A Novel Mutation Operator for Search-based Test Case Selection

Aitor Arrieta, Miren Illarramendi

Mondragon University, Mondragon, Spain
{aarrieta, millarramendi}@mondragon.edu

Abstract. Test case selection has been a widely investigated technique to increase the cost-effectiveness of software testing. Because the search space in this problem is huge, search-based approaches have been found effective, where an optimization algorithm (e.g., a genetic algorithm) applies mutation and crossover operators guided by corresponding objective functions with the goal of reducing the test execution cost while maintaining the overall test quality. The de-facto mutation operator is the bit-flip mutation, where a test case is mutated with a probability of $1/N$, N being the total number of test cases in the original test suite. This has a core disadvantage: an effective test case and an ineffective one have the same probability of being selected or removed. In this paper, we advocate for a novel mutation operator that promotes selecting effective test cases while removing the ineffective ones. To this end, instead of applying a probability of $1/N$ to every single test case in the original test suite, we calculate new selection and removal probabilities. This is carried out based on the adequacy criterion of each test case, determined before executing the algorithm (e.g., based on historical data). We integrate our approach in the domain of Cyber-Physical Systems (CPSs) within a widely applied dataset. Our results suggests that the proposed mutation operator can increase the effectiveness of search-based test case selection methods, especially when the time budget for executing test cases is low.

Keywords: Search-based test case selection, Regression test optimization

1 Introduction

Search algorithms have been found to be effective in solving multiple software engineering tasks, including test case generation [1,15,16,17,26], fault localization [23], and regression test optimization [8,10,12,14,19,29,31,32,35]. Because of the time it requires for test cases to execute, in the last few years, significant effort has been devoted to the field of test case selection [6,8,28,29,31,35]. Test case selection aims at reducing the number of test cases to execute, while maintaining as high as possible the overall test quality. As the search space of the test case selection problem is huge (i.e., 2^N, N being the total number of test cases), search algorithms have been found appropriate in multiple domains, including, software product lines [33], cyber-physical systems [21] and deep neural networks [4].

The test case selection problem has been investigated from different perspectives. On the one hand, a large corpus of studies investigate the effect of different fitness functions (and their combination) in the cost-effectiveness of the selected test cases [8,7]. On the other hand, other studies investigate how different search algorithms perform when selecting test cases [33]. A third group of studies focus on giving algorithmic solutions adapted to the context of test case selection. For instance, Panichella et al. [29] propose the inclusion of diversity in multi-objective genetic algorithms. This was carried out by using orthogonal design and orthogonal evolution mechanisms during the search process [29]. Arrieta et al. [5,6] propose a set of seeding strategies adapted to the test case selection problem. Specifically, instead of generating the initial population pure randomly, they propose different strategies, e.g., generating solutions in the population with a different amount of test cases selected or injecting diversity in the initial population [5,6]. Olsthoorn and Panichella [27] proposed a linkage learning approach to optimize test case selection by using unsupervised learning techniques at the crossover level to yield new individuals by inferring structures. Despite all the work in the field of search-based test case selection, to the best of our knowledge, there is no study that has investigated how to enhance the test case selection problem from the mutation operator perspective. Indeed, most of the studies employ the default value of bit-flip mutation with a probability of $1/N$, N being the number of test cases in the initial test suite [27,29,33,31,7,5,6].

In this paper we propose a novel mutation operator specifically designed for targeting the test case selection problem. Instead of giving the same probability for being selected or removed from the initial test suite to all test cases, we advocate for giving different probabilities for each test case based on their effectiveness. This way, the mutation operator has higher probabilities of selecting effective test cases and higher probabilities of removing ineffective ones.

Our evaluation was conducted by using a well-known test case selection dataset for Cyber-Physical Systems (CPSs), using four different case study systems and a total of 15 instances (i.e., four different instances for each case study system, except for one of them, where we used three different instances). Our approach showed superiority in 8 out of 15 instances with statistical significance. Conversely, in 5 out of 15 instances, the performance was worse than the traditionally employed mutation operator. Overall, we can summarize the key contributions of this paper as follows:

- Technique: We propose a novel mutation operator that fosters selection of effective test cases and reduction of weak test cases.
- Evaluation: We evaluate the approach in four CPS case study systems of a well-known test case selection dataset.
- Replication package: We provide a replication package with the code, experiments, results and analysis: `https://dx.doi.org/10.6084/m9.figshare.23998029`

The rest of the paper is structured as follows. Section 2 proposes the mutation operator. Section 3 explains the conducted evaluation to assess the approach. Section 4 positions our work with the current state-of-the-art. Lastly, we conclude and discuss future research avenues in Section 5.

