POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

Self-Adaptation under Uncertainty using Bayesian Model Averaging and Many Objective Search

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

Author: Umberto Messuti

Advisor: Prof. Raffaela Mirandola

Co-advisor: Matteo Camilli

Academic year: 2022-2023

1. Introduction

Cyber-Physical Systems (CPS) are a type of computing systems with the peculiarity of combining physical and computational elements; they are used to interact with the physical world in order to achieve a desired goal. For this purpose, a CPS has to self-adapt to changes in the environment, i.e., find a new internal configuration, in order to guarantee continuous operation while avoiding potential safety risks or more in general dependability issues. The resiliency of CPS is closely linked to the system's ability to achieve equilibrium, a condition in which the system meets all of its requirements. A classical approach to achieve resiliency is based on identifying a limited number of conditions under which the system should continuously operate while maintaining equilibrium. However, a more dynamic and flexible approach involves the integration of statistical models used to make the system able to learn from past events and adapt autonomously to new situations. In particular, in case of disequilibrium, find a new configuration of the CPS that is most likely to bring the system back to its equilibrium condition where all the requirements of interest are satisfied. Choosing the right statistical model for the system is a complex task hence we make use Bayesian Model Averaging (BMA) [5], a useful technique for managing the uncertainty in model selection. We will use BMA to find an appropriate statistical model that predicts the satisfaction of the requirements under changing operating conditions, composed of environmental factors and system configurations. Our work consists in creating a framework that makes use of the BMA modelling abilities to autonomously search for viable new configurations for the CPS. Specifically, we developed a CPS simulation tool and a program that implements BMA together with Many Objective search with the goal of enforcing the robustness of CPS. The former is used to simulate CPS, giving us the ability to check for the requirements satisfaction, while the latter represents the application of an optimization search problem used to find out a set of optimal configurations given the starting configuration and a set of requirements of a CPS.

2. Starting Point

2.1. RUNE

RUNE (RUNtime Equilibrium verification) [2] is an approach, based on the notion of equilibrium,

used to check at run-time if a CPS meets all its requirements. The approach is defined by four steps.
The first phase involves the creation of a model of the system's behaviours using five components: parametric Markov Decision Process (pMDP), uncertain regions, semantic space (SS), prior knowledge, and Probabilistic Computational Tree Logic (PCTL).
The modeling formalism that we will use to schematize the systems is the Markov Decision Process (MDP) [8] which is defined as a tuple consisting of a finite set of states, a set of actions, a transition function, and a reward function. A parametric MDP (pMDP) is an extention of the MDP in which there are parametric values in the transition functions. In the context of pMDPs, we refer to prior knowledge as any information or belief about the likely value of these parameters before any observations or data are collected. An uncertain region is the union of an action and the transitions associated to it in which parameters do exist. The SS consists of a set of variables (SS variables) through which the parameters of the uncertain regions are manipulated, and the PCTL are a formal representation of the desired reliability or availability of a system and are expressed in terms of probabilistic constraints.
In the second phase equilibrium constraints are computed. These constraints are defined as value intervals on each of the pMDP parameters to ensure the satisfiability of all PCTL dependability requirements, regardless of any possible assignment of the SS.
The third phase is meant to check if the behavior of CPS at run-time complies with the required equilibrium conditions. During this phase, multiple samples of run-time evidence are collected. These data are used to feed the adaptive observation aging mechanism (AOAM) of RUNE which uses this data to build two different sequences, one with short and one with long memory. Such sequences are leveraged to yield a more accurate estimate on the system parameters that take into account both the long term and the possible sudden changes observed in the shorter term. The equilibrium condition (not to be confused with the equilibrium constraints) related to an uncertain region holds when all the estimations of its parameters are within their specified equilibrium constraints, i.e. the equilibrium constraints are satisfied for all the parameters included in the uncertain region. Alternatively, the disequilibrium condition holds when at least one equilibrium constraint is not satisfied.
In the last phase, in case the disequilibrium condition holds, an enforcement mechanism that employs an optimal policy is triggered. The policy aims to maximize the probability of enforcing equilibrium by assigning low rewards to transitions that lead to uncertain regions in disequilibrium, and high rewards to transitions that lead to equilibrium.

