSE practitioners to select quality indicators for assessing which search method is "better". The authors first divided eight most frequently-used quality indicators in SBSE into four categories: *Convergence*, *Diversity*, *Combination* and *Coverage*. Then, through extensively empirical investigations they have drawn several fundamental conclusions about the selection of quality indicators. For example, they have concluded that it dose not matter which indicators to select within the same *Convergence* or *Combination* category, and also it dose not matter which indicators to select across *Convergence* and *Coverage* categories. Finally, they summarised a guide on how to select indicators in SBSE.

Wang et al.'s paper represents a growing recent development on this particular research topic; since 2016, it has been followed and exploited by a good few SBSE research groups for different problems, e.g., in [10] and [11]. However, we argue that there are discrepancies between Wang et al.'s work and the general goal of multiobjective optimisation, in terms of both the analytical method

and the conclusions, which may mislead the SBSE community. We feel that respectful scientific debates are very important for sustainable research, particularly in such an interdisciplinary topic where research from the well-established community of multiobjective optimisation may still be relatively new to the software engineering researchers. Indeed, explicit criticism may timely reveal the opposing ideas and can often excite significant growth of the research field (e.g., see [9]). The above motivates this essay with two goals. First, we aim at explaining why we disagree with Wang et al.'s paper and why the reasons behind this disagreement are important to SBSE. Second, we aim at providing the SBSE community a new, but more clarified guide of quality indicator selection and design, based upon information availability of the DM.

We start by discussing Wang et al.'s paper (Section 2). We show that some incomprehensive observations from Wang et al.'s paper are due to an inaccurate classification of the quality indicators studied (Section 3.1). Then, we show that even if an accurate classification of quality indicators is made, one still cannot ever draw the conclusions like one quality indicator being able to replace another (Section 3.2). This is because there is no equivalence between a (or a group of) quality indicator and the outperformance relation between solution sets. Finally, we explain that a reasonable selection of quality indicators should be in line with the preferences of the DM, and accordingly provide a more clarified guide on indicator selection with or without the DM's preferences (Section 4).

2 BRIEF OF WANG ET AL.'S WORK

Presented at the ICSE'2016 [15], Wang et al.'s work chose eight commonly-used quality indicators in SBSE and placed them into four categories, *Convergence*, *Diversity*, *Combination* of convergence and diversity, and *Coverage*. The authors then tested these indicators in three industrial problems (test suite minimisation, test case prioritisation and requirements allocation), and from that they have drawn the following main conclusions:

- For the category *Convergence* or *Combination*, it does not matter which indicator within the same category to select; however, it does matter for the category *Diversity*.
- It does matter to select indicators across the categories except for *Convergence* and *Coverage*.

Finally, on the basis of the above observations, a guide of how to select indicators has been provided for the SBSE community.

3 WHY WANG ET AL.'S WORK IS MISGUIDED

This section isolates two crucial points in Wang et al.'s paper which can be misguided: the classification of the quality indicators and the conclusions of indicator selection.

3.1 Misguided Categories of Quality Indicators

Wang et al.'s paper has considered eight quality indicators and classified them into four categories. They are {*Convergence*: GD [13], ED [3], ϵ [20]}, {*Diversity*: GS [17], PFS [5]}, {*Combination* of convergence and diversity: IGD [4], HV [19]}, and {*Coverage*: C [19]}. Convergence refers to how close a solution set is to the Pareto front. Diversity refers to how well the solutions in a set are distributed; it can be further divided into solutions' uniformity and spread. Coverage refers to how well a solution set covers the Pareto front, which is similar to the concept of the spread of Pareto front [12].

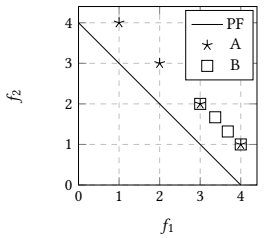

Figure 1: An example that illustrates the ϵ-indicator being capable of reflecting the diversity (spread) of a solution set ($\epsilon(A) = 1 < \epsilon(B) = 3$), while PFS and C not ($PFS(A) = PFS(B) = 4$, $C(A, B) = C(B, A) = 0.5$).

We question the classification that the indicator ϵ falls into *Convergence*, PFS into *Diversity*, and C into *Coverage*. The indicator (additive) ϵ of a solution set to the Pareto front measures the minimum value that can be added to each point in the Pareto front such that it can be weakly dominated by (i.e., inferior to or equal to) at least one solution in the evaluated set. In addition to evaluating convergence, the ϵ-indicator can also measure diversity of a solution set. Figure 1 gives an example that two solution sets A and B have the same convergence to the Pareto front, but different diversity among their solutions. The ϵ-indicator evaluates A significantly better than B ($\epsilon(A) = 1 < \epsilon(B) = 3$) since the upper-left points of the Pareto front need to move far away to make them be weakly dominated by a point in B.