2 The Mutation Operator

2.1 Formalization

Let $TS = \{tc_1, tc_2, ..., tc_N\}$ be a test suite composed of N test cases (tc). Because executing all test cases in TS is often not practical, the test case selection problem aims at selecting a subset of test cases from TS, such that $TS' = \{tc_{x1}, tc_{x2}, ..., tc_x M\}$ is a subset of TS (i.e., $TS' \subseteq TS$) and $M \leq N$. To guide the search algorithm, the test quality and cost of TS' needs to be measured. In the context of multi-objective test case selection, this is carried out by a set of p objective functions, i.e., $OF = \{of_1, of_2, ..., of_p\}$. Usually, the objective functions cover at least one quality objective function (e.g., code coverage [35], failure detection rate [31]) and one cost function (e.g., test execution time [35]).

The most typical way for representing a solution in multi-objective test case selection is through binary coding representation [27,29,33,31,7,5,6], where the i-th digit of the binary string represents whether a test case has been selected (when the digit is 1) or not selected (when the digit is 0). For instance, given a test suite of $N = 6$ test cases, an individual k can be represented as $s_k = (1, 0, 0, 1, 1, 0)$. For the individual s_k, test cases tc_1, tc_4 and tc_5 are selected whereas test cases tc_2, tc_3 and tc_6 are not.

2.2 Approach

When selecting a set of test cases from TS, it is necessary to measure the cost and overall effectiveness of the subset of selected test cases. To this end, each test case encompasses certain degree of adequacy, which can be measured based on different criteria. For instance, one typical criterion is *test coverage*, where, if tc_1 is able to exercise a larger portion of code than tc_2, it is said that tc_1 is more adequate than tc_2. Another typical criterion is the *fault detection capability* of test cases, which can be measured based on the number of failures the test cases have triggered during their historical executions.

In this paper we advocate for promoting the selection of adequate test cases and deselection of non-adequate test cases. To this end, we propose to consider the adequacy of a test case during the mutation process of the search-based test case selection algorithm. Our technique is simple, yet effective. For each test case, instead of having a mutation probability of $1/N$, we obtain two probabilities based on a predefined adequacy criterion: the selection probability (p_{sel_i}) and the removal probability (p_{rem_i}). The former provides the probability for the i-th test case for being selected when this is not selected in s_k. The latter provides the probability for the i-th test case for being removed when it is selected in s_k. An effective test case, (i.e., that with a high adequacy score) should have a higher p_{sel_i} and a lower p_{rem_i} than $1/N$. Conversely, a non-effective test case should have a lower p_{sel_i} and a higher p_{rem_i} than $1/N$.

Let as_{tc_i} be the normalized adequacy score of test case i, where the higher the value, the higher its adequacy. Given a test suite of N test cases, we obtain the selection probability for tc_i as follows:

$$p_{sel_i} = \frac{as_{tc_i}}{\sum_{i=1}^{N} as_{tc_i}} \tag{1}$$

On the other hand, the removal probability for tc_i can be obtained as follows:

$$p_{rem_i} = \frac{1 - as_{tc_i}}{\sum_{i=1}^{N}(1 - as_{tc_i})} \tag{2}$$

It is important to recall that p_{sel_i} is only applied when the tc_i is not selected in the individual (i.e., it has a 0 in the binary string representation of the solution). Conversely, p_{rem_i} is only applied when the tc_i is selected in the individual (i.e., it has a 1 in the binary string representation of the solution). That is, our mutation operator iterates the status of each test case in each solution provided by the search algorithm and applies the corresponding probability depending on the test case selection status (i.e., selected or not selected).

2.3 Applicability and Limitations

Our approach requires to retrieve the predefined probabilities for being selected/removed from the initial test suite during the mutation process. To address this, the adequacy of a test case needs to be obtained. As a result, our approach is applicable for regression testing contexts, where data from previous executions exists.

Conversely, our approach is not applicable for cases in which the similarity of test cases is the only metric to assess the adequacy of a test case. A potential way could be to extract the adequacy of each test case by measuring the distance to the closest test case. However, this would require to compute distance metrics of each test case every time a new test case is selected or removed during the mutation process as well as every time the mutation operator is invoked. Consequently, this approach would not scale in terms of computational cost, especially in those cases with large test suites.

3 Evaluation

3.1 Research Questions

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

- **RQ1 – Overall cost-effectiveness:** How does the proposed mutation operator perform when compared to the traditionally employed bit-flip mutation operator in terms of overall cost-effectiveness?

- **RQ2 – Fault detection capabilities:** Is the proposed mutation operator capable of detecting more faults than the traditional bit-flip mutation operator given a test execution time budget? When is it beneficial to use the proposed mutation operator?

With the first RQ, we aimed at assessing the overall cost-effectiveness of the new mutation operator compared to the traditional mutation operator. With the second RQ, we aimed at assessing the effectiveness of the new mutation operator in terms of fault detection when a decision maker selects one solution of the Pareto-frontier given a time-budget [3]; this may provide us some insights about when the proposed approach is beneficial.