2.2. TUNE

TUNE (Taming model UNcErtainty) [1] aims to mitigate uncertainty related to model selection by introducing the Bayesian Model Averaging (BMA) technique. BMA, accounts for model selection uncertainty by combining the predictions obtained from multiple logistic models, weighted by their posterior probabilities (i.e. the probability of the model given the data). Logistic models [7] are chosen, over other alternatives, since they are meant to predict only binary values as the requirement fulfillment is expressed as *True* or *False*. Given a set of variables X, that constitutes a CPS configuration, and a single requirement the BMA estimates predict the likelihood of a new configuration that fulfills the requirement. The BMA model is obtained from the model space $M = \{M_1, ..., M_n\}$ where $n = 2^{|X|} - 1$ and the i^{th} model M_i is obtained by including in the predictor only a subset of the explanatory variables in X; the more the variables of the configuration, the greater the number of models in M. Besides BMA, TUNE also leverages many-objective search [6] to explore the configuration space. It finds alternative assignments of the configuration variables that are capable of minimizing the adaptation cost while maximizing the likelihood of satisfying the requirement.

3. Our Proposal

3.1. Problem Formulation

We focus on developing an optimization problem that, given a CPS with a set of variables and requirements, has the goal of finding the best

variable configuration in order to keep satisfied all the requirements. The aforementioned variables are a set of configuration and environment dimensions describing some internal and external quantities; the optimization problem aims to modify only the former. We model the CPS as a pMDP and the optimization problem involves a set of logistical models (one for each requirement) to determine whether or not the requirements are satisfied while minimizing the cost of applying a new configuration.
The first step of our work consists in the the development of a pMDP simulator used to gather data from a set of CPS's simulations. Such data has been used in the second part of our study in which we focus on the definition and solution of the aforementioned optimization problem.

3.2. Simulator

The simulator is the first essential component for our proposal and takes inspiration from the RUNE approach described in Section 2.1. This includes the definition of a CPS with partial knowledge using the pMDP formalism, runtime verification of the satisfaction of equilibrium constraints and their enforcement. The most important parts needed to comprehend this component are its input, software architecture and execution result definition.

Input For what concerns the input definition, three are the main elements: the pMDP, its uncertain regions and the SS.
The pMDP element serves as the foundation for all the other specifications. To configure it, states, actions, and transition functions must be describable. A state is defined by a unique code, a label, and its type (Observable or Controllable). Each state must have one or more defined actions, which are tuples containing an action identifier, a probability, and the outcome state identifier.
Uncertain regions are defined by the combination of a state and one of its actions. Furthermore, it is also needed to specify the prior knowledge and the equilibrium constraints. The former is a series of integers used as parameters for the Dirichlet prior density function, while the latter consists of pairs of values that define constraint intervals for the probability of each outcome. Instead of calculating the equilibrium constraints offline, as in the RUNE approach, we assume them for already computed.
The SS is a set of configuration and environment dimensions that describe internal and external quantities affecting uncertain regions in an pMDP. Each variable in the SS (SSV) is defined by a domain, interval, default value, and one or more combinations specifying the uncertain areas it impacts together with the function (in terms of set of instructions) to apply. If different SSV functions affect the same uncertain region, they are executed consecutively. The first one uses its own internal assignment together with the actual values of the uncertain region's parameters to determine a new transition function. The subsequent procedures use their own internal value together with the result of the previously applied procedure.

Architecture Since a flexible and adaptable framework for future use and development was needed, we decided to employ the "event-driven" design pattern. It manages events that occur within the system by exchanging them between system components that can customly react to each received event. The simulator architecture includes an event producer and multiple event consumers that subscribe to topics of interest. The event producer can be seen as the main component as it implements the behaviours of the pMDP. Everything else is implemented as an extension of it. To achieve our goal, two extensions were built: a "monitor" that updates the estimate of uncertain areas and a "log and plot" component that logs and plots all the new estimates calculated for each uncertain area.

