Employing Multi-Objective Search to Enhance Reactive Test Generation and Prioritization for Testing Industrial Cyber-Physical Systems

Aitor Arrieta
University of Mondragon
Mondragon, Spain

Shuai Wang
Testify
Oslo, Norway

Urtzi Markiegi, Goiuria Sagardui, Leire Etxeberria
University of Mondragon
Mondragon, Spain

ABSTRACT

Cyber-Physical Systems (CPSs) integrate digital cyber computations with physical processes. Testing these systems in a time consuming task. Furthermore, the input space for test cases is huge, and several issues need to be considered when generating them. In [2] we tackle the test case generation and prioritization problem for CPSs. The approach is empirically evaluated with four different case studies from different domains and complexities. Five pareto-based algorithms were evaluated and overall, the NSGA-II algorithm outperformed the remaining algorithms. The NSGA-II algorithm improved RS by 43.8% on average for each objective and 49.25% for the Hypervolume quality indicator.

ACM Reference Format:
Aitor Arrieta, Shuai Wang, and Urtzi Markiegi, Goiuria Sagardui, Leire Etxeberria. 2018. Employing Multi-Objective Search to Enhance Reactive Test Generation and Prioritization for Testing Industrial Cyber-Physical Systems. In *GECCO '18 Companion: Genetic and Evolutionary Computation Conference Companion, July 15–19, 2018, Kyoto, Japan,* Jennifer B. Sartor, Theo D'Hondt, and Wolfgang De Meuter (Eds.). ACM, New York, NY, USA, Article 4, 2 pages. https://doi.org/10.1145/3205651.3208212

1 INTRODUCTION

Cyber-Physical Systems (CPSs) integrate digital cyber technologies with parallel physical processes [5]. These systems have been cataloged as "untestable" due to the difficulty to test them [4]. One such difficulty lies on the generation of test cases, as the input space for CPSs is huge. Thus, search algorithms are appropriate to generate test cases, and in our work we propose a method based on multi-objective search algorithms for that [2].

The test cases that we aim to generate are reactive test cases. These test cases stimulate the inputs of the system and observe its outputs to react om them. In figure 1, an example of three reactive test cases for testing the cruise control system of a car is proposed. In our paper [2], we propose a whole test suite generation approach with prioritization for testing CPSs based on reactive test cases. The test prioritization in the context of reactive test cases is important for two main reasons. Firstly, to detect faults as fast as possible. Secondly, because the test execution time of each test case can vary depending on the previously prioritized test case, as demonstrated in previous work [3].

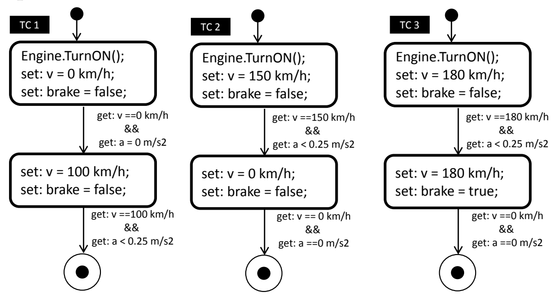

Figure 1: Example of three reactive test cases for the cruise control system testing of a vehicle [2]

2 METHOD

2.1 Cost-effectiveness measures

Four different cost effectiveness measures were selected [2]. The first one was functional requirements coverage, which measures the number of requirements that a specific test suite covers. The second measure was related to similarity of test cases. Notice that the more dissimilar the test cases are, the higher the chances to detect faults. In this context, a similarity measure was proposed to measure the distance between reactive test cases. The third measure was the prioritization-aware similarity. This measure aimed at prioritizing the most dissimilar test cases with its preceeding test cases. This would allow to detect faults sooner. The last measure was related to cost. In this case we selected the test execution time, which measured the time required by a test suite to execute.

2.2 Solution representation

A solution returned by our search algorithms in this context was a prioritized test suite (TS) composed by at least one test case (TC), i.e., $TS = \{TC_1, TC_2, ..., TC_N\}$. Each TC was composed by a set of states (S), (i.e., $TC_i = \{S_1, S_2, ..., S_{N_{TC_i}}\}$), where N_{TC_i} is the number of states that the i-th test case is composed of). Finally, each state had a predefined set of stimulation signal that must be connected to the simulation model of the CPS. These stimuli signals were based on the CPS model being tested and each of them had a maximum and a minimum value.

2.3 Crossover operator

We implemented a single-point crossover that exchanged test cases of two different test suites. Furthermore, given the nature of test

GECCO '18 Companion, July 15–19, 2018, Kyoto, Japan

ACM ISBN 978-1-4503-5764-7/18/07.
https://doi.org/10.1145/3205651.3208212

prioritization, this crossover eliminated those repeated test cases in a specific test suite. Refer to the original paper for further details [2].

2.4 Mutation operators

Two mutation operators were proposed (mutation at test suite level and mutation at test case level), each of which operated at one level, and had specific mutation sub-operators.

Regarding the mutation at test suite level, three sub-mutation operators were developed: (1) addition of a new test case, (2) removal of a test case and (3) exchange of two test cases. The first sub-mutation operator randomly selected a new test case in a new position. The second sub-mutation operator randomly removed one of the test cases from the test suite. The third sub-mutation operator randomly picked two test cases and swapped their position.