3.2 Experimental Design

Dataset and case study systems: We employed the dataset and case study systems provided by Arrieta et al., [8] for test case selection. This dataset has been widely used by different researchers for test case selection studies [6,21,20,5,3]. Moreover, a recent study [3] confirmed with this dataset that the revisited hypervolume metric (i.e., one of the metrics used to in our evaluation setup) is an appropriate metric to assess multi-objective test case selection approaches. The dataset involves a total of 6 Simulink models (encompassing each a different CPS case study) of different characteristics and complexities (i.e., in terms of size, number of inputs and outputs). Similar to other studies [3,20,21], we did not use two of the Simulink models due to their simplicity. Table 1 summarizes the main characteristics of the selected Simulink models. Each Simulink model encompassed between 120 to 150 test cases. Moreover, the dataset employs a set of mutants, appropriately filtered out in order to remove duplicate and trivial mutants [8].

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

Simulink models	**# 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

Fitness functions and adequacy scores: Similar to recent test case selection studies using the dataset used in this paper, we derived a total of four different fitness function combinations. Each of the combinations encompassed one effectiveness metric and the test execution time (TET) as the cost metric. Related to the effectiveness metrics, we employed four black-box metrics defined by Arrieta et al. [8], which are commonly employed by other researchers too [20,21]. Similar to recent studies, we discarded similarity metrics because (1) their performance was low [8] and (2) distance-based metrics cannot be applied with our mutation operator as an adequacy criterion (see Section 2.3 for details).

For the adequacy criterion in the mutation operator, for each of the combinations used, we employed the same as the one used in the effectiveness fitness function. Although our approach permits the use of an adequacy criterion different to the one used to compute the fitness, we opted to use the same as this would be a natural choice by a developer (i.e., the fitness function employed in the algorithm should be the most adequate one). This would also permit to have a "syncrhonization" between the mutation operator and the effectiveness fitness function.

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

Effectiveness metric	**Cost metric**
c2 Growth to infinity	Test Execution Time
c2 Growth to infinity	Test Execution Time
c3 Instability	Test Execution Time
c4 MinMax	Test Execution Time

Evaluation metrics: To answer the first RQ, we employed the revisited Hypervolume (rHV). Proposed first by Panichella et al. [29], this metric determines the overall cost-effectiveness of the algorithm by deriving a second Pareto-frontier from the original Pareto-frontier returned by the multi-objective test case selection algorithm. This second Pareto-frontier is derived by considering the actual fault detection capability (e.g., obtained through mutation testing) and the cost. In a recent empirical study [3] (using the dataset from this paper), it was demonstrated that this metric is appropriate for multi-objective test case selection.

As for the second RQ, since the goal was to measure to which extent the proposed mutation operator showed benefits when detecting faults, we employed the mutation score. Multi-objective search algorithms return a set of solutions, i.e., the Pareto-frontier. Therefore, to obtain the mutation score, a decision maker (DM) needs to select one solution among all. To this end, we implemented a DM proposed in our previous study [3], 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). It is noteworthy that the DM does not have prior information of the detected faults. Specifically, we configured the DM 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, i.e., the same as Arrieta [3].

Algorithms setups: We used the NSGA-II algorithm as it is the most widely used algorithm for test case selection [27,29,33,31,7,5,6]. We use the parameters used in prior studies [35,31,7,5,6]. The population size was set to 100, the crossover probability was 0.8 and we used the binary tournament selection operator. Regarding the baseline mutation operator, we employed the bit-flip mutation operator with probability $1/N$, N being the number of test cases in the test suite.

Runs and statistical tests: As the employed Pareto-based search algorithm (i.e., NSGA-II) is stochastic, based on recommendations by Arcuri and Briand [2], we repeated its execution 50 times. In addition, we analyzed the results through statistical tests. We first measured the normality distribution of the obtained results through the Shapiro-Wilk test. As most of the data was not normally distributed (i.e., 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. Moreover, the Vargha and Delaney $\hat{A}_{12}$ value was employed to assess the difference between the different algorithms.

3.3 Analysis of the Results

RQ1 – Overall cost-effectiveness Table 3 provides the $\hat{A}_{12}$ and p-values when comparing the proposed mutation operator with the traditional one for the *rHV* metric. On the one hand, the $\hat{A}_{12}$ value provides the probability of a technique being better than the other one. The higher the $\hat{A}_{12}$, the higher probability that the traditional mutation operator (i.e., 1/N) was better than the proposed one in this study. On the other hand, the p-value indicates whether there was statistical significance. We consider so if the p-value was below 0.05.