Execution Result The products of the simulator, functional for the subsequent phases of our proposal, consist in the final state of the simulated pMDP and a set of plots. The former contains all the attributes of the pMDP, such as states, actions and estimates, allowing us to conduct a posteriori investigations in order to assess the requirements satisfaction. The latter depicts the evolutions of the estimations for each state-action-outcome combination of all uncertain areas and can be seen as a visual feedback to understand what happened during the execution. An example plot is provided in Figure 1 where, given an action a, a starting state $s0$ and an outcome $s2$, the evolutions of point and interval estimates, computed on both a longer and shorter timeline, are depicted, as well as the actual probability and the equilibrium constraint.

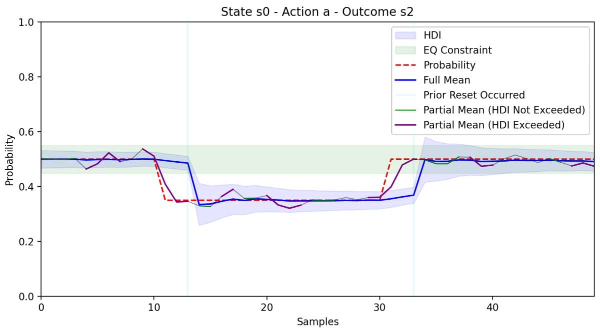

Figure 1: Example plot for a state $s0$, an action a and an outcome $s2$

3.3. BMA and Many-Objective Search for Adaptation

In this second part of the study we discuss the approach used to support adaptation, which involves the use of the CPS simulator, BMA, and many-objective search. In particular, a dataset is created based on the configuration values together with the requirement satisfaction and then the TUNE approach is extended using multiple models. Starting from the dataset obtained, multiple BMA models are trained and used in an optimization problem, which generates a set of Pareto-optimal solutions [6].

Dataset A dataset is built by running the CPS simulator multiple times with different configurations of the SS: its features are the configurations of the SS, while its labels are boolean values, one for each requirement, representing whether or not a requirement has been satisfied. The dataset is crucial for training BMA models, which will later be used to make predictions about a requirement fulfillment given a specific variable configuration.

Model Creation and Fit In this phase a BMA model is created for each requirement that has to be satisfied. The creation and fit process of each BMA model involves normalizing the predictor variables and performing the model selection. The best models are determined by comparing all possible logistic models computed by BMA. The resulting BMA model is an object capable of making predictions on new data. These predictions yield a probability value representing the likelihood of a requirement being satisfied.
During this phase, a BMA model is constructed for every requirement that must be fulfilled. This involves normalizing the predictor variables and selecting the appropriate model. The best models are determined by comparing all possible logistic models computed by BMA. The resulting BMA model is capable of producing predictions about new data, providing a probability value that indicates the probability of a requirement being met.

Many-Objective Search In this last phase we tackle the final part of our approach which is the concrete definition of the optimization problem mentioned in the problem formulation section (§ 3.1). Specifically, we implement the optimization problem as a many-objective search. Starting from a configuration of the SS, the goal is to find a set of new configurations that are able to satisfy as many requirements as possible.
The number of objectives used in the optimization problem is equal to the number of requirements to satisfy plus the cost function. The latter defines just the distance between the starting configuration and the new configurations proposed. The result of this search will be a set of Pareto-Optimal Solutions. These solutions, also known as the Pareto frontier or Pareto set, represent the best possible configurations that cannot be improved without causing a reduction in benefits for at least one between all the competing objectives. Any solution that is not in the set is considered inferior.
After defining the problem, Non-dominated Sorting Genetic Algorithm III (NSGA-III) is used to solve it. This is a popular meta-heuristic optimization algorithm used for solving many-objective optimization problems. It is an extension of the original NSGA algorithm and it uses a specialized genetic algorithm to minimize multiple, conflicting objectives simultaneously. It attempts to find a set of Pareto optimal solutions for the given problem by iteratively applying crossover, mutation, and selection operators to a population of candidate solutions until a convergence criterion is met.

