Is the Revisited Hypervolume an Appropriate Quality Indicator to Evaluate Multi-Objective Test Case Selection Algorithms?

Aitor Arrieta
Mondragon University
Mondragon, Spain
aarrieta@mondragon.edu

ABSTRACT

Multi-objective test case selection techniques are widely investigated with the goal of devising novel solutions to increase the cost-effectiveness of verification processes. When evaluating such approaches the entire Pareto-frontier of the algorithm needs to be considered. To do so, several quality indicators exist. The *hypervolume* (HV) is one of the most well-known and applied quality indicator. However, in the context of test case selection, this metric has certain limitations. For instance, two different fitness function combinations are not comparable if this metric is used at the search algorithm's objective function level. Consequently, researchers proposed the revisited HV (*rHV*) indicator. To compute the *rHV*, each solution of the search algorithm is individually assessed through two external utility functions: the cost and the fault detection capability (FDC). However, this increases the risk of having dominated solutions, which in practice may lead a decision maker (DM) to select such dominated solution. In this paper we assess whether the *rHV* is an appropriate quality indicator to assess multi-objective test case selection algorithms. To do so, we empirically assess whether the results between the *rHV* and the FDC of the different DM instances hold. Long story short, the *rHV* is an appropriate quality indicator.

CCS CONCEPTS

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

KEYWORDS

Test case selection, Decision Makers, Multi-objective search

ACM Reference Format:
Aitor Arrieta. 2022. Is the Revisited Hypervolume an Appropriate Quality Indicator to Evaluate Multi-Objective Test Case Selection Algorithms?. In *Genetic and Evolutionary Computation Conference (GECCO '22), July 9–13, 2022, Boston, MA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3512290.3528717

GECCO '22, July 9–13, 2022, Boston, MA, USA

ACM ISBN 978-1-4503-9237-2/22/07…$15.00
https://doi.org/10.1145/3512290.3528717

1 INTRODUCTION

Software development is expensive, to a large extent, due to verification costs. Testing is one of the predominant software verification techniques in industry, and its effort is usually large. Because of this, the research community has been actively devising novel techniques that improve the cost-effectiveness of software testing methods. Search algorithms have shown to be effective at helping on the automation of several software testing tasks, including test generation [3, 4, 19, 20, 22], test oracle improvement [50] and regression test optimization [8, 10, 15, 27, 39, 42, 45, 47, 53].

Test case selection has been one of the most active areas in software testing research, as it has shown to be an effective technique to improve the cost-effectiveness of software verification stages in industry [21, 45]. The test case selection problem is multi-objective in nature. On the one hand, adding a new test case will not decrease the fault detection capability of an existing test suite, but it will increase its cost [13]. On the other hand, removing a test case will not increase the execution cost of a test suite, but it can decrease its fault detection capability [13]. For this reason, Yoo and Harman proposed the first Pareto-based test case selection approach [53]. After this pioneering work, many researchers advocated for the use of multi-objective search algorithms for selecting test cases in different application areas, including SPLs [51], cyber-physical systems [7, 8, 30] and automotive systems [27].

Unlike single-objective search algorithms, multi-objective search algorithms return a set of non-dominated solutions. To evaluate such approaches, guidelines argue that the entire Pareto-frontier needs to be considered [1, 2, 28, 52]. To this end, and specific to the multi-objective test case selection problem, Panichella et al. revisited the well-known Hypervolume (HV) metric [42]. Their revisited HV (rHV) metric considers two external utility functions: (i) the cost and (ii) the percentage of faults revealed by each solution in the Pareto-frontier. Therefore, their proposal was not to use the HV in the objective space of the search algorithm, but to measure the fault revealing capability and actual cost of each solution in the Pareto-frontier returned by the search algorithm. With this, a second Pareto-frontier is obtained, because, there might be solutions that may be dominated by others in these external utility functions. This metric has been later used by other approaches using Pareto-based search algorithms [6–8]. Other approaches have followed a similar strategy for assessing their multi-objective test case selection algorithms, but instead using the *rHV*, they use a revisited version of the Area Under Curve (AUC) [41], which is similar in this case and faces the same problems as the *rHV*.

Nevertheless, to apply multi-objective test case selection algorithms in practice, decision makers (DMs) need to be considered. A DM takes the Pareto-frontier solutions as input and chooses its

favorite solution [28]. These DMs need to select their solution on the basis of certain criteria. Unfortunately, the percentage of faults to be detected by the solutions returned by the search algorithm is unknown to the DM. In contrast the *rHV* metric assumes that the DM will always select one solution from the Pareto-frontier. In fact, based on the studies by Arrieta et al., [6–8], after computing the external utility functions, many solutions returned by the search algorithms are dominated and not used to compute the *rHV*. For instance, for the NSGA-II algorithm in the four simulation models used in this study, which replicates [8], on average, only 11.88% of the solutions were non-dominated.

As a result, the DM has a high probability of selecting a solution that is dominated by others. Therefore, this paper aims at studying whether the *rHV* is an appropriate quality indicator to assess multi-objective test case selection algorithms. To this end, firstly, we propose two types of DMs that may be used in industrial settings. Secondly, we replicate a previous study [7, 8] where the *rHV* is used as quality indicator to assess the cost-effectiveness of different multi-objective test case selection techniques. In this replication study we analyze whether (1) the same conclusion holds when using the *rHV* quality indicator and the mutation scores (i.e., artificially seeded faults) under different instances of the DM, and (2) whether there is correlation between these two metrics.

The main contribution of this paper is an evaluation to prove the validity of the *rHV* when selecting test cases using multi-objective search algorithms. The results of our empirical evaluation suggests that the *rHV* quality indicator is appropriate to evaluate multi-objective test case selection algorithm. However, it poses some disadvantages and some risks that need to be highlighted when evaluating multi-objective test selection techniques. Specifically, this paper makes the following contributions:

- We analyze the problems that the *rHV* can bring when evaluating multi-objective test case selection algorithms.
- We propose two types of DMs for multi-objective test case selection algorithms.
- We compare the results of the mutation scores of the solutions returned by the different DMs with the results provided by the *rHV* quality indicator. To this end, we replicated the study Arrieta et al., [8] proposed in the context of test case selection of cyber-physical simulation models.

2 PROBLEM STATEMENT AND MOTIVATION

The Hypervolume (HV) quality indicator is the most widely used metric in the evolutionary multi-objective community [48]. This technique provides different advantages [48], such as, (1) being Pareto compliant, (2) being able to evaluate convergence and the diversity of a solution set simultaneously and (3) only requiring one reference point. Figure 1 illustrates the HV of three solutions for a two-objective maximization problem. The higher the HV, the higher the performance of the Pareto-frontier returned by the search algorithm.

Usually, for most evolutionary multi-objective optimization problems, the *HV* is applied at the algorithm's objective level. However, this poses several disadvantages in the context of multi-objective test case selection. Specifically, the ultimate goal of a test case selection algorithm is to reduce as much as possible the test execution

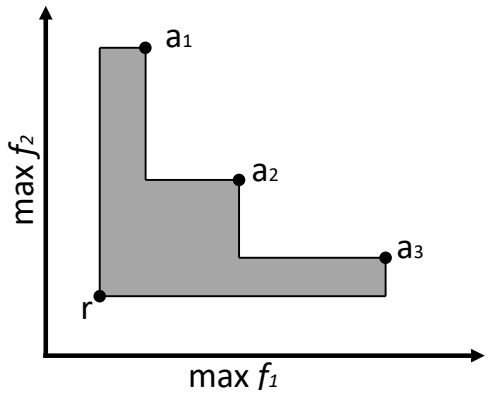

Figure 1: Illustration of the Hypervolume (HV). a_1, a_2 and a_3 are three solutions and r is the reference point

cost while maintaining the fault detection capability of the original test suite. The test selection algorithm guides the search through a set of fitness functions that measure certain adequacy of the selected sub-set of test cases (e.g., code coverage). Such fitness functions are hopefully correlated with the fault detection capability of the selected solutions. However, a higher adequacy criteria does not always lead to a higher fault detection capability [24]. In addition, the application of the *HV* at the algorithm's objective function space makes it not possible to compare the cost-effectiveness of different fitness function combinations.