Table 3: RQ1 – Summary of the statistical test results *rHV* metric for the different case study systems and different cost-effectiveness metrics

	c1		**c2**		**c3**		**c4**	
	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val	$\hat{A}_{12}$	p-val
ACEngine	**0.71**	**<0.001**	**0.68**	**0.002**	0.53	0.6466	**0.61**	**0.048**
CW	**0.06**	**<0.001**	**-**	**-**	**0.15**	**<0.001**	**0.99**	**<0.001**
EMB	**0.29**	**<0.001**	**0.98**	**<0.001**	**0.25**	**<0.001**	0.50	0.967
TwoTanks	**0.33**	**0.003**	**0.36**	**0.017**	**0.16**	**<0.001**	**0.30**	**<0.001**

The results indicate that in 8 out of 15 of the analyzed fitness function combinations, there was statistical significance in favor of our mutation operator. Specifically, in three out of four case study systems (i.e., CW, EMB and

TwoTanks), there was at least two fitness function combinations for which our mutation operator outperformed with statistical significance the commonly used mutation operator. Conversely, for 5 out of 15 of the studied scenarios, the proposed operator was not favorable. In fact, for the ACEngine case study system, our approach seemed to rather reduce the overall cost-effectiveness. A potential explanation for this could be that the test execution time of each test case also has an important impact in this specific case study, which was not considered in the mutation operator.

It is noteworthy that the results seemed more beneficial when employing the c1 and c3 fitness function combinations, which relate to the combinations that employ the discontinuity and instability anti-patterns to measure the effectiveness fitness function. Based on prior results employing these case study systems [8,6], these two fitness functions were found to be the most effective ones. We recall that to obtain the selection and removal probabilities of each test case, we employed the same as those used for the fitness function calculations. Therefore, if those attributes are not appropriate, it is expectable that the mutation operator does not favor the multi-objective search algorithm towards obtaining better results. Therefore, RQ1 can be answered as follows:

RQ1: In more than half of the studied cases, the proposed mutation operator showed positive results with statistical significance for the rHV metric. However, in some of the cases, the mutation operator showed a negative impact, which may be attributed to (1) ineffective adequacy criterion used to compute the probability array and (2) need for considering the test execution time too in the mutation operator.

RQ2 – Fault detection capability The graphs from Figure 1 show the average mutation scores obtained for the different configurations in which our novel mutation operator showed a positive influence according to the results from RQ1. Specifically, it can be appreciated that when employing our technique, the mutation scores were higher when the time budgets were low. Some eye-catching results in which the benefit of our proposed mutation operator were high involve (1) CW case study system for the c1 and c3 configurations; (2) EMB case study system for c1 and c3 configurations; and (3) TwoTanks case study system for c3 and c4 configurations. For instance, in the CW case study system, when the time budget for executing the test suite should not exceed a 5% of the original test suite's test execution time, the mutation score increased over 3-fold and 2-fold for c1 and c3, respectively.

Conversely, the graphs in Figure 2 show those cases in which our mutation operator had a negative effect according to the previous RQ. This is also confirmed when plotting the mutation scores based on the decision provided by the DMs. Specifically, the results for CW-c4 and EMB-c2 were significantly worse than those used by the traditional mutation operator. The core reason is that those fitness function configurations involved the anti-patterns related to the growth to infinity (GTI) and the difference between minimum and maximum