The indicator PFS, which counts the number of nondominated solutions in a set (i.e., cardinality), cannot evaluate the diversity of the set. Applying the example of Figure 1, both A and B have four nondominated solutions (so $PFS(A) = PFS(B) = 4$), but A's solutions are diversified better than B.

Given two solution sets, the indicator C measures the proportion of solutions of one set that are weakly dominated by at least one solution of the other set. The C result does not reflect the diversity or coverage difference between two sets, but it can *partially* reflects their convergence difference as the dominance relation does not tell how much better one is superior to the other.

In addition to the above three indicators, the indicator ED only partially reflects the convergence of a solution set since it considers the closest solution of the set to the ideal point of the Pareto front. The indicator GS only partially reflects the spread of a solution set with more than two objectives since the closest distance between solutions in the set fails to indicate how well the set covers in a high-dimensional space [8]. Summarising the above, Table 1 provides the quality aspect(s) that the eight quality indicators actually reflect in evaluating multiobjective solution sets.

An accurate classification of quality indicators is of high importance. It tells people how a solution set performs on a specific

Table 1: The eight most frequently-used quality indicators in SBSE, as surveyed in Wang et al.'s paper. Diversity consists of spread (i.e., coverage) and uniformity. "+" means that the indicator can well reflect a specific quality feature and "−" means that the indicator can partially reflect that specific quality feature.

	GD	ED	ϵ-indicator	GS	PFS	IGD	HV	C
Convergence	+	−	+			+	+	−
Spread			+	−		+	+	
Uniformity			+	+		+	+	
Cardinality			−		+	−	−	

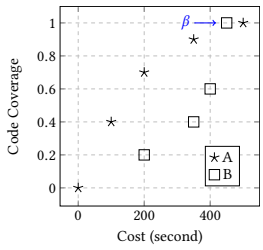

Figure 2: Two nondominated solutions sets, A and B, for optimising the code coverage and the cost of testing time [16]. A is evaluated better than B on all the eight indicators considered in Wang et al's paper: $GD(A) = 0.02 < GD(B) = 0.26$, $ED(A) = 0.5 < ED(B) = 0.89$, $\epsilon(A) = 0.1 < \epsilon(B) = 0.3$, $GS(A) = 0.15 < GS(B) = 0.46$, $PFS(A) = 5 > PFS(B) = 4$, $IGD(A) = 0.02 < IGD(B) = 0.27$, $HV(A) = 0.77 > HV(B) = 0.43$, $C(A) = 0.8 > C(B) = 0.25$. However, the DM may be more interested in B (specifically solution β) if they favour the full code coverage and then possible low cost.

quality aspect. In Wang et al.'s paper, the authors have observed inconsistent results obtained by GS and PFS, and thus concluded that it matters which indicator to select in the category *Diversity*. We think that an important reason for this observation is that PFS in fact does not reflect the diversity of a solution set (but rather the cardinality), and it is likely for solution sets to perform differently in distinct quality aspects.

3.2 Misguided Selection of Quality Indicators

In this section, we discuss the conclusions on quality indicator selection derived from Wang et al.'s study. We argue that even if an accurate classification of quality indicators is made, we still cannot draw the conclusions that it does not matter which indicator to select, whatever in the same category or across different categories.

In multiobjective optimisation, the general goal for the algorithm designer is to supply the DM a set of solutions which are the most consistent with their preferences. When the DM's preferences are not known *a priori*, the quality features that an indicator measures for is merely an assumption of the DM's possible preferences. A solution set being evaluated better than another by an indicator means that the former is superior under the assumption that the indicator reflects the DM's preferences, but not certainly being preferred. This also applies to the combination of several indicators. Figure 2 gives an example on the software test cases generation problem [16]. As shown, the set A is evaluated better than the set B by all the eight quality indicators considered in Wang et al.'s paper. However, depending on the contexts, the DM might first favour the full code coverage and then possible low cost [16]. This will lead to set B to be of more interest, as it has the most preferred solution (β) that achieves full coverage and lower cost than the ones in A.

The above example indicates that without considering the DM's preferences (if existing), we may use inappropriate indicators for the quality features which the DM does not care about. From the perspective of set-based comparison, the underlying reason behind this is that the two sets in the example are effectively incomparable. Next, we introduce an important set-based relation in multiobjective optimisation (which has been missing in Wang et al.'s paper). Suppose there are two solution sets A and B:

Relation 1. *[Better relation between two sets [20]] We say that A is better than B (denoted as $A \lhd B$) if for every solution $b \in B$ there exists at least one solution $a \in A$ that weakly dominates b, but there exists at least one solution in A that is not weakly dominated by any solution in B.*

The *better* relation represents the most general form of superiority between two solution sets; in other words, $A \lhd B$ means that A is at least as good as B while B is not so good as A (thus A always being preferred by the DM). Unfortunately, there is no equivalence between a (or a group of) quality indicator and the *better* relation, which has been proven by Zitzler et al. [20]. This means no matter how many indicators we use, we cannot guarantee that the better-evaluation-result solution set is certainly preferred by the DM. Back to the example in Figure 2, there exists no solution in A that dominates the solution β, so $A \not\lhd B$. We thus cannot say A being superior to B, despite that A outperforms B on all the quality aspects, i.e., convergence, spread, uniformity, and cardinality.