To solve this issue, Panichella et al., proposed the revisited hypervolume (*rHV*) quality indicator, which permits measuring the fault detection capability of multi-objective test case selection approaches [42]. Instead of applying the *HV* metric at the search algorithm's objective function space, for each of the solutions provided by the search algorithm in the Pareto-frontier, two external utility functions are applied: (1) test execution cost and (2) fault detection capability. With these two external utility functions, a second Pareto-frontier is obtained, which is used to measure the *rHV*. This plausible approach permits (1) measuring whether the selected subset of test cases is actually able to reduce the test execution cost while guaranteeing its fault detection capability, and (2) comparing techniques that have different fitness functions and therefore, the traditional HV is not applicable. However, this approach assumes the availability of "ideal" decision makers (DMs) that are able to discard dominated solutions from the second Pareto-frontier.

```
1 int max (int n, int m){
2   if(n>m)          //mutant 1: if(n<m)
3     return n;      //mutant 2: return 0;
4   else
5     return m;
6 }
```

Listing 1: Code snippet of a function for returning the highest value from two integer inputs

Consider the example code snippet in Listing 1, which implements a function for returning the highest value of two integer numbers. For the sake of motivating the problem, consider also two mutants used for evaluating the test case selection problem. The first mutant changes the relational operator > by < in Line 2,

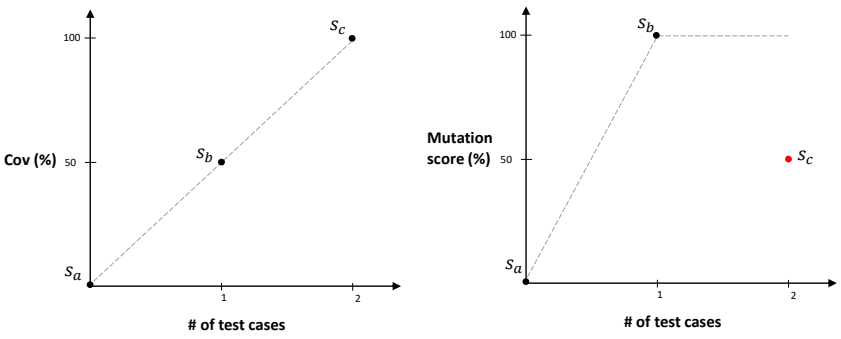

Figure 2: Example of a case where a solution is dominated by the rest of the solutions from the originally returned Pareto frontier by the search algorithm when applying two external utility functions for test case selection

whereas the second one changes the return value in Line 3, returning the value 0 instead of the value in n. A mutant is considered *killed* if the returned value of the function is different between the original non-mutated version, and the mutated version of the code.

Consider an initial test suite (TS) composed of four different test cases aiming to test the code snippet in Listing 1: $TS = \{tc_1, tc_2, tc_3, tc_4\}$. Let the implementation of each test case in TS be the following:

- $tc_1 = \{n = 0, m = -1\}$
- $tc_2 = \{n = 1, m = 3\}$
- $tc_3 = \{n = 5, m = 5\}$
- $tc_4 = \{n = 7, m = 2\}$

Suppose a multi-objective test case selection approach with two competing objective functions. The first one, reducing the number of test cases in the original test suite. The second objective function increasing the test coverage. With the aforementioned test suite, the search algorithm could return the following set of non-dominated solutions:

- $s_a = \{\emptyset\}$
- $s_b = \{tc_4\}$
- $s_c = \{tc_1, tc_2\}$

As can be seen in Figure 2, in terms of test coverage, all these three test cases are non-dominated solutions among themselves. However, when applied the external utility functions of (i) number of test cases (minimize) and (ii) mutation score (maximize), it can be appreciated that the solution s_c is dominated by the solution s_b, as solution s_b includes test case tc_4, which detects both seeded mutants. Conversely, while solution s_c includes two test cases with a stronger test coverage, non of these test cases is able to kill the second mutant. Given such a case, a DM that aims to select a solution with the highest test coverage as possible, it would select solution s_c.

When considering the replication package by Arrieta et al. [8], we noticed that this issue happened in most of the algorithm runs. Not only a few of the solutions were dominated when applying the external utility functions of test execution time (minimize) and mutation score (maximize), but a large amount of them (as previously mentioned, only 11.88% were non-dominated solutions). This problem, in practice, may lead the DM to select a dominated solution. Therefore, this issue rises the following question that this paper aims to answer:

Is the revisited hypervolume (rHV) an appropriate quality indicator to evaluate the cost-effectiveness of multi-objective test case selection algorithms?

To this end, firstly, we proposed two types of DMs aiming to select solutions returned by the test selection algorithm. It is important to recall that the DMs select one solution of the Pareto-frontier returned by the search algorithm. These DMs, similar to what happens in practice, do not have information related to the number of faults detected by the selected test cases. Secondly, we replicate our previous study [7, 8], for which the rHV quality indicator was used. This replication is a way to study whether the rHV is an appropriate quality indicator. To this end, we compared whether whether the results of the DMs and the rHV differ or not, and whether there are correlations by means of three research questions (RQs).

3 PROPOSED DECISION MAKERS

In practice, when using a Pareto-based search algorithm, a *decision maker* (DM) must select one of the solutions returned by the search algorithm. In this section, we propose two types of DMs that could be applicable to the test case selection context.

3.1 Time-Budget Decision Maker

A potential way of selecting a solution in the context of Pareto-based search algorithms could be to give the DM a targetted time budget. In such a case, the DM looks for the solution with highest quality that does not exceed the proposed time budget. These types of DMs are of interest for industry. For instance, in a domain analysis carried out in an industrial setting with partners from the maritime domain, they showed interest in executing as many test cases as possible within a given time budget [44]. From a test engineers' perspective, there may also be interest in wanting/needing to have some test results in a given time-frame (e.g., when the test engineer has a meeting, when they go to have lunch). Therefore, we implemented such a DM, which takes as input (1) a set of non-dominated solutions returned by the search algorithm and (2) a given time-budget provided by the user. Since in our experiments we only considered two fitness functions (i.e., test execution time and an effectiveness function), the DM returns the solution which is closer to the time-budget without exceeding it. If such solution does not exist, it returns a null (i.e., it is not possible to execute a test suite given that time budget).

In the remainder of this paper, we refer to these decision makers as "DM_tb%", *tb* being the time budget given to the decision maker. For instance, DM_10% refers to a DM whose goal is to provide the solution with the highest effectiveness metric that does not exceed the 10% of the cost of the entire test suite.

3.2 Maximum Effectiveness Objective Decision Maker

Another option could be to select a solution which obtains the maximum score of one of the effectiveness objective functions. This could be the case of, for instance, test coverage. The test engineer may want to ensure the highest degree of test coverage, but remove those test cases that do not provide additional coverage. Another scenario could be when using fitness functions that measure the similarity among the selected test cases. In such a case, a test engineer may want to remove redundant test cases. Therefore, this DM can be applied in a context where time is not critical, but some optimization is required and/or desired.

In the remainder of this paper, we refer to this DM as "DM_MAX".