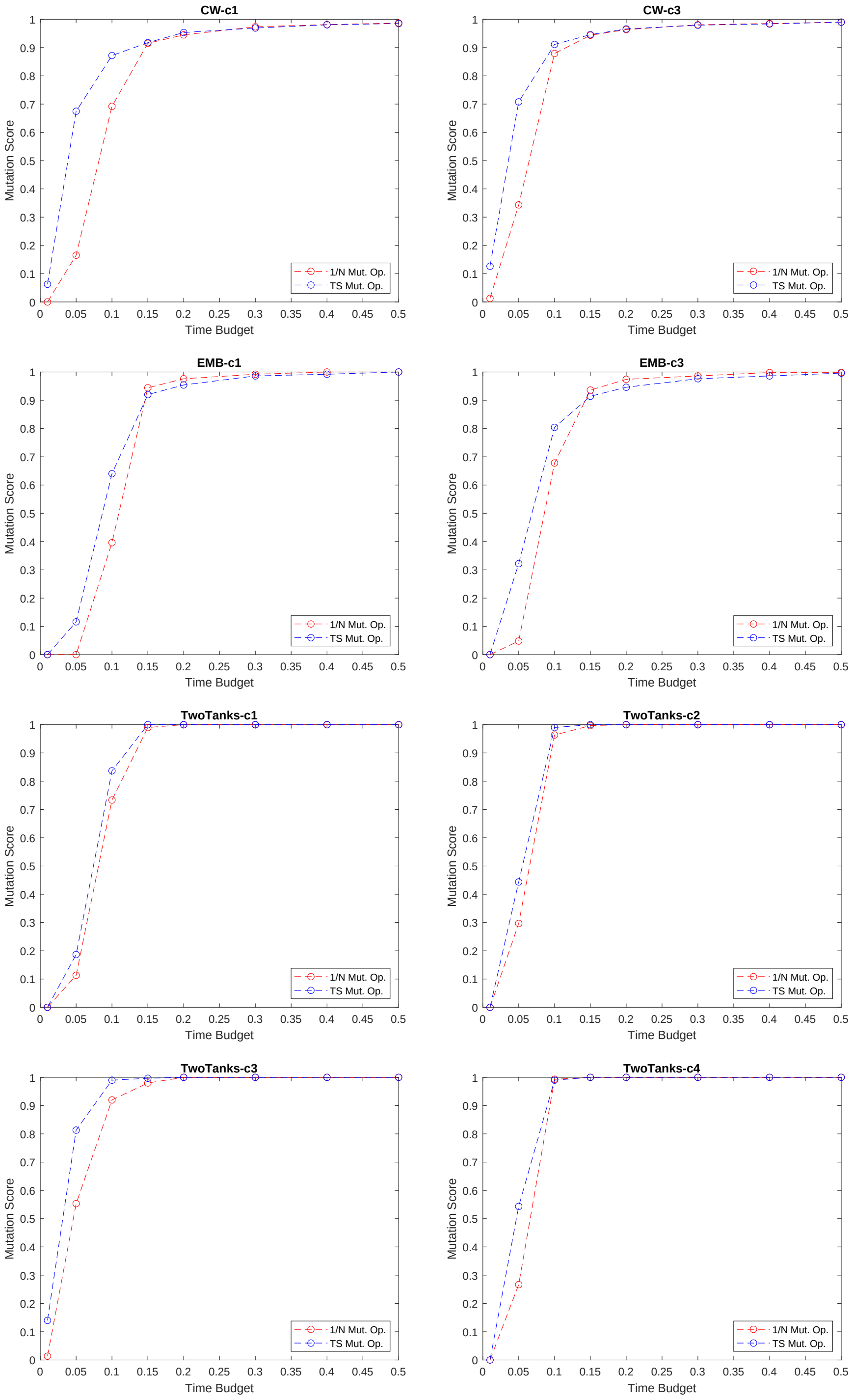

Fig. 1: Average mutation scores of the 50 runs for the cases where our algorithm showed a positive effect in the mutation score

values. When looking more into the detail of these results, we found that the test cases that had high MinMax and GTI values also involved long test cases in terms of execution time. As a result, in both cases, there were no solutions in the Pareto-frontier that involved test suites with test cases that encompassed lower time budgets than the 15% it took to execute the original test suite. Consequently, the mutation score values for three of the configurations for these cases were 0, as no test case could be executed. We conjecture that a potential way of solving this issue could be to include information of the test execution time when obtaining the selection and removal probabilities. This could be carried out, for instance, by employing a weighted approach.

Lastly, the graphs in Figure 3 show the mutation score values for those cases in which, for RQ1, there was no statistical significance. We show that the average value for the 50 runs was mostly the same in all cases. This could be attributed to the low effectiveness of the fitness functions used in those specific combinations.

Based on the obtained results, the second RQ can be answered as follows:

RQ2: The proposed mutation operator helped increase the mutation score in 8 out of the 15 studied cases. The approach shows benefits when the time budget to execute test cases is low, showing significant increase in the fault detection capabilities in those cases.

3.4 Threats to Validity

Internal validity: An internal validity threat of our study relates to the mutants. Specifically, the number of mutants was not large, but it is comparable to similar studies in which CPS models are employed [9,11,22,23,24,25]. This is mainly because such kind of systems take a long time to execute and therefore, it is not practical to use a large amount of mutants. However, to reduce this threat, we employed the same mutants as previous studies [6,7,8,21]. Moreover, such studies already removed duplicated and trivial mutants as suggested by Papadakis et al. [30]. Another internal validity threat refers to the employed algorithm, which was the NSGA-II. Further validation of our approach is required to see how the proposed mutation operator performs with other algorithms. Note, however, that the NSGA-II is the most widely used multi-objective algorithm in test case selection studies [7,8,34,35]. Lastly, the parameters of the algorithms might have an influence in the obtained results. We used the same parameters as previous studies [5,6,7,8,35] to mitigate this internal validity threat.

External validity: The main external validity threat of our evaluation relates to the generalizability of the results. We only employed four different case study systems and more case studies are required to further validate the approach, which we foresee to target it in the close future. However, the selected case study systems were diverse in terms of complexity and characteristics and have been widely used for multi-objective test case selection research.

