Multi-requirement Parametric Falsification

Matteo Camilli
Politecnico di Milano
Milano, Italy
matteo.camilli@polimi.it

Raffaela Mirandola
Karlsruhe Institute of Technology
Karlsruhe, Germany
raffaela.mirandola@kit.edu

ABSTRACT

Falsification is a popular simulation-based testing method for Cyber-Physical Systems to find inputs that violate a formal requirement. However, detecting violations considering multiple probabilistic requirements simultaneously with a dense space of changing factors in the execution scenario is an open problem. We address this problem by proposing a novel approach that combines parametric model checking and many-objective optimization. Results of a preliminary empirical evaluation show the effectiveness of the approach compared to selected baseline methods.

CCS CONCEPTS

• **Software and its engineering → Software verification and validation**; **Extra-functional properties**.

KEYWORDS

falsification, multi-requirement, probabilistic requirements

ACM Reference Format:
Matteo Camilli and Raffaela Mirandola. 2024. Multi-requirement Parametric Falsification. In *2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion '24), April 14–20, 2024, Lisbon, Portugal*. ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3639478.3643120

1 INTRODUCTION

Falsification is an established testing method for Cyber-Physical Systems (CPSs), such as autonomous driving systems, to detect violations of safety (or dependability) requirements usually expressed through a formal notation.However, existing falsification approaches exhibit limitations. According to Haq et al. [4], the majority of them do not consider there are often many (possibly independent) requirements R_1, ..., R_n that must be considered together in practice. Testing each requirement R_i often needs expensive simulations. On the other hand, when the simulation data is available checking the satisfaction of R_i is very efficient. Such a cost imbalance makes it undesirable to test each requirement R_i individually, even though this represents a common approach. An alternative approach is to test the conjunction $\Phi = R_1 \wedge ... \wedge R_n$. In this case, Φ is violated if R_i does not hold for any i. Although this solution is computationally efficient, it leads to traceability issues since it hides the contributions of each requirement to the violation of Φ. Further, existing techniques dealing with multiple requirements do not account for the *reproducibility issue* [1]. According to Afzal et al. [1], nondeterminism in CPS simulators is a critical source of difficulties when testing since multiple simulations may lead to different outcomes. Thus, requirement violations are stochastic phenomena that shall satisfy *probabilistic* requirements to quantify the likelihood of violations rather than spot a possible occurrence.

To overcome these limitations, we present Parametric Falsification (PF), a novel falsification approach that deals with multiple probabilistic requirements. PF combines two distinct techniques: (1) parametric model checking [2] to shift part of the problem complexity offline by pre-computing, for each probabilistic requirement, the corresponding numerical intervals constraining the parameters of a stochastic specification describing the testing scenario; (2) many-objective optimization [3] to push (many) parameters of the stochastic specification out of the pre-computed constraints. The two techniques are applied in sequential steps: (1) *offline analysis* of the testing scenario and the target probabilistic requirements through parametric model checking [5], and then (2) *online optimization* to automatically generate test cases that increase the likelihood of violations using many-objective search. We also propose two alternative versions of the online optimization step relying on Evolutionary Search [3] (ES) and Reinforcement Learning [10] (RL) algorithms to study and compare their effectiveness.

2 PARAMETRIC FALSIFICATION

We describe the two steps: (1) offline analysis and (2) online optimization in its two alternative versions based on many-objective ES, and many-objective RL.

Offline Analysis. The analysis step shifts part of the complexity of the problem offline. It takes as input the set of requirements and, for each one of them, it calculates constraints for its satisfaction in terms of numerical intervals. These intervals constrain the parameters of a stochastic model describing the simulated scenario. The stochastic model is a parametric Markov Decision Process [8] (pMDP) automatically generated from a high-level specification of the testing scenario in terms of a finite state-machine. Given a pMDP, requirements can be expressed using Probabilistic Computation Tree Logic [5] (PCTL) formulas. Given a set of PCTL requirements and the pMDP specification of the scenario, we use parametric model checking to calculate a mapping from sets of intervals (parameter valuations) to truth values (true/false) for each requirement. Essentially, for each requirement R_j, we pre-compute the interval I_{x_i} for each parameter x_i. The set of intervals $R_j^c = \{I_{x_i}, \forall i\}$ represents the constraint for the satisfaction of R_j. The set of constraints (one for each requirement) represents the input of the online optimization.

ICSE-Companion '24, April 14–20, 2024, Lisbon, Portugal

ACM ISBN 979-8-4007-0502-1/24/04…$15.00
https://doi.org/10.1145/3639478.3643120

Online Optimization. Our novel approach defines the optimization problem as follows. Let S be the CPS under test and F the set of controllable factors of the simulation. Let a test case $t = (f_1, ..., f_K)$ be a tuple of values assigned to factors F for a simulation of S. Consider for instance simulation of an autonomous driving system. Here factors F may include vehicle speed, position, speed, orientation of the pedestrian, and road shape. Given a set of probabilistic requirements $\{R_1, R_2, ...\}$, the degree of a violation *dist* for a requirement R_j produced by S under test case t depends on the actual value of the parameters compared to the numerical intervals prescribed by the constraint R_j^c. The actual value of the parameters can be measured by monitoring the simulation of S for t. To estimate the concentration parameters we use a common approach based on Bayesian inference [9]. For each x_i, we estimate the Credible Interval $CI(x_i)$, by calculating the highest density region of the distribution for a given credibility level (e.g., 95%). At the end of the execution of a test case t, we measure the degree of violation for each x_i. Ideally, we would like to find the test cases that yield the smallest negative *dist* value for all x_i. These represent the most severe requirement violations. Thus, our optimization problem uses n *dist* functions as objectives $dist(x_1), ..., dist(x_n)$: one objective function for each specification parameter. The set of controllable factors F defines instead the search space.