4 A CLARIFIED GUIDE

In this section, we provide new, more clarified guidelines on how to select quality indicators to evaluate multiobjective solution sets for SBSE. Such a guide, as discussed before, needs to be in line with the DM's preferences. When the DM's preferences are unavailable, the set-based relation $\lhd$ is the simplest comparison method to check whether a set is better than another, and it generally meets any preference potentially articulated by the DM. However, the $\lhd$ relation may leave sets incomparable; in fact, in most cases, two sets under consideration are nondominated to each other. This necessitates quality indicators which represent certain assumptions about the DM's preferences. In general, when the DM's preferences are unknown, a set of solutions which well represent the Pareto front are desirable as the DM is likely to find their interested solution from them. Therefore, quality indicators have arrived to reflect this "representation" to the Pareto front, which often involve several quality aspects — convergence, spread, uniformity, and cardinality.

When articulation of the DM's preferences is clear, such as the situation that a weight for each objective can be explicitly specified, quality indicators need to be selected, or even designed, directly according to the preferences. However, in the software engineering domain, it is not uncommon that the DM may experience difficulty in precisely articulating their preferences. The DM may only be able to provide some vague preference information such as a fuzzy region around one point or a set of weights in certain space, or they are more interested in some parts of the Pareto front (e.g., knee). As such, quality indicators need to be selected or designed in accordance with different situations.

Below we summarise four general situations of how to select/design quality indicators based on the availability of DM's preferences.

(1) When articulation of the DM's preferences is clear, quality indicators can be easily selected or designed according to the preferences. Taking the problem in Figure 2 as an example, an indicator that hierarchically compares the code coverage and then the cost of test cases can be used to evaluate solution sets. Such a hierarchical indicator is also useful for the software product line configuration problem [6, 10] where the Feature Model's dependency compliance is always more important than the richness of the model; thus only the solutions that achieve full dependency compliance are of interest. This is obvious, as violation of dependency implies faulty configuration, which has no value in practice.

(2) When articulation of the DM's preferences is vague/rough, quality indicators need to be selected or designed to incorporate the preferences. Some indicators, e.g., HV [19] and IPF [1], allow to be integrated with a set of biased weights to reflect the DM's preferences [1, 18]. They use a set of uniformly-distributed weights (often in the unit simplex) to represent the whole Pareto front, and then restricts the weight space according to some partial information of the DM's preferences. Those indicators are most likely to be useful for the software performance management [11], where the preferences may be vaguely specified as terms in Service Level Agreement or Goal Model. An typical example could be "*the performance should be high and the energy consumption should be reasonable*".

(3) When the DM is more interested in some specific part(s) of the Pareto front (while willing to look at the whole front), quality indicators which can deliver that specification need to be selected. For example, if the DM is more interested in around the knee of the Pareto front, the HV indicator could be a good choice. This is particularly true for the cloud autoscaling problem [2], in which different cloud tenants (users) may impose conflicting objectives due to the interference and shared infrastructure. Here, from the prospective of the cloud vendor, ensuring fairness among tenants of the same class is often the top priority and thus the knee solutions are more of interest. If the DM is interested in the boundary solutions, HV having the reference point fairly distant from the solution sets' boundary (e.g., 2 times of it) can be an option. This indicator is likely to be useful for the service composition problem [14] where one may prefer extreme solutions around the edges, e.g., those with low latency but high cost, or vice versa.

(4) When the DM's preferences are completely unavailable, using combined quality indicators, which can reflect all the general quality aspects (convergence, spread, uniformity, and cardinality), is always a good practice; e.g., HV and IGD [4]. If condition permits, we recommend to use more than one combined indicator as they deliver different preferences (thus having a high probability of fitting the DM), such as HV in favour of the knee region and IGD in favour of the uniformity. If one is more interested in separate assessment of solution sets' quality, s/he can use several indicators to respectively work on different quality aspects, such as GD [13] for convergence, DCI [8] for diversity, and PFS [5] for cardinality.