Conclusion validity: A conclusion validity threat of our evaluation relates to the non-determinism of the search algorithms. This was mitigated by running

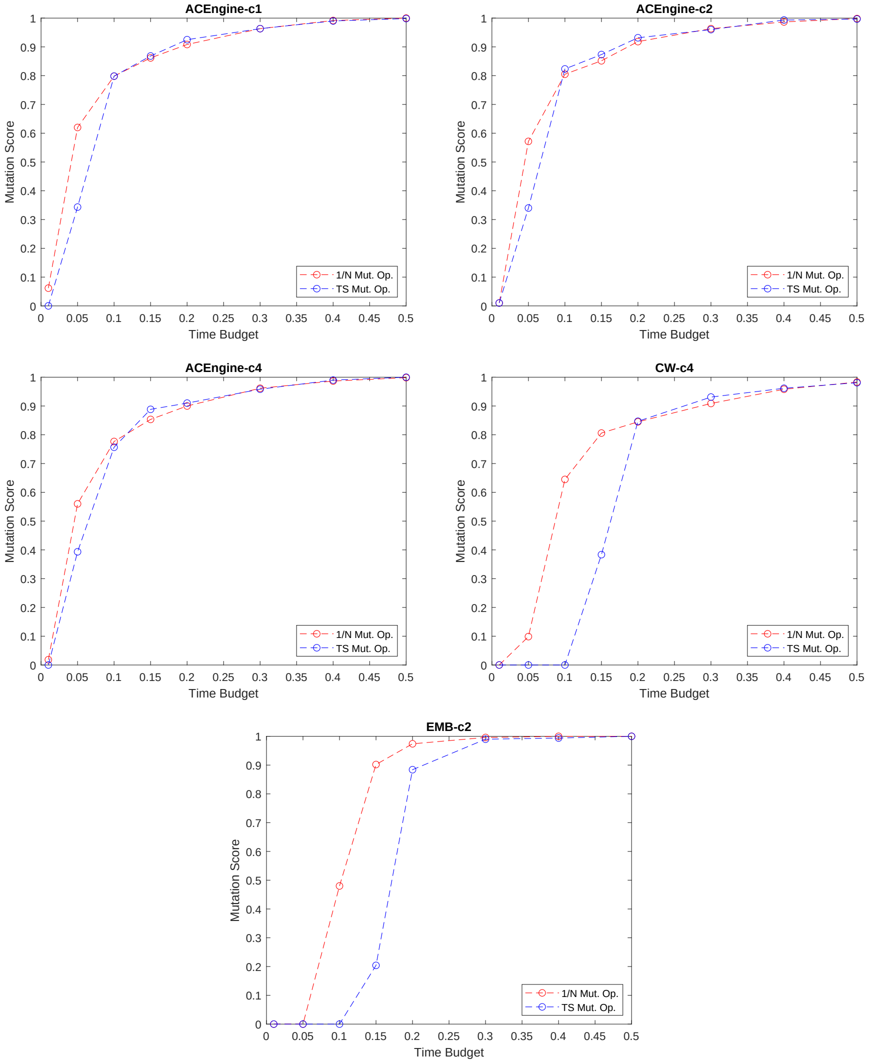

Fig. 2: Average mutation scores of the 50 runs for the cases where our mutation score showed a negative effect in the mutation score

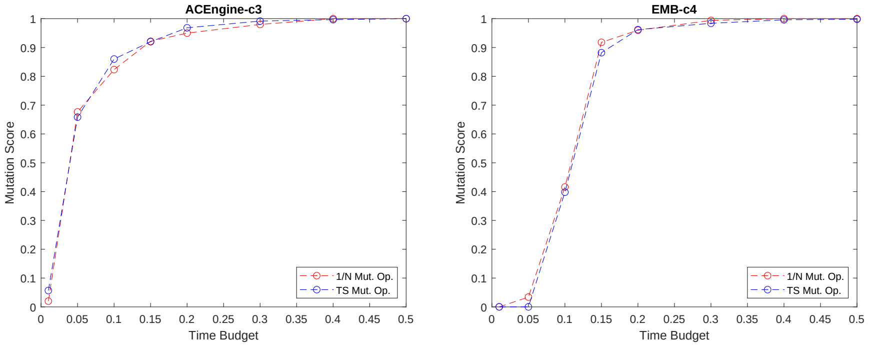

Fig. 3: Average mutation scores of the 50 runs for which the proposed mutation score did not have any effect

each algorithm instance 50 times and using statistical tests to analyze the results, as recommended by related guidelines [2].

Construct validity: To mitigate construct validity threats of our study, we employed the same setup for all algorithms in terms of population size and number of generation. This way, the NSGA-II configured to use our mutation operator or the commonly used one (i.e., probability of 1/N) employ the same number of fitness function computations.