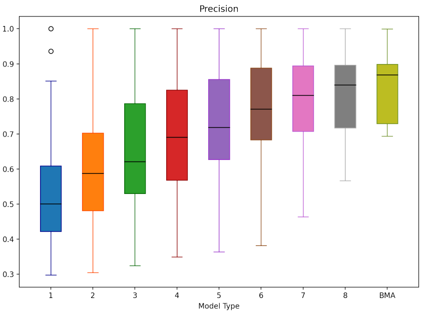

Figure 2: RQ1 Precision

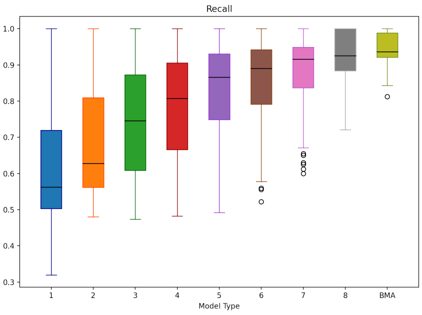

Figure 3: RQ1 Recall

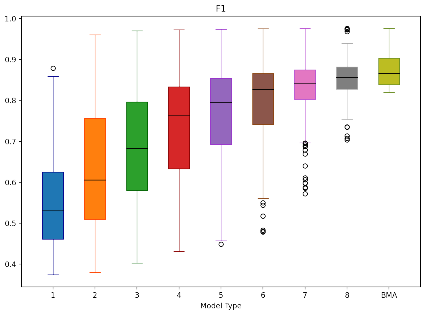

Figure 4: RQ1 F1

4. Evaluation

After testing the simulator, checking its ability, scalability, and effectiveness to recognize violations in equilibrium constraints and, when possible, avoid them, we focused the evaluation on BMA and meta-heuristic for adaptation. Concerning BMA, the objective is to investigate how much this technique costs in terms of computational time and how accurate it is compared to logistical models. Lastly, the efficiency in mitigating violations of the meta-heuristic approach is evaluated compared to a random one. The following research questions were employed:

RQ1 What is the accuracy of the BMA estimates?

RQ2 What is the cost of calculating the BMA models to mitigate the model uncertainty?

RQ3 How effective is the many-objective search in computing new configurations of the SS in order to restore the equilibrium constraints?

4.1. RQ1

To understand and quantify the potential of BMA in improving inference compared to individual Logistic models, we used a dataset of 20,000 samples, split into 80% training and 20% testing. This dataset has been gathered using a pMDP representing a rescue robot inspired by an existing case study [3].
The result shows that BMA outperforms all Logistic models in terms of precision (Figurte 2), recall (Figure 3), and F1 score (Figure 4), thanks to its ability of combining information from multiple models more efficiently. Additionally, it suggests that increasing the number of variables considered in Logistic models leads to higher results without never surpassing BMA's.

4.2. RQ2

We focus on the computational cost of calculating BMA models to mitigate model uncertainty using the deterministic and Markov Chain Monte Carlo (MCMC) [4] methods. The results (Figure 5), computed using a synthetic dataset (custom generated with 400 samples and 64 variables), show that the deterministic method is impractical as the time required increases exponentially with the addition of variables. On the other hand, the MCMC method evaluates a fraction of the possible models based on the number of variables, hence its results are more approximated than the deterministic one. As a consequence, MCMC's behaviour results in a linear duration increment as the number of variables increases, making it a useful computational tool for complex situations where exact calculations are infeasible.