4 EMPIRICAL EVALUATION

In our empirical evaluation, we aimed at answering the following three research questions (RQs):

- ***RQ1 – Sanity check:*** *Do the results between the rHV and the mutation scores provided by the different DM instances hold when comparing the NSGA-II algorithm with a baseline algorithm?* To assess whether the problem to solve is non-trivial, researchers often compare the multi-objective algorithm against a baseline technique. Usually, that baseline technique is Random Search (RS) [7, 8, 44, 51]. Therefore, the first RQ has as a goal to see whether the results hold when using the *rHV* quality indicator and the mutation scores provided by the DM instances when comparing a sophisticated multi-objective algorithm (i.e., NSGA-II) with RS.
- ***RQ2 – Fitness function comparison:*** *Do the results between the rHV and the mutation scores provided by the different DM instances hold when comparing different fitness function combinations?* Other typical comparisons when assessing multi-objective test case selection techniques is to assess the cost-effectiveness of different fitness functions when integrating them with the multi-objective search algorithm. This RQ has as a goal to assess whether the results between the *rHV* and the mutation scores provided by the different DM instances hold when comparing different fitness function combinations.
- ***RQ3 – Correlation:*** *How does the rHV correlate with respect to the mutation scores provided by the different DM instances?* The third RQ has as a goal to study whether a higher *rHV* has a higher chances of getting a higher fault detection capability by a solution chosen by the DM.

4.1 Experimental Setup

We now explain the experimental setup we carried out to answer the defined RQs.

4.1.1 Replicated study. To answer our RQs, we replicated the work by Arrieta et al., [7, 8], which targets the test case selection problem in the context of simulation-based testing of cyber-physical systems. This study was selected due to a variety of reasons. Firstly, the availability of a replication package that includes different simulation models, test cases and mutants. Secondly, in this study the authors used the *rHV* quality indicator. Lastly, this study has been used for replication purposes of other studies both by the same authors [6] as well as other authors [30].

4.1.2 Simulink Models. We used a total of four Simulink models. These four Simulink models were the same as the experimental evaluation by Arrieta et al., [8], which have also been used in other similar studies [6, 7, 34, 37, 38, 40]. Similar to Ling and Menzies [30], we did not use two of the original models [8] (i.e., Tiny and CC) due to their complexity being too low in terms of number of blocks. The key characteristics of the selected Simulink models are summarized in Table 1.

Table 1: Key characteristics of the selected Simulink models in the first application context

Case Study	# of Blocks	# of Inputs	# of Outputs	# of Test Cases	Initial set of mutants	Final set of mutants
CW	235	15	4	133	250	98
EMB	315	1	1	150	40	10
AC Engine	257	4	1	120	20	12
Two Tanks	498	11	7	150	34	6

4.1.3 Fitness function selection and combination. We selected five different fitness functions proposed by Arrieta et al., [8]. One of them relates to the cost (i.e., Test Execution Time (TET)) and the other four to effectiveness (i.e., instability, discontinuity, growth to infinity and min-max). We did not select similarity metrics because (1) these metrics will always result similar normalized values (0.95-0.99) with different test case selections [30] and (2) their performance was low [8]. We used the combination of the TET with the four effectiveness functions because the previous study showed that using two objective functions was better than combining more than two objective functions [8]. Thus, a total of four fitness function combinations were formed (Table 2). The details of each of these objectives functions is available in [8].

Table 2: Selected fitness function combinations based on the metrics proposed by Arrieta et al., [8]

	Effectiveness metric	Cost metric
c1	Discontinuity	Test execution time
c2	Growth to infinity	
c3	Instability	
c4	MinMax	

4.1.4 Time budgets of the decision makers. Based on some preliminary experimental runs, we selected eight different time budgets for the first proposed DM. Specifically, the DM is configured to select solutions incorporating a test suite that does not exceed the 1%, 5%, 10%, 15%, 20%, 30%, 40% and 50% of the original test suite's

test execution time. DM_Max is not configurable. Thus, in total we used nine different DM instances.

4.1.5 Evaluation metrics. Mutation testing was employed to assess the effectiveness of the solutions provided by the DM. This technique has been found to be a good substitute of real faults [25]. We used the same subset of mutants as the original evaluation study [8], which employed the mutation operators for Simulink models proposed by Binh et al., [12]. Information related to the initial and final set of mutants is available in Table 1. Notice that Arrieta et al., [8] removed (1) duplicated mutants (i.e., mutants equivalent to one another but not to the original program) as recommended by Papadakis et al., [43], (2) mutants that were killed by all test cases (as they considered them to be too weak mutants) and (3) mutants that were not killed by any test case (to avoid the inclusion of equivalent mutants). Thus, for each of the solutions returned by the different DM instances, we measured the mutation score (MS). The MS is normalized between the values 0-1, where 0 means that there were no mutants detected and 1 means that all mutants were detected. In addition to the mutation score, we obtained the revisited HV (*rHV*) for the solutions returned by the search algorithm [8].

4.1.6 Statistical tests. Since the employed algorithms are non-deterministic, we run each algorithm 50 times, as recommended by Arcuri and Briand [5], and as in the study we are replicating [8]. After executing the experiments, we assessed the normality distribution of the study through the Shapiro-Wilk test. For RQ1 and RQ2, since most data was not normally distributed (p-value in Shapiro-Wilk test < 0.05), we used the Mann-Whitney U-test to assess the significance of the results produced by the different algorithms. In addition, we used the Vargha and Delaney $\hat{A}_{12}$ value to assess the difference between the different algorithms. For RQ3, the Spearman's rank correlation ρ was applied, which measures the correlation of the mutation scores (MS) with respect to the *rHV*. The test returns a ρ value within the range [-1,1] and a p-value. A positive ρ value means that the correlation is positive, i.e., the MS is larger for larger *rHV* values, whereas a negative value means the opposite. A p-value lower than 0.05 means that there is statistical significance in the correlation.

4.1.7 Algorithm setup. We used the NSGA-II as search algorithm, and its configuration was the same as prior studies [7, 8, 53]. The population was set to 100 and the number of fitness evaluations was 25,000. The crossover rate was set to 0.8 and the mutation probability was done with 1/N probability rate, N being number of test cases. We used the the binary tournament selection operator [7, 8]. To answer the first RQ, RS was used. The total number of fitness evaluations for this algorithm was also set to 25,000 [7, 8].

4.2 Analysis of the Results and Discussion

RQ1 – Comparison between NSGA-II and RS: Table 3 shows the statistical tests carried out to answer the first RQ. The table shows, for both the *rHV* and mutation scores of the solutions returned by the DM (1) the Vargha and Delaney $\hat{A}_{12}$ value and (2) a p-value indicating the statistical significance. The $\hat{A}_{12}$ indicates the probability of NSGA-II being better than RS. As can be seen, when comparing NSGA-II with RS using the *rHV* quality indicator, the $\hat{A}_{12}$ was 1 in all cases, with strong statistical significance. When comparing the mutation scores provided by both the RS and the NSGA-II algorithm, we could see that at least in three of the DMs the $\hat{A}_{12}$ was in favour of NSGA-II with statistical significance.

However, in 1 out of the 16 of the combinations, the mutation scores were in favor of RS. This was in the CW simulation model for the c2 fitness combination. In that case, DM30, DM40 and DM_Max showed better mutation score results in the case of RS than in the case of NSGA-II, with statistical significance. For such a case, the fitness function combination employs the growth to infinity score, which was not useful in such simulation model due to the maximum outputs of the system being limited to a maximum number [8].