4 Related Work

As previously explained, search-based test case selection studies can be divided in three main groups: (1) studies that investigate which search algorithms perform best when selecting test cases (in specific domains); (2) studies that investigate which are the fitness functions that yield best cost-effectiveness; and (3) studies that propose algorithmic enhancement focused on test case selection. This paper focuses on this last group. Within this last group, different perspectives have been proposed, such as, seeding the initial population [5,6], injecting diversity during the search process [29] and proposing novel crossover operators [27]. Unlike all these studies, our algorithm investigates enhancements in the mutation operator. To the best of our knowledge, this is the first paper that investigates enhancing search-based test case selection from the mutation operator perspective.

Besides test selection, other studies have studied different mutation operators with different goals, such as, enhancing differential evolution [13]. As for software engineering tasks, to the best of our knowledge, the only work focusing on the mutation operator relates to Guizzo et al., [18] who focus on pattern-based mutation operators to optimize product line architectures, i.e., a different goal as our's.

5 Conclusion and Future Work

Search-based test case selection requires different genetic operators. This paper lies in the context of the mutation operator, for which, up to now, to the best of our knowledge, has not been investigated for the context of test case selection. Specifically, instead of all test cases having the same probability for being selected or removed from the original test suite, we advocate for having different probabilities based on their adequacy score. This fosters the removal of weak test cases and the selection of the strong ones. We evaluate this novel approach for selecting test cases in a dataset involving 4 case study systems related to CPSs and 4 different fitness function combinations for each (except for the CW, for which one of them did not have sense to be applied). The mutation operator showed higher cost-effectiveness in 8 of the 15 studied scenarios. Specifically, our findings suggest that our approach is beneficial when the time budget to execute test cases is low. However, in 5 of the 15 studied cases, the cost-effectiveness of the approach was reduced when compared to the traditional mutation operator.

Future research avenues include approaches to gain confidence in our technique so as to be appropriate in all situations. To this end, we aim at investigating two core aspects. Firstly, we want to investigate strategies to include the test execution time in the mutation operator, as we found that many most "adequate" test cases took longer execution time. Secondly, we want to investigate the combination of metrics between fitness function and the adequacy criterion used for extracting the probability metrics. Besides these aspects, we would like to further validate the performance of our approach in other contexts and domains, such as those related to continuous integration (CI).

Replication Package: The full replication package can be found here: `https://dx.doi.org/10.6084/m9.figshare.23998029`

Acknowledgments

This work was partially founded by the Basque Government through their Elkartek program (EGIA project, ref. KK-2022/00119). The authors are part of the Software and Systems Engineering research group of Mondragon Unibertsitatea (IT1519-22), supported by the Department of Education, Universities and Research of the Basque Country.

References

1. Almulla, H., Gay, G.: Learning how to search: Generating effective test cases through adaptive fitness function selection. Empirical Software Engineering **27**(2), 1–62 (2022)
2. Arcuri, A., Briand, L.: A practical guide for using statistical tests to assess randomized algorithms in software engineering. In: 2011 33rd International Conference on Software Engineering (ICSE). pp. 1–10. IEEE (2011)

3. Arrieta, A.: Is the revisited hypervolume an appropriate quality indicator to evaluate multi-objective test case selection algorithms? In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 1317–1326 (2022)
4. Arrieta, A.: Multi-objective metamorphic follow-up test case selection for deep learning systems. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 1327–1335 (2022)
5. Arrieta, A., Agirre, J.A., Sagardui, G.: Seeding strategies for multi-objective test case selection: an application on simulation-based testing. In: Proceedings of the 2020 Genetic and Evolutionary Computation Conference. pp. 1222–1231 (2020)
6. Arrieta, A., Valle, P., Agirre, J.A., Sagardui, G.: Some seeds are strong: Seeding strategies for search-based test case selection. ACM Transactions on Software Engineering and Methodology **32**(1), 1–47 (2023)
7. Arrieta, A., Wang, S., Arruabarrena, A., Markiegi, U., Sagardui, G., Etxeberria, L.: Multi-objective black-box test case selection for cost-effectively testing simulation models. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 1411–1418 (2018)
8. Arrieta, A., Wang, S., Markiegi, U., Arruabarrena, A., Etxeberria, L., Sagardui, G.: Pareto efficient multi-objective black-box test case selection for simulation-based testing. Information and Software Technology **114**, 137–154 (2019)
9. Arrieta, A., Wang, S., Sagardui, G., Etxeberria, L.: Search-based test case prioritization for simulation-based testing of cyber-physical system product lines. Journal of Systems and Software **149**, 1–34 (2019)
10. Assunção, W.K.G., Colanzi, T.E., Vergilio, S.R., Pozo, A.: A multi-objective optimization approach for the integration and test order problem. Information Sciences **267**, 119–139 (2014)
11. Binh, N.T., Tung, K.T., et al.: A novel fitness function of metaheuristic algorithms for test data generation for simulink models based on mutation analysis. Journal of Systems and Software **120**, 17–30 (2016)
12. Birchler, C., Khatiri, S., Derakhshanfar, P., Panichella, S., Panichella, A.: Single and multi-objective test cases prioritization for self-driving cars in virtual environments. ACM Transactions on Software Engineering and Methodology **32**(2), 1–30 (2023)
13. Das, S., Abraham, A., Chakraborty, U.K., Konar, A.: Differential evolution using a neighborhood-based mutation operator. IEEE transactions on evolutionary computation **13**(3), 526–553 (2009)
14. De Lucia, A., Di Penta, M., Oliveto, R., Panichella, A.: On the role of diversity measures for multi-objective test case selection. In: 2012 7th International Workshop on Automation of Software Test (AST). pp. 145–151. IEEE (2012)
15. Fraser, G., Arcuri, A.: Whole test suite generation. IEEE Transactions on Software Engineering **39**(2), 276–291 (2012)
16. Fraser, G., Arcuri, A., McMinn, P.: A memetic algorithm for whole test suite generation. Journal of Systems and Software **103**, 311–327 (2015)
17. Gay, G.: Generating effective test suites by combining coverage criteria. In: International Symposium on Search Based Software Engineering. pp. 65–82. Springer (2017)
18. Guizzo, G., Colanzi, T.E., Vergilio, S.R.: A pattern-driven mutation operator for search-based product line architecture design. In: Search-Based Software Engineering: 6th International Symposium, SSBSE 2014, Fortaleza, Brazil, August 26-29, 2014. Proceedings 6. pp. 77–91. Springer (2014)