Optimization using Evolutionary Search. We address the optimization problem described above by using many-objective ES. We use NSGA-III [3], a mainstream many-objective optimization algorithm. We call this approach PF-ES. Similar to existing work, PF-ES uses the notion of archive to keep the set of test cases satisfying the objectives. It takes as input a set of specification parameters X, a set of constraints C (each one for each requirement), and returns an archive (test suite) A that aims to maximally achieve individual objectives, that is, minimize $dist(x_1), ..., dist(x_n)$ according to C.

Optimization using Reinforcement Learning. We adopt an approach similar to MORLOT [4]. The approach combines: Q-Learning [10] to search for effective changes of controllable factors that cause violations; and many-objective search for test case generation to achieve many independent objectives individually within a given budget. To take into account many objectives simultaneously (i.e., minimization of *dist* for each parameter), the algorithm extends standard Q-Learning algorithms by maintaining multiple Q-tables, each of them addressing one objective. In our problem, each Q-table captures the best action (change of controllable factors) for one objective in a given state (current state of controllable factors). Since different actions can be chosen for different objectives, the algorithm selects the Q-table based on the objective that achieved the maximum reward (smallest *dist* value) in the previous iteration. This is because the corresponding objective is closer to violations.

Our falsification approach PF that makes use of this method is referred to as PF-RL. Notice that PF-RL takes inspiration from MORLOT but is different. MORLOT uses requirements directly as objectives, while objectives in PF-RL are determined by specification parameters. MORLOT dynamically changes the factors of a scenario over multiple steps of a single simulation, while PF-RL keeps controllable factors constant for each simulation to collect a sample and estimate the probability of violations under a given assignment.

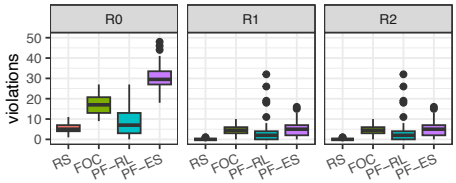

Figure 1: Effectiveness of PF compared to selected baseline approaches in terms of number of violations.

3 PRELIMINARY EVALUATION

We show some preliminary results of our empirical evaluation using an existing benchmark in the area of autonomous driving [7]. The tested scenario includes a vehicle equipped with an automated emergency braking component. The vehicle drives down an urban street, while a pedestrian starts crossing. Safety requirements here predicate, for instance, over the probability of violating a given protective human-vehicle distance, and the probability that the vehicle goes out of the lane. Our experiments focus on the effectiveness of PF compared to Random Search (RS) and a selected baseline for multi-requirement falsification, so-called Focused Falsification [6] (FOC). Here we measure the effectiveness in terms of requirement violations. Figure 1 illustrates the distribution of violations for each requirement (R_0,R_1,R_2) using RS, FOC, PF-RL, and PF-ES over 30 executions with the same budget (900 simulations). Preliminary results show that PF significantly outperforms RS. PF-ES is the best option and it is always better than FOC and PF-RL.

4 CONCLUSION

We present PF, a novel approach for CPS falsification that combines parametric model checking and many-objective optimization to falsify multiple (independent) probabilistic requirements. We plan to extend PF by leveraging surrogate models to reduce the cost of online optimization while preserving its effectiveness.

REFERENCES

[1] A. Afzal, D. S. Katz, C. Le Goues, and C. S. Timperley. 2021. Simulation for Robotics Test Automation: Developer Perspectives. In *2021 14th IEEE ICST*. 263–274.
[2] Conrado Daws. 2004. Symbolic and Parametric Model Checking of Discrete-Time Markov Chains. In *ICTAC'04*. Springer-Verlag, 280–294.
[3] K. Deb and H. Jain. 2014. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation* 18, 4 (2014), 577–601.
[4] F. Ul Haq, D. Shin, and L. C. Briand. 2023. Many-Objective Reinforcement Learning for Online Testing of DNN-Enabled Systems. In *IEEE/ACM ICSE 2023*. 1814–1826.
[5] M. Kwiatkowska, G. Norman, and D. Parker. 2011. PRISM 4.0: Verification of Probabilistic Real-Time Systems. In *Computer Aided Verification*. Springer.
[6] J. Lidén Eddeland, A. Donzé, and K. Åkesson. 2022. Multi-Requirement Testing Using Focused Falsification. In *Proceedings of the 25th ACM International Conference on Hybrid Systems: Computation and Control*. ACM.
[7] S. Nejati, L. Sorokin, Safin D., F. Formica, M. M. Mahboob, and C. Menghi. 2023. Reflections on Surrogate-Assisted Search-Based Testing: A Taxonomy and Two Replication Studies based on Industrial ADAS and Simulink Models. *Information and Software Technology* 163 (2023), 107286.
[8] M. L. Puterman. 2014. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons.
[9] C. P. Robert. 2007. *The Bayesian Choice: From Decision-Theoretic Foundations to Computational Implementation* (2nd ed.). Springer.
[10] C. J. C. H. Watkins and P. Dayan. 1992. Q-learning. *Machine Learning* 8, 3 (01 May 1992), 279–292.