Thus, the first RQ can be answered as follows:

> ***In general, when the rHV is in favor of the NSGA-II, the mutation scores of the solutions provided by the DM are in favor too. However, we saw that in some cases there might be exceptions.***

RQ2 – Comparison of different fitness functions: Table 4 shows the statistical tests carried out to answer the second RQ, which aims to answer whether the results obtained by the *rHV* hold when considering the mutation scores obtained by the solutions selected by the different DMs. When considering the *rHV* metric, in 16 out of 24 combinations, there was statistical significance. When analyzing those 16 cases, we could see that there was also statistical significance in at least one of the DMs when considering the mutation score. For instance, for combinations c1 and c3 of the ACEng model, the p-value for the *rHV* was 0.01. In such a case, there was also statistical significance in the mutation scores provided by the DMs DM15, DM20, DM30 and DM40. Moreover, when considering the $\hat{A}_{12}$ values, the results held similar for the mutation scores and the *rHV* metric (i.e., if the $\hat{A}_{12}$ values are negative for one specific fitness function combination for the *rHV*, the mutation score for the DMs will also be negative, and vice versa). In other words, when the *rHV* suggested a better performance of one specific fitness function combination, the mutation scores provided by the DM instances also suggested the same.

In 8 out of the 24 combinations, there was no statistical significance between the compared fitness function combinations. In most of those situations, there was also no statistical significance when considering the mutation scores. The only exception was in the EMB model for the c1 vs c4 combination, where the p-value for the *rHV* was 0.951, but there was statistical significance in the mutation scores provided by DM15. This was, however, only a minor exception, and when considering the $\hat{A}_{12}$ value of that case ($\hat{A}_{12}$ = 0.611), it could be seen that the significance difference level was only *medium* according to a related categorization [46]. Thus, we can answer the second RQ as follows:

> ***The rHV quality indicator holds with the mutation scores provided by the different DM instances, minor exceptions aside.***

RQ3 – Correlation between MS and rHV: Table 5 reports the results of the Spearman's rank correlation between the *rHV* and the MS obtained by the different solutions provided by the DMs. Notice that we applied such test for each MS of each DM instance and each simulation model. Furthermore, we assessed the data both with and without the results provided by RS. We took this decision because both the *rHV* and MS values of RS were very low, and therefore,

Table 3: RQ1 - Results of the $\hat{A}_{12}$ and Mann-Whitney U-test p-values when comparing the NSGA-II vs RS algorithm. $\hat{A}_{12}$>0.5 means that the NSGA-II is better than RS. p-value < 0.05 means that there is statistical significance

		rHV		MS DM_1%		MS DM_5%		MS DM_10%		MS DM_15%		MS DM_20%		MS DM_30%		MS DM_40%		MS DM_50%		MS DM_max	
		$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val
ACEng	**c1**	1.000	<0.001	0.560	0.012	0.940	<0.001	1.000	<0.001	1.000	<0.001	0.935	<0.001	0.529	0.571	0.540	0.254	0.520	0.159	0.500	-
	c2	1.000	<0.001	0.510	0.327	0.930	<0.001	1.000	<0.001	1.000	<0.001	0.941	<0.001	0.542	0.413	0.472	0.404	0.500	1.000	0.500	-
	c3	1.000	<0.001	0.520	0.159	0.990	<0.001	1.000	<0.001	1.000	<0.001	0.993	<0.001	0.654	0.002	0.530	0.082	0.520	0.159	0.500	-
	c4	1.000	<0.001	0.520	0.159	0.930	<0.001	1.000	<0.001	1.000	<0.001	0.976	<0.001	0.576	0.146	0.532	0.426	0.490	0.327	0.500	-
CW	**c1**	1.000	<0.001	0.500	-	0.610	<0.001	0.900	<0.001	1.000	<0.001	1.000	<0.001	0.958	<0.001	0.658	0.004	0.586	0.110	0.768	<0.001
	c2	1.000	<0.001	0.500	-	0.580	0.003	0.900	<0.001	1.000	<0.001	1.000	<0.001	0.218	<0.001	0.309	0.001	0.567	0.215	0.373	<0.001
	c3	1.000	<0.001	0.510	0.327	0.720	<0.001	0.990	<0.001	1.000	<0.001	1.000	<0.001	0.962	<0.001	0.744	<0.001	0.697	<0.001	0.852	<0.001
	c4	1.000	<0.001	0.500	-	0.570	0.007	0.920	<0.001	1.000	<0.001	1.000	<0.001	0.550	0.388	0.502	0.980	0.713	<0.001	0.833	<0.001
EMB	**c1**	1.000	<0.001	0.500	-	0.500	-	0.720	<0.001	0.990	<0.001	1.000	<0.001	0.967	<0.001	0.690	<0.001	0.670	<0.001	0.580	0.003
	c2	1.000	<0.001	0.500	-	0.500	-	0.770	<0.001	0.980	<0.001	1.000	<0.001	0.967	<0.001	0.750	<0.001	0.640	<0.001	0.590	0.002
	c3	1.000	<0.001	0.500	-	0.530	0.082	0.870	<0.001	1.000	<0.001	1.000	<0.001	0.986	<0.001	0.720	<0.001	0.610	0.001	0.610	<0.001
	c4	1.000	<0.001	0.500	-	0.520	0.159	0.730	<0.001	0.990	<0.001	1.000	<0.001	0.986	<0.001	0.750	<0.001	0.660	<0.001	0.560	0.012
TwoTanks	**c1**	1.000	<0.001	0.500	-	0.560	0.012	0.880	<0.001	1.000	<0.001	1.000	<0.001	0.990	<0.001	0.500	-	0.500	-	0.500	-
	c2	1.000	<0.001	0.500	-	0.660	<0.001	0.990	<0.001	1.000	<0.001	1.000	<0.001	1.000	<0.001	0.500	-	0.500	-	0.500	-
	c3	1.000	<0.001	0.510	0.327	0.790	<0.001	0.960	<0.001	0.990	<0.001	1.000	<0.001	0.970	<0.001	0.500	-	0.500	-	0.500	-
	c4	1.000	<0.001	0.500	-	0.640	<0.001	1.000	<0.001	1.000	<0.001	1.000	<0.001	1.000	<0.001	0.500	-	0.500	-	0.500	-

Table 4: RQ2 - Results of the $\hat{A}_{12}$ and Mann-Whitney U-test p-values for the different algorithms configurations both for the *rHV* and the mutation score provided by the decision makers