19. Lachmann, R., Felderer, M., Nieke, M., Schulze, S., Seidl, C., Schaefer, I.: Multi-objective black-box test case selection for system testing. In: Proceedings of the Genetic and Evolutionary Computation Conference. pp. 1311–1318 (2017)
20. Ling, X., Menzies, T.: Faster multi-goal simulation-based testing using doless (domination with least square approximation). arXiv preprint arXiv:2112.01598 (2021)
21. Ling, X., Menzies, T.: What not to test (for cyber-physical systems). IEEE Transactions on Software Engineering (2023)
22. Liu, B., Lucia, Nejati, S., Briand, L.C., Bruckmann, T.: Simulink fault localization: an iterative statistical debugging approach. Software Testing, Verification and Reliability **26**(6), 431–459 (2016)
23. Liu, B., Nejati, S., Briand, L.C., et al.: Effective fault localization of automotive simulink models: achieving the trade-off between test oracle effort and fault localization accuracy. Empirical Software Engineering **24**(1), 444–490 (2019)
24. Matinnejad, R., Nejati, S., Briand, L.C., Bruckmann, T.: Automated test suite generation for time-continuous simulink models. In: proceedings of the 38th International Conference on Software Engineering. pp. 595–606 (2016)
25. Matinnejad, R., Nejati, S., Briand, L.C., Bruckmann, T.: Test generation and test prioritization for simulink models with dynamic behavior. IEEE Transactions on Software Engineering **45**(9), 919–944 (2018)
26. McMinn, P.: Search-based software test data generation: a survey. Software testing, Verification and reliability **14**(2), 105–156 (2004)
27. Olsthoorn, M., Panichella, A.: Multi-objective test case selection through linkage learning-based crossover. In: International Symposium on Search Based Software Engineering. pp. 87–102. Springer (2021)
28. Pan, R., Ghaleb, T.A., Briand, L.: Atm: Black-box test case minimization based on test code similarity and evolutionary search. In: 2023 IEEE/ACM 45th International Conference on Software Engineering (ICSE). pp. 1700–1711. IEEE (2023)
29. Panichella, A., Oliveto, R., Di Penta, M., De Lucia, A.: Improving multi-objective test case selection by injecting diversity in genetic algorithms. IEEE Transactions on Software Engineering **41**(4), 358–383 (2014)
30. Papadakis, M., Jia, Y., Harman, M., Le Traon, Y.: 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, pp. 936–946. IEEE (2015)
31. Pradhan, D., Wang, S., Ali, S., Yue, T., Liaaen, M.: 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). pp. 367–378. IEEE (2017)
32. Saber, T., Delavernhe, F., Papadakis, M., O'Neill, M., Ventresque, A.: A hybrid algorithm for multi-objective test case selection. In: 2018 IEEE Congress on Evolutionary Computation (CEC). pp. 1–8. IEEE (2018)
33. Wang, S., Ali, S., Gotlieb, A.: Cost-effective test suite minimization in product lines using search techniques. Journal of Systems and Software **103**, 370–391 (2015)
34. Wang, S., Ali, S., Yue, T., Li, Y., Liaaen, M.: 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. pp. 631–642 (2016)
35. Yoo, S., Harman, M.: Pareto efficient multi-objective test case selection. In: Proceedings of the 2007 international symposium on Software testing and analysis. pp. 140–150 (2007)