As for the mutation operator at test case level, four sub-mutation operators were developed: (1) state addition sub-mutation operator (2) state removal sub-mutation operator (3) state exchange sub-mutation operator and (4) change of variable sub-mutation operator for the test case level. The first sub-mutation operator randomly selected one test case and it randomly added a new state in a random position. The second operator randomly selected a test case and it randomly removed one of its states. The third sub-mutation operator randomly selected a test case, and it exchanged the position of two of its states. The last operator randomly selected a test case and changed one of its stimulation signals.

3 RESULTS

Three Research Questions were raised to evaluate our approach:

- RQ1: Are the selected multi-objective algorithms cost-effective when compared to Random Search (RS) for solving the TC generation and prioritization problem?

This RQ was defined a sanity check. We compared our approach with different multi-objective search algorithms with RS. When we ensured that these algorithms outperformed RS, we assessed which multi-objective search algorithms could assist our approach in achieving the best performance:.

- RQ2: Which of the selected multi-objective algorithms fares best when solving the TC generation and prioritization problem?

RQ2 was defined to identify the best multi-objective search algorithm to solve our problem. We chose five state-of-the-art multi-objective search algorithms: Non-dominated Sorting Genetic Algorithm-II (NSGA-II), strength pareto evolutionary algorithm 2 (SPEA2), multi-objective evolutionary algorithm based on decomposition (MOEA/D), pareto envelope-based selection algorithm II (PESA-II), and non-dominated sorting genetic algorithm III (NSGA-III). Once the best multi-objective search algorithm was obtained, we studied if different crossover and mutation rates could affect the performance of our approach, which is the key motivation of RQ3:

- RQ3: How do the designed crossover and mutation operators with different crossover and mutation rates affect the performance of our approach?

RQ3 is defined to assess the performance of the crossover and mutation operators along with different rates. To deal with this RQ, we chose in total five CXR (i.e., 0, 0.2, 0.5, 0.8, and 1) and three MR (i.e., 1/N, 2/N, and 5/N) based on the guidelines from [1].

Four case studies modeled in MATLAB/Simulink were employed in the proposed empirical evaluation. These case studies included different characteristics and complexities, and they were defined together with our industrial partners. The hypervolume (HV) quality indicator was selected as the evaluation metric for the empirical evaluation. Furthermore, the four fitness values of each selected objectives were measured.

Figure 2 shows the distribution of the algorithms when considering the HV quality indicator.[1] As for RQ1, it can be observed, generally all algorithms outperformed RS. As for RQ2, when considering the HV, the NSGA-II showed best performance. However, MOEA/D performed quite well when considering both similarity measures. We selected NSGA-II to assess RQ3. We observed that when the mutation probability was N/2 the performance increased. Furthermore, all crossover rates higher than 0 outperformed with statistical significance the configuration that did not have the crossover operator (i.e., crossover rate = 0); this suggests that the proposed crossover operator works well.

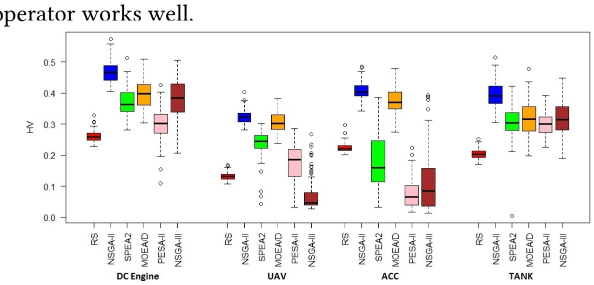

Figure 2: Distribution of the results for the HV for each algorithm in the four case studies

4 CONCLUSION

This paper proposes a test generation and prioritization approach based on multi-objective search algorithms for CPSs. To this end, corresponding crossover and mutation operators were developed and integrated within five different pareto-based search algorithms. An empirical evaluation with four case studies showed that the NSGA-II was the algorithm performing best.

REFERENCES

[1] Andrea Arcuri and Gordon Fraser. 2013. Parameter tuning or default values? An empirical investigation in search-based software engineering. *Empirical Software Engineering* 18, 3 (2013), 594–623.

[2] Aitor Arrieta, Shuai Wang, Urtzi Markiegi, Goiuria Sagardui, and Leire Etxeberria. 2018. Employing Multi-Objective Search to Enhance Reactive Test Case Generation and Prioritization for Testing Industrial Cyber-Physical Systems. *IEEE Transactions on Industrial Informatics* 14, 3 (2018), 1055–1066.

[3] Aitor Arrieta, Shuai Wang, Goiuria Sagardui, and Leire Etxeberria. 2016. Test Case Prioritization of Configurable Cyber-Physical Systems with Weight-Based Search Algorithms. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016 (GECCO '16)*. ACM, New York, NY, USA, 1053–1060. https://doi.org/10.1145/2908812.2908871

[4] Lionel Briand, Shiva Nejati, Mehrdad Sabetzadeh, and Domenico Bianculli. 2016. Testing the Untestable: Model Testing of Complex Software-intensive Systems. In *Proceedings of the 38th International Conference on Software Engineering Companion (ICSE '16)*. ACM, 789–792. https://doi.org/10.1145/2889160.2889212

[5] Edward A. Lee and Sanjit A. Seshia. 2015. *Introduction to Embedded Systems - A Cyber-Physical Systems Approach* (2 ed.). Lee and Seshia.

[1] detailed results can be found in https://sites.google.com/view/tii2017