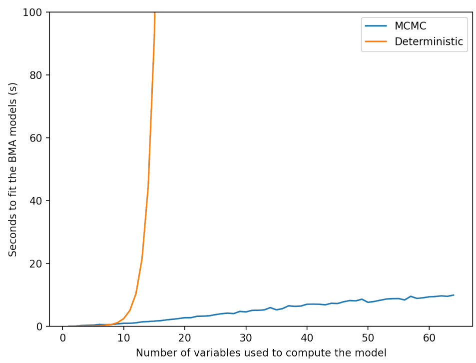

Figure 5: RQ2 BMA Computation Time

4.3. RQ3

To evaluate the efficacy of using BMA and NSGA-III jointly in computing new SS configurations capable of restoring equilibrium constraints, we utilized the same dataset described in Section 4.1. The findings reveal that this approach has a high success rate (refer to Figure 6) in identifying new configurations that meet requirements while minimizing costs (refer to Figure 7). Therefore, utilizing BMA in conjunction with NSGA-III is a more favorable approach compared to the random method, despite the increased computational time required. The degree of improvement compared to random sampling is above 150% in terms of the satisfaction of requirements and below -20% in terms of the cost of adaptation.

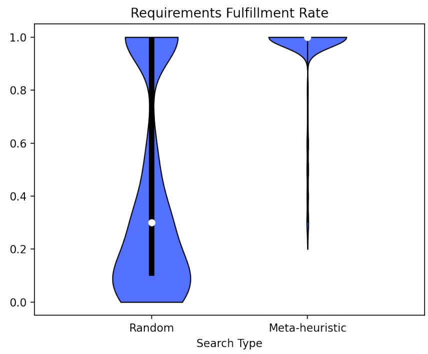

Figure 6: Success Rate

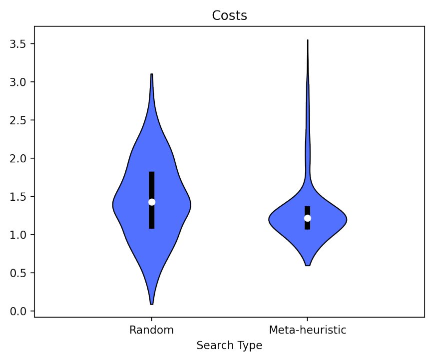

Figure 7: Cost

5. Conclusions

This work aimed to implement and extend the RUNE and TUNE approaches to satisfy as many requirements as possible in a generic CPS by computing new sets of configurations for its SS. The approach demonstrated the ability to maintain and restore equilibria in the presence of external perturbations by reaching a state that can be defined as "meta-stable". A partial and temporary equilibrium that is still susceptible to further perturbations, but from which it is always possible to recover.
The results show that the proposed approach has the potential to preserve the fulfillment of equilibrium constraints and the resilience of the CPS being examined. In future works, we aim to expand our approach to case studies in various domains, including self-driving cars. We will also continue testing additional statistical models beyond logistic regression and explore alternative meta-heuristic approaches for adaptation.

References

[1] Matteo Camilli, Raffaela Mirandola, and Patrizia Scandurra. Taming model uncertainty in self-adaptive systems using bayesian model averaging. In *Proceedings of the 17th Symposium on Software Engineering for Adaptive and Self-Managing Systems*, page 25–35, Pittsburgh Pennsylvania, May 2022. ACM.

[2] Matteo Camilli, Raffaela Mirandola, and Patrizia Scandurra. Enforcing resilience in cyber-physical systems via equilibrium verification at runtime. *ACM Transactions on Autonomous and Adaptive Systems*, page 3584364, Feb 2023.

[3] Naeem Esfahani and Sam Malek. *Uncertainty in Self-Adaptive Software Systems*, pages 214–238. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[4] D. Gamerman and H.F. Lopes. *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference, Second Edition*. Chapman & Hall/CRC Texts in Statistical Science. Taylor & Francis, 2006.

[5] Jennifer A. Hoeting, David Madigan, Adrian E. Raftery, and Chris T. Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14(4):382–401, 1999.

[6] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence)*, pages 2419–2426, 2008.

[7] F.C. Pampel. *Logistic Regression: A Primer*. Quantitative Applications in t. SAGE Publications, Incorporated, 2021.

[8] Martijn van Otterlo. Markov decision processes: Concepts and algorithms.