Finally, note that the considered quality indicators are desired to be (weakly) compatible with the $\lhd$ relation; that is, for two sets A and B, if $A \lhd B$ then A will be always evaluated better (not worse) than B by the indicators. Although this property cannot guarantee that the better-evaluation-result set between two sets is certainly preferred by the DM, it can rule out the misleading situation that the worse-evaluation-result set is always preferred. Among the indicators mentioned above, IPF and HV are (weak) compatible, while GD, IGD, PFS and DCI not. To make them compatible (or weak compatible at least), GD and IGD can be replaced by GD^+ [7] and IGD^+ [7], and PFS and DCI can be modified by solely considering the nondominated solutions w.r.t. other sets in their calculations.

5 CONCLUSION

Quality assessment of solution sets is an important issue in multiobjective optimisation, but stay relatively new to the SBSE researchers and practitioners. Taking Wang et al's paper as starting point, this paper has presented the importance of understanding the goal of quality assessment, and accordingly provided a pragmatic guide of quality indicator selection based upon the availability of the DM's preferences, serving as a new direction for the SBSE community.

ACKNOWLEDGMENT

This work is supported by the DAASE Programme Grant from the EPSRC (Grant No. EP/J017515/1).

REFERENCES

[1] B. Bozkurt, J. W. Fowler, E. S. Gel, B. Kim, M. Köksalan, and J. Wallenius. Quantitative comparison of approximate solution sets for multicriteria optimization problems with weighted Tchebycheff preference function. *Operations Research*, 58(3):650–659, 2010.

[2] T. Chen and R. Bahsoon. Self-adaptive trade-off decision making for autoscaling cloud-based services. *IEEE Transactions on Services Computing*, 10(4):618–632, 2017.

[3] J. L. Cochrane and M. Zeleny. *Multiple Criteria Decision Making*. University of South Carolina Press, 1973.

[4] C. A. C. Coello and M. R. Sierra. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *Proceedings of the Mexican International Conference on Artificial Intelligence (MICAI)*, pages 688–697, 2004.

[5] C. Henard, M. Papdakis, M. Harman, and Y. L. Traon. Combining multi-objective search and constraint solving for configuring large software product lines. In *Proceedings of the 2015 International Conference on Software Engineering*, pages 517–528, 2015.

[6] R. M. Hierons, M. Li, X. Liu, S. Segura, and W. Zheng. SIP: optimal product selection from feature models using many-objective evolutionary optimization. *ACM Transactions on Software Engineering and Methodology*, 25(2):17, 2016.

[7] H. Ishibuchi, H. Masuda, Y. Tanigaki, and Y. Nojima. Modified distance calculation in generational distance and inverted generational distance. In *Evolutionary Multi-Criterion Optimization (EMO)*, pages 110–125. 2015.

[8] M. Li, S. Yang, and X. Liu. Diversity comparison of Pareto front approximations in many-objective optimization. *IEEE Transactions on Cybernetics*, 44(12):2568–2584, 2014.

[9] M. Monperrus. A critical review of automatic patch generation learned from human-written patches: essay on the problem statement and the evaluation of automatic software repair. In *Proceedings of the 36th International Conference on Software Engineering*, pages 234–242, 2014.

[10] T. Saber, D. Brevet, G. Botterweck, and A. Ventresque. Is seeding a good strategy in multi-objective feature selection when feature models evolve? *Information and Software Technology*, 2017.

[11] T. Saber, J. Thorburn, L. Murphy, and A. Ventresque. VM reassignment in hybrid clouds for large decentralised companies: A multi-objective challenge. *Future Generation Computer Systems*, 2017.

[12] S. Sayın. Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming. *Mathematical Programming*, 87(3):543–560, 2000.

[13] D. A. Van Veldhuizen and G. B. Lamont. Evolutionary computation and convergence to a Pareto front. In *Late Breaking Papers at the Genetic Programming Conference*, pages 221–228, 1998.

[14] H. Wada, J. Suzuki, Y. Yamano, and K. Oba. E^3: A multiobjective optimization framework for sla-aware service composition. *IEEE Transactions on Services Computing*, 5(3):358–372, 2012.

[15] S. Wang, S. Ali, T. Yue, Y. Li, and M. Liaaen. A practical guide to select quality indicators for assessing Pareto-based search algorithms in search-based software engineering. In *Proceedings of the 38th International Conference on Software Engineering (ICSE)*, pages 631–642, 2016.

[16] W. Zheng, R. M. Hierons, M. Li, X. Liu, and V. Vinciotti. Multi-objective optimisation for regression testing. *Information Sciences*, 334:1–16, 2016.

[17] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang. Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. In *IEEE Congress on Evolutionary Computation*, pages 892–899, 2006.

[18] E. Zitzler, D. Brockhoff, and L. Thiele. The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. In *Evolutionary multi-criterion optimization*, pages 862–876, 2007.

[19] E. Zitzler and L. Thiele. Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, 1999.

[20] 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*, 7(2):117–132, 2003.