		rHV		MS DM_1%		MS DM_5%		MS DM_10%		MS DM_15%		MS DM_20%		MS DM_30%		MS DM_40%		MS DM_50%		MS DM_max	
		$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val
ACEng	**c1 vs c2**	0.546	0.430	0.550	0.051	0.595	0.094	0.497	0.960	0.543	0.435	0.472	0.583	0.500	1.000	0.530	0.378	0.510	0.327	0.500	-
	c1 vs c3	0.313	0.001	0.540	0.141	0.482	0.753	0.433	0.233	0.295	<0.001	0.320	0.001	0.389	0.018	0.450	0.023	0.500	1.000	0.500	-
	c1 vs c4	0.564	0.270	0.540	0.143	0.607	0.060	0.567	0.236	0.524	0.664	0.554	0.295	0.510	0.845	0.530	0.378	0.510	0.327	0.500	-
	c2 vs c3	0.245	<0.001	0.490	0.568	0.371	0.023	0.433	0.232	0.283	<0.001	0.349	0.005	0.389	0.018	0.420	0.003	0.490	0.327	0.500	-
	c2 vs c4	0.526	0.657	0.490	0.568	0.516	0.779	0.570	0.210	0.484	0.773	0.581	0.121	0.510	0.845	0.500	1.000	0.500	-	0.500	-
	c3 vs c4	0.745	<0.001	0.500	1.000	0.640	0.013	0.637	0.015	0.718	<0.001	0.715	<0.001	0.621	0.011	0.580	0.003	0.510	0.327	0.500	-
CW	**c1 vs c2**	0.808	<0.001	0.500	-	0.540	0.319	0.786	<0.001	0.953	<0.001	0.986	<0.001	0.968	<0.001	0.839	<0.001	0.602	0.054	0.540	0.354
	c1 vs c3	0.224	<0.001	0.490	0.327	0.382	0.015	0.190	<0.001	0.163	<0.001	0.214	<0.001	0.302	<0.001	0.400	0.060	0.415	0.100	0.410	0.004
	c1 vs c4	0.813	<0.001	0.500	-	0.545	0.245	0.770	<0.001	0.995	<0.001	0.994	<0.001	0.992	<0.001	0.890	<0.001	0.619	0.025	0.522	0.601
	c2 vs c3	0.042	<0.001	0.490	0.327	0.333	<0.001	0.029	<0.001	0.008	<0.001	<0.001	<0.001	0.012	<0.001	0.103	<0.001	0.320	0.001	0.370	<0.001
	c2 vs c4	0.480	0.738	0.500	-	0.506	0.868	0.517	0.774	0.575	0.198	0.582	0.159	0.579	0.169	0.572	0.202	0.505	0.926	0.483	0.700
	c3 vs c4	0.968	<0.001	0.510	0.327	0.669	<0.001	0.962	<0.001	1.000	<0.001	1.000	<0.001	0.998	<0.001	0.934	<0.001	0.701	<0.001	0.610	0.001
EMB	**c1 vs c2**	0.475	0.667	0.500	-	0.500	-	0.454	0.389	0.578	0.120	0.502	0.959	0.480	0.407	0.500	-	0.500	-	0.500	-
	c1 vs c3	0.332	0.004	0.500	-	0.470	0.082	0.331	0.002	0.591	0.074	0.510	0.822	0.530	0.343	0.510	0.327	0.510	0.327	0.500	-
	c1 vs c4	0.504	0.951	0.500	-	0.480	0.159	0.487	0.805	0.611	0.030	0.572	0.121	0.490	0.702	0.500	-	0.500	-	0.500	-
	c2 vs c3	0.348	0.009	0.500	-	0.470	0.082	0.372	0.022	0.507	0.903	0.507	0.869	0.550	0.083	0.510	0.327	0.510	0.327	0.500	-
	c2 vs c4	0.529	0.620	0.500	-	0.480	0.159	0.531	0.563	0.524	0.657	0.569	0.143	0.510	0.655	0.500	-	0.500		0.500	-
	c3 vs c4	0.670	0.004	0.500	-	0.510	0.668	0.652	0.006	0.514	0.799	0.563	0.184	0.460	0.187	0.490	0.327	0.490	0.327	0.500	-
TwoTanks	**c1 vs c2**	0.235	<0.001	0.500	-	0.402	0.020	0.350	0.001	0.480	0.315	0.500	-	0.500	-	0.500	-	0.500	-	0.500	-
	c1 vs c3	0.183	<0.001	0.490	0.327	0.268	<0.001	0.346	<0.001	0.481	0.330	0.500	-	0.500	-	0.500	-	0.500	-	0.500	-
	c1 vs c4	0.209	<0.001	0.500	-	0.419	0.046	0.315	<0.001	0.470	0.082	0.500	-	0.500	-	0.500	-	0.500	-	0.500	-
	c2 vs c3	0.366	0.021	0.490	0.327	0.355	0.005	0.484	0.601	0.500	1.000	0.500	-	0.500	-	0.500	-	0.500	-	0.500	-
	c2 vs c4	0.492	0.888	0.500	-	0.513	0.781	0.460	0.141	0.490	0.327	0.500	-	0.500	-	0.500	-	0.500	-	0.500	-
	c3 vs c4	0.646	0.012	0.510	0.327	0.651	0.003	0.478	0.370	0.490	0.327	0.500	-	0.500	-	0.500	-	0.500	-	0.500	-

Table 5: RQ3 - Results of the Spearman rank correlation between the *rHV* and the mutation score provided by the different DMs. A positive ρ means that there is positive correlation. and a p-value less than 0.05 that the correlation is statistically significant

		MS DM_1%		MS DM_5%		MS DM_10%		MS DM_15%		MS DM_20%		MS DM_30%		MS DM_40%		MS DM_50%		MS_max	
		ρ	p-val	ρ	p-val	ρ	p-val	ρ	p-val	ρ	p-val	ρ	p-val	ρ	p-val	ρ	p-val	ρ	p-val
NSGA-II Data	ACEng	0.056	0.429	0.567	<0.001	0.352	<0.001	0.398	<0.001	0.300	<0.001	0.114	0.109	0.189	0.007	0.015	0.835	-	-
	CW	0.122	0.085	0.623	<0.001	0.791	<0.001	0.664	<0.001	0.672	<0.001	0.624	<0.001	0.482	<0.001	0.293	<0.001	0.240	0.001
	EMB	-	-	0.269	<0.001	0.846	<0.001	0.288	<0.001	0.073	0.307	0.008	0.907	-0.008	0.911	-0.056	0.432	-	-
	TwoTanks	0.122	0.085	0.799	<0.001	0.477	<0.001	0.089	0.211	-	-	-	-	-	-	-	-	-	-
NSGA-II + RS Data	ACEng	0.165	0.001	0.851	<0.001	0.851	<0.001	0.858	<0.001	0.839	<0.001	0.165	0.001	0.081	0.105	0.101	0.043	-	-
	CW	0.086	0.084	0.522	<0.001	0.867	<0.001	0.891	<0.001	0.892	<0.001	0.651	<0.001	0.223	<0.001	0.283	<0.001	0.472	<0.001
	EMB	-	-	0.192	<0.001	0.765	<0.001	0.844	<0.001	0.833	<0.001	0.858	<0.001	0.465	<0.001	0.318	<0.001	0.276	<0.001
	TwoTanks	0.086	0.084	0.636	<0.001	0.861	<0.001	0.863	<0.001	0.866	<0.001	0.868	<0.001	-	-	-	-	-	-

obtaining a positive correlation would be easier. Additionally, we aimed to assess whether the *rHV* is useful both, when comparing different algorithms (e.g., RS and NSGA-II) and when comparing different fitness function combinations when using one specific multi-objective search algorithm.

From Table 5, it can be observed that for 45 out of 72 cases, the correlation was positive (i.e., $\rho > 0$) with statistical significance (i.e., p-value < 0.05). On the one hand, when not considering the results data from RS, in 17 out of 36 cases there was positive correlation with statistical significance. On the other hand, when considering the data from both NSGA-II and RS, there was a positive correlation with statistical significance in 28 out of the 36 cases. The correlation between the *rHV* and MS had strong correlation with statistical significance in all the MSs provided by the solutions of the DMs configured to provide test suites not exceeding the time budget of 5 and 10 % of the entire test suite. In 15 out of the 72 cases, the correlation could not be measured because all the values of the MS were 1 or 0 (i.e., 0 for DMs configured to provide solutions not exceeding 1% of the time, and 1 for the rest). The rest of the solutions did not have statistical significance. In such cases, in all cases except two of them, the ρ value was positive. In two of the cases (i.e., DM40 and DM50 for EMB, NSGA-II only), the ρ value was negative. However, these values were very low (-0.008 and -0.056) and the p-values were higher than 0.05 (i.e., there was no statistical significance).

Figure 3 shows how the mutation scores are distributed along with their corresponding *rHV* for the CW model when using the DM_10%. In this distribution, where $\rho = 0.867$, it can be appreciated

the positive correlation of the mutation score with the *rHV*. It is also noticeable the difference between RS and NSGA-II results. On the other hand, Figure 4 shows the distribution of the *rHV* for the EMB model when using the DM_40%, where $\rho = -0.008$ for only NSGA-II values. In this scatter plot, it can be appreciated that one of the solutions for the NSGA-II did not obtain the maximum mutation score. When having a closer look, we found that this solution was a dominated solution from the second Pareto-frontier. This is clearly an example of the threats that implies the usage of the *rHV* for assessing multi-objective test case selection algorithms. In some cases, some solutions can be dominated when using the external utility function. Such solutions are not considered for computing the *rHV*. However, the DM, which in practice does not have information related to the fault detection capability of the solutions, can select one of those dominated solutions.

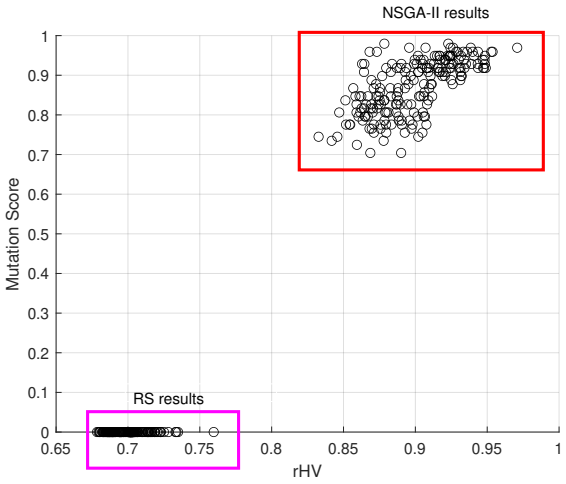

Figure 3: Scatter plot of the *rHV* and mutation scores of the results obtained for DM_10% in the CW model

These results indicate that there is positive correlation between the *rHV* and the MS values. This means that the higher the *rHV* for a set of solutions, the chances for selecting a solution with higher fault detection capability by a DM are higher. Therefore, we can answer the third RQ as follows:

> ***The rHV metric shows a positive correlation with respect to the mutation score in at least two DM instances in each simulation model with statistical significance. Only in two instances showed a small negative correlation, but without statistical significance.***

Summary of results and practical implications

In this study, we found that, in general, the same conclusion holds either when using the *rHV* or the different mutation scores of the solutions selected by the DM instances. Specifically, we showed that, in general, (1) the results were consistent between the *rHV* and the mutation scores provided by the different DM instances when comparing two different algorithms (i.e., NSGA-II and the baseline RS); (2) the results were consistent between the *rHV* and

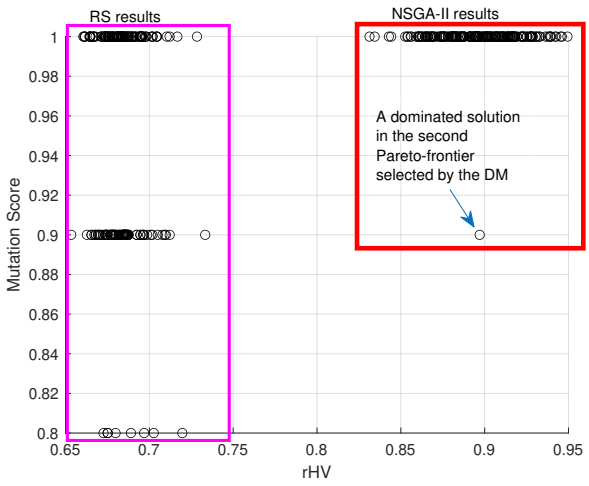

Figure 4: Scatter plot of the *rHV* and mutation scores of the results obtained for DM_40% in the EMB model

the mutation scores provided by the different DM instances when comparing two different fitness function combinations; and (3) there was positive correlation between the *rHV* and the different mutation scores provided by most DM instances.

Our results have the following implications for researchers that need to evaluate multi-objective test case selection approaches:

- We demonstrated that the *rHV* is an appropriate quality indicator to assess the cost-effectiveness of multi-objective test case selection algorithms.
- However, our results also indicate that there might be exceptions. Thus, it is important to highlight in the threats to validity section that the *rHV* assumes the availability of an "ideal" DM, and therefore, there is some risk that a real DM may select a dominated solution (e.g., Figure 4).
- Besides, it is noteworthy that the *rHV* has also some drawbacks. Specifically, with the *rHV* metric, it is difficult to explain which practical implications of using one algorithm or another are, which is paramount for transferring software engineering methods to industry. For instance, when using the *rHV* it is not possible make statements like, e.g., *"Technique X can detect n% more faults in m% less time than Technique Y"*. Therefore, we recommend researchers complementing the *rHV* metric with other metrics when assessing the cost-effectiveness of multi-objective test case selection algorithms. Specifically, we recommend using different instances of DMs, such as those proposed in Section 3.

4.3 Threats to validity

We now summarize the threats to validity of our study. An *internal validity* threat in our study could be related to the employed mutants. It is noteworthy that the number of mutants used is not large, but it is comparable to other similar studies targeting simulation-based testing of cyber-physical systems (e.g., [9, 12, 31–33, 35–37]). To mitigate such threat, we used the same mutants as the replicated study [7, 8], which also removed duplicated and trivial mutants to

reduce this threat, as suggested by Papadakis et al., [43]. Another *internal validity* threat in our evaluation involves the parameters used in the algorithms, which were not changed. We used the same parameters as previous studies [7, 8, 53] to mitigate this threat.

An *external validity* threat in our study relates to the generalization of results. To mitigate such threat we used four different simulation models of different sizes and complexity. All these models had more than 235 blocks, which are larger than most public subject models according to a previous study [14]. Furthermore, one of such studies relates to an industrial simulation model developed by engineers from Bosch [49]. In addition, for each of these studied models, we used four different fitness function combinations.

As in any other study involving randomized algorithms, a *conclusion validity* threat involves the non-determinism of the employed algorithms. We mitigated this threat by running each algorithm 50 times to account for random variations, as recommended in related guidelines [5]. Moreover, we used the appropriate statistical tests to carefully analyze the results and differences among the different techniques.

Construct validity threats arise when the measures used are not comparable across algorithms. To mitigate this threat we used the same stopping criterion for all the algorithms (i.e., the total number of fitness evaluations was set to 25,000).

5 RELATED WORK

Multi-objective test case selection has been an active research area in the last few years [6–8, 27, 41, 42, 44, 45, 51, 53]. Most approaches focus on proposing cost-effective objective functions either for general-purpose software systems [53] or specific to a domain (e.g., product lines [51], simulation models [8]). Some studies have focused on improving the search process by, for instance, injecting diversity in genetic algorithms[42], using seeding techniques to initialize the population [6] or proposing a novel crossover function [41]. Unlike all these papers, which focus on devising novel search-based test selection approaches, this paper focuses on evaluating whether the *rHV* is an appropriate quality indicator for assessing such approaches.

Wang et al., [52] were the first in proposing practical guidelines to appropriately select quality indicators to assess search-based software engineering problems based on the results of an study using three industrial and real-world case studies. Ali et al., [1] empirically evaluated eight quality indicators and six multi-objective search algorithms within nine search-based software engineering problems. They assessed quality indicator agreements and their relation with search-based software engineering problems and multi-objective search algorithms. In both studies, test case selection and minimization algorithms were included. However, their assessment of quality indicators was at the algorithms' objective space. Conversely, the *rHV* uses the external utility functions of cost and fault detection capability. Li et al., [28] proposed a guide to evaluate solutions in Pareto-based search-based software engineering problems based on a theoretical analysis. However, neither the test case selection/minimization problem nor the potential issues of the *rHV* quality indicator were contemplated.

The HV quality indicator faces several open issues. Thus, it has attracted the attention of evolutionary computation researchers [29]. Some papers have studied how to solve the computational complexity this indicator faces (e.g., [11, 17]), which exponentially increases with the number of objectives. Other studies focus on the problem of incremental update [16, 23, 26], which aim at measuring how much the HV varies when a solution is added or removed from the Pareto-frontier. Another relevant area studied by the evolutionary computation community is related to the subset selection [18, 29], which aim at finding the optimal K-size subset with respect to HV from an N-size solution set, with the objective of maximizing the probability that the DM will find at least one acceptable solution in the subset. Unlike all these studies, our paper analyzes whether the *rHV* is an appropriate quality indicator for test case selection. For the *rHV*, the computational complexity is not large, as it only considers two external objective functions (i.e., cost and fault detection capability). The problem, instead, relates to the possibility of having dominated solutions from the original Pareto-frontier, which may be selected by the DM.

6 CONCLUSION AND FUTURE WORK

In this paper, we analyze whether the revisited HV (*rHV*) quality indicator is an appropriate metric to evaluate multi-objective test case selection algorithms. One of the core problems of such technique lies in the assumption of the existence of "ideal" decision makers (DMs). However, in practice, DMs do not have information of the fault detection capability of each of the solutions returned by the search algorithm. In this paper we propose two types of DMs for test selection and replicate a previous study [7, 8]. In such replication, we assess whether the conclusions hold when using the *rHV* metric and the mutation scores of different instances of the DMs. The results of our empirical evaluation suggest that the *rHV* is an appropriate quality indicator to assess multi-objective test case selection algorithms. However, in some cases, such technique may have certain limitations that need to be acknowledged in the threats to validity sections. Furthermore, we recommend to complement such technique with different instances of DMs to reduce such threats and report the practical implications of the results.

In the future, we would like to analyze other quality indicators used in other studies. With this data, we plan to feed the software tool developed by Ali et al., [1] intended to assess and recommend quality indicators in search-based software engineering, contributing this way to the community to use appropriate quality indicators by using more accurate observations. Furthermore, we would like to extend our empirical evaluation by including other application domains (e.g., multi-objective test generation).

Replication package: The replication package of the study is available in Zenodo: https://doi.org/10.5281/zenodo.6389707

ACKNOWLEDGMENTS

This publication is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 871319.

REFERENCES

[1] Shaukat Ali, Paolo Arcaini, Dipesh Pradhan, Safdar Aqeel Safdar, and Tao Yue. 2020. Quality indicators in search-based software engineering: an empirical evaluation. *ACM Transactions on Software Engineering and Methodology (TOSEM)* 29, 2 (2020), 1–29.

[2] Shaukat Ali, Paolo Arcaini, and Tao Yue. 2020. Do Quality Indicators Prefer Particular Multi-objective Search Algorithms in Search-Based Software Engineering?. In *International Symposium on Search Based Software Engineering*. Springer, 25–41.

[3] Hussein Almulla and Gregory Gay. 2020. Generating Diverse Test Suites for Gson Through Adaptive Fitness Function Selection. In *International Symposium on Search Based Software Engineering*. Springer, 246–252.

[4] Hussein Almulla and Gregory Gay. 2022. Learning how to search: Generating effective test cases through adaptive fitness function selection. *Empirical Software Engineering* 27, 2 (2022), 1–62.

[5] Andrea Arcuri and Lionel Briand. 2011. A practical guide for using statistical tests to assess randomized algorithms in software engineering. In *2011 33rd International Conference on Software Engineering (ICSE)*. IEEE, 1–10.

[6] Aitor Arrieta, Joseba Andoni Agirre, and Goiuria Sagardui. 2020. Seeding strategies for multi-objective test case selection: an application on simulation-based testing. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference*. 1222–1231.

[7] Aitor Arrieta, Shuai Wang, Ainhoa Arruabarrena, Urtzi Markiegi, Goiuria Sagardui, and Leire Etxeberria. 2018. Multi-objective black-box test case selection for cost-effectively testing simulation models. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 1411–1418.

[8] Aitor Arrieta, Shuai Wang, Urtzi Markiegi, Ainhoa Arruabarrena, Leire Etxeberria, and Goiuria Sagardui. 2019. Pareto efficient multi-objective black-box test case selection for simulation-based testing. *Information and Software Technology* 114 (2019), 137–154.

[9] Aitor Arrieta, Shuai Wang, Goiuria Sagardui, and Leire Etxeberria. 2019. Search-Based test case prioritization for simulation-Based testing of cyber-Physical system product lines. *Journal of Systems and Software* 149 (2019), 1–34.

[10] Wesley Klewerton Guez Assunção, Thelma Elita Colanzi, Silvia Regina Vergilio, and Aurora Pozo. 2014. A multi-objective optimization approach for the integration and test order problem. *Information Sciences* 267 (2014), 119–139.

[11] Nicola Beume, Carlos M Fonseca, Manuel Lopez-Ibanez, Luis Paquete, and Jan Vahrenhold. 2009. On the complexity of computing the hypervolume indicator. *IEEE Transactions on Evolutionary Computation* 13, 5 (2009), 1075–1082.

[12] Nguyen Thanh Binh, Khuat Thanh Tung, et al. 2016. A novel fitness function of metaheuristic algorithms for test data generation for simulink models based on mutation analysis. *Journal of Systems and Software* 120 (2016), 17–30.

[13] Yiqun T Chen, Rahul Gopinath, Anita Tadakamalla, Michael D Ernst, Reid Holmes, Gordon Fraser, Paul Ammann, and René Just. 2020. Revisiting the relationship between fault detection, test adequacy criteria, and test set size. In *Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering*. 237–249.

[14] Shafiul Azam Chowdhury, Soumik Mohian, Sidharth Mehra, Siddhant Gawsane, Taylor T Johnson, and Christoph Csallner. 2018. Automatically finding bugs in a commercial cyber-physical system development tool chain with SLforge. In *Proceedings of the 40th International Conference on Software Engineering*. 981–992.

[15] Andrea De Lucia, Massimiliano Di Penta, Rocco Oliveto, and Annibale Panichella. 2012. On the role of diversity measures for multi-objective test case selection. In *2012 7th International Workshop on Automation of Software Test (AST)*. IEEE, 145–151.

[16] Michael Emmerich, Nicola Beume, and Boris Naujoks. 2005. An EMO algorithm using the hypervolume measure as selection criterion. In *International Conference on Evolutionary Multi-Criterion Optimization*. Springer, 62–76.

[17] Michael Emmerich and André Deutz. 2014. Time complexity and zeros of the hypervolume indicator gradient field. In *EVOLVE-a bridge between probability, set oriented numerics, and evolutionary computation III*. Springer, 169–193.

[18] Michael TM Emmerich, André H Deutz, and Iryna Yevseyeva. 2015. A Bayesian approach to portfolio selection in multicriteria group decision making. *Procedia Computer Science* 64 (2015), 993–1000.

[19] Gordon Fraser and Andrea Arcuri. 2012. Whole test suite generation. *IEEE Transactions on Software Engineering* 39, 2 (2012), 276–291.

[20] Gordon Fraser, Andrea Arcuri, and Phil McMinn. 2015. A memetic algorithm for whole test suite generation. *Journal of Systems and Software* 103 (2015), 311–327.

[21] Vahid Garousi, Ramazan Özkan, and Aysu Betin-Can. 2018. Multi-objective regression test selection in practice: An empirical study in the defense software industry. *Information and Software Technology* 103 (2018), 40–54.

[22] Gregory Gay. 2017. Generating effective test suites by combining coverage criteria. In *International Symposium on Search Based Software Engineering*. Springer, 65–82.

[23] Andreia P Guerreiro and Carlos M Fonseca. 2017. Computing and updating hypervolume contributions in up to four dimensions. *IEEE Transactions on Evolutionary Computation* 22, 3 (2017), 449–463.

[24] Laura Inozemtseva and Reid Holmes. 2014. Coverage is not strongly correlated with test suite effectiveness. In *Proceedings of the 36th international conference on software engineering*. 435–445.

[25] René Just, Darioush Jalali, Laura Inozemtseva, Michael D Ernst, Reid Holmes, and Gordon Fraser. 2014. Are mutants a valid substitute for real faults in software testing?. In *Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering*. 654–665.

[26] Joshua D Knowles, David W Corne, and Mark Fleischer. 2003. Bounded archiving using the Lebesgue measure. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, Vol. 4. IEEE, 2490–2497.

[27] Remo Lachmann, Michael Felderer, Manuel Nieke, Sandro Schulze, Christoph Seidl, and Ina Schaefer. 2017. Multi-objective black-box test case selection for system testing. In *Proceedings of the Genetic and Evolutionary Computation Conference*. 1311–1318.

[28] Miqing Li, Tao Chen, and Xin Yao. 2020. How to Evaluate Solutions in Pareto-based Search-Based Software Engineering? A Critical Review and Methodological Guidance. *IEEE Transactions on Software Engineering* 01 (2020), 1–1.

[29] Miqing Li and Xin Yao. 2019. Quality evaluation of solution sets in multiobjective optimisation: A survey. *ACM Computing Surveys (CSUR)* 52, 2 (2019), 1–38.

[30] Xiao Ling and Tim Menzies. 2021. Faster Multi-Goal Simulation-Based Testing Using DoLesS (Domination with Least Square Approximation). *arXiv preprint arXiv:2112.01598* (2021).

[31] Bing Liu, Lucia, Shiva Nejati, Lionel C Briand, and Thomas Bruckmann. 2016. Simulink fault localization: an iterative statistical debugging approach. *Software Testing, Verification and Reliability* 26, 6 (2016), 431–459.

[32] Bing Liu, Shiva Nejati, Lionel C Briand, et al. 2017. Improving fault localization for Simulink models using search-based testing and prediction models. In *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 359–370.

[33] Bing Liu, Shiva Nejati, Lionel C Briand, et al. 2019. Effective fault localization of automotive Simulink models: achieving the trade-off between test oracle effort and fault localization accuracy. *Empirical Software Engineering* 24, 1 (2019), 444–490.

[34] Reza Matinnejad, Shiva Nejati, and Lionel C Briand. 2017. Automated testing of hybrid Simulink/Stateflow controllers: industrial case studies. In *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*. 938–943.

[35] Reza Matinnejad, Shiva Nejati, Lionel C Briand, and Thomas Bruckmann. 2015. Effective test suites for mixed discrete-continuous stateflow controllers. In *Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering*. 84–95.

[36] Reza Matinnejad, Shiva Nejati, Lionel C Briand, and Thomas Bruckmann. 2016. Automated test suite generation for time-continuous simulink models. In *proceedings of the 38th International Conference on Software Engineering*. 595–606.

[37] Reza Matinnejad, Shiva Nejati, Lionel C Briand, and Thomas Bruckmann. 2018. Test generation and test prioritization for simulink models with dynamic behavior. *IEEE Transactions on Software Engineering* 45, 9 (2018), 919–944.

[38] Claudio Menghi, Shiva Nejati, Khouloud Gaaloul, and Lionel C Briand. 2019. Generating automated and online test oracles for simulink models with continuous and uncertain behaviors. In *Proceedings of the 2019 27th acm joint meeting on european software engineering conference and symposium on the foundations of software engineering*. 27–38.

[39] Debajyoti Mondal, Hadi Hemmati, and Stephane Durocher. 2015. Exploring test suite diversification and code coverage in multi-objective test case selection. In *2015 IEEE 8th International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 1–10.

[40] Shiva Nejati, Khouloud Gaaloul, Claudio Menghi, Lionel C Briand, Stephen Foster, and David Wolfe. 2019. Evaluating model testing and model checking for finding requirements violations in Simulink models. In *Proceedings of the 2019 27th acm joint meeting on european software engineering conference and symposium on the foundations of software engineering*. 1015–1025.

[41] Mitchell Olsthoorn and Annibale Panichella. 2021. Multi-objective test case selection through linkage learning-based crossover. In *International Symposium on Search Based Software Engineering*. Springer, 87–102.

[42] Annibale Panichella, Rocco Oliveto, Massimiliano Di Penta, and Andrea De Lucia. 2014. Improving multi-objective test case selection by injecting diversity in genetic algorithms. *IEEE Transactions on Software Engineering* 41, 4 (2014), 358–383.

[43] Mike Papadakis, Yue Jia, Mark Harman, and Yves Le Traon. 2015. Trivial compiler equivalence: A large scale empirical study of a simple, fast and effective equivalent mutant detection technique. In *2015 IEEE/ACM 37th IEEE International Conference on Software Engineering*, Vol. 1. IEEE, 936–946.

[44] Dipesh Pradhan, Shuai Wang, Shaukat Ali, and Tao Yue. 2016. Search-based cost-effective test case selection within a time budget: An empirical study. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. 1085–1092.

[45] Dipesh Pradhan, Shuai Wang, Shaukat Ali, Tao Yue, and Marius Liaaen. 2017. CBGA-ES: A cluster-based genetic algorithm with elitist selection for supporting multi-objective test optimization. In *2017 IEEE International Conference on Software Testing, Verification and Validation (ICST)*. IEEE, 367–378.

[46] Jeanine Romano, Jeffrey D Kromrey, Jesse Coraggio, Jeff Skowronek, and Linda Devine. 2006. Exploring methods for evaluating group differences on the NSSE and other surveys: Are the t-test and Cohen'sd indices the most appropriate choices. In *annual meeting of the Southern Association for Institutional Research*. Citeseer, 1–51.

[47] Takfarinas Saber, Florian Delavernhe, Mike Papadakis, Michael O'Neill, and Anthony Ventresque. 2018. A hybrid algorithm for multi-objective test case selection. In *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[48] Ke Shang, Hisao Ishibuchi, Linjun He, and Lie Meng Pang. 2020. A survey on the hypervolume indicator in evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation* 25, 1 (2020), 1–20.

[49] Thomas Strathmann and Jens Oehlerking. 2015. Verifying Properties of an Electro-Mechanical Braking System.. In *ARCH@ CPSWeek*. 49–56.

[50] Valerio Terragni, Gunel Jahangirova, Paolo Tonella, and Mauro Pezzè. 2020. Evolutionary improvement of assertion oracles. In *Proceedings of the 28th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*. 1178–1189.

[51] Shuai Wang, Shaukat Ali, and Arnaud Gotlieb. 2015. Cost-effective test suite minimization in product lines using search techniques. *Journal of Systems and Software* 103 (2015), 370–391.

[52] Shuai Wang, Shaukat Ali, Tao Yue, Yan Li, and Marius Liaaen. 2016. 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*. 631–642.

[53] Shin Yoo and Mark Harman. 2007. Pareto efficient multi-objective test case selection. In *Proceedings of the 2007 international symposium on Software testing and analysis*. 140–150.