The Journal of Systems & Software 143 (2018) 140–158

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Efficient synthesis of robust models for stochastic systems

Radu Calinescu[a], Milan Češka[b], Simos Gerasimou[*,a], Marta Kwiatkowska[c], Nicola Paoletti[d]

[a] Department of Computer Science, University of York, UK
[b] Faculty of Information Technology, Brno University of Technology, Czechie
[c] Department of Computer Science, University of Oxford, UK
[d] Department of Computer Science, Stony Brook University, USA

ARTICLE INFO

Keywords:
Software performance and reliability engineering
Probabilistic model synthesis
Multi-objective optimisation
Robust design

ABSTRACT

We describe a tool-supported method for the efficient synthesis of parametric continuous-time Markov chains (*p*CTMC) that correspond to *robust designs* of a system under development. The *p*CTMCs generated by our RObust DEsign Synthesis (RODES) method are resilient to changes in the system's operational profile, satisfy strict reliability, performance and other quality constraints, and are Pareto-optimal or nearly Pareto-optimal with respect to a set of quality optimisation criteria. By integrating sensitivity analysis at designer-specified tolerance levels and Pareto optimality, RODES produces designs that are potentially slightly suboptimal in return for less sensitivity—an acceptable trade-off in engineering practice. We demonstrate the effectiveness of our method and the efficiency of its GPU-accelerated tool support across multiple application domains by using RODES to design a producer-consumer system, a replicated file system and a workstation cluster system.

1. Introduction

Robustness is a key characteristic of both natural (Kitano, 2004) and human-made (Phadke, 1995) systems. Systems that cannot tolerate change are prone to frequent failures and require regular maintenance. As such, engineering disciplines like mechanical and electrical engineering treat robustness as a first-class citizen by designing their systems based on established tolerance standards (e.g. International Organization for Standardization, 2010; International Organization for Standardization, 2013). By comparison, software engineering is lagging far behind. Despite significant advances in software performance and reliability engineering (Balsamo et al., 2004; Bondy, 2014; Becker et al., 2009; Fiondella and Puliafito, 2016; Stewart, 2009; Woodside et al., 2014), the quality attributes of software systems are typically analysed for point estimates of stochastic system parameters such as component service rates or failure probabilities. Even the techniques that assess the sensitivity of quality attributes to parameter changes (e.g. Gokhale and Trivedi, 2002; Lo et al., 2005; Huang and Lyu, 2005; Kamavaram and Goseva-Popstojanova, 2003; Filieri et al., 2016) focus on the analysis of a given design at a time instead of systematically designing robustness into the system under development (SUD).

To address these limitations, we propose a tool-supported method for the efficient synthesis of parametric continuous-time Markov chains (*p*CTMCs) that correspond to robust SUD designs. Our RObust DEsign Synthesis (RODES) method generates sets of *p*CTMCs that:

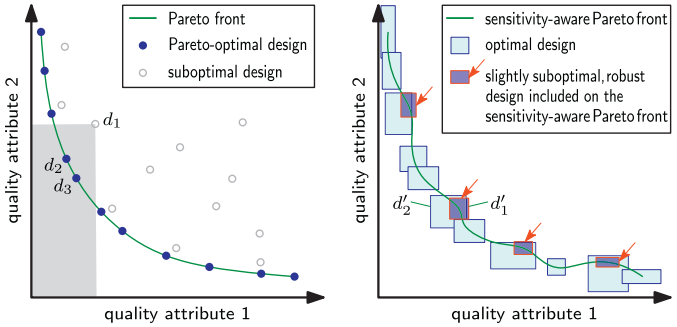
(i) are resilient to pre-specified *tolerances* in the SUD parameters, i.e., to changes in the SUD's operational profile;
(ii) satisfy strict performance, reliability and other quality constraints;
(iii) are Pareto-optimal or nearly Pareto optimal with respect to a set of quality optimisation criteria.

RODES comprises two steps. In the first step, the SUD design space is modelled as a *p*CTMC with discrete and continuous parameters corresponding to alternative system architectures and to ranges of possible values for the SUD parameters, respectively. In the second step, a multi-objective optimisation technique is used to obtain a set of low-sensitivity, Pareto-optimal or nearly Pareto-optimal SUD designs by fixing the discrete parameters (thus selecting specific architectures) and restricting the continuous parameters to bounded intervals that reflect the pre-specified tolerances. The designs that are slightly suboptimal have the advantage of a lower sensitivity than the optimal designs with similar quality attributes, achieving a beneficial compromise between optimality and sensitivity. A *sensitivity-aware Pareto dominance relation* is introduced in the paper to formally capture this trade-off.

Fig. 1 shows the differences between a traditional Pareto front, which corresponds to a fixed SUD operational profile, and a sensitivity-aware Pareto front generated by RODES, which corresponds to a SUD operational profile that can change within pre-specified bounds. Accordingly, the designs from the RODES sensitivity-aware Pareto front are bounded regions of quality-attribute values for the system. The size

* Corresponding author.
E-mail address: simos.gerasimou@york.ac.uk (S. Gerasimou).

https://doi.org/10.1016/j.jss.2018.05.013
Received 5 October 2017; Received in revised form 2 April 2018; Accepted 9 May 2018
Available online 16 May 2018



(a) Traditional Pareto front: design d_1 is suboptimal because designs d_2 and d_3 from the shaded area have better (i.e., lower) quality attributes than d_1

(b) Sensitivity-aware Pareto front: the slightly suboptimal design d'_1 belongs to the front because it has a much lower sensitivity than optimal design d'_2

Fig. 1. Traditional Pareto front (a) versus sensitivity-aware Pareto front (b) for two quality attributes that require minimisation (e.g., response time and probability of failure).

and shape of these regions convey the sensitivity of the synthesised designs to parameter changes within the pre-specified tolerances. Small quality-attribute regions correspond to particularly robust designs that cope with variations in the system parameters without exposing users to significant changes in quality attributes. These designs require reduced maintenance, and can be implemented using high-variability components that are cheaper to develop or obtain off-the-shelf than low-variability components. Large quality-attribute regions from a RODES Pareto front—while still the most robust for the quality attribute trade-offs they correspond to—are associated with designs that are sensitive to SUD parameters variations. These designs may involve high maintenance and/or development costs, so they should only be used if justified by their other characteristics (e.g. desirable quality attribute trade-offs).

To the best of our knowledge, RODES is the first solution that integrates multi-objective stochastic model synthesis and sensitivity analysis into an end-to-end, tool-supported design method. As we show in detail in Section 7, the existing research addresses the challenges associated with design synthesis (e.g. Gerasimou et al., 2015; Martens et al., 2010) and sensitivity analysis (e.g. Gokhale and Trivedi, 2002; Lo et al., 2005; Huang and Lyu, 2005; Kamavaram and Goseva-Popstojanova, 2003; Filieri et al., 2016) separately. The main contributions of our paper are:

1. The extension of the notion of *parameter tolerance* from other engineering disciplines for application to software architecture.
2. The definitions of the parametric Markov chain synthesis problem and of the sensitivity-aware Pareto dominance relation for the synthesis of robust models for stochastic systems.
3. The RODES method for the generation of sensitivity-aware Pareto fronts by integrating multi-objective probabilistic model synthesis and precise *p*CTMC parameter synthesis.
4. A GPU-accelerated tool that implements the RODES method and is available preinstalled on an easy-to-use VirtualBox instance from our project website https://www.github.com/gerasimou/RODES/wiki.
5. A repository of case studies demonstrating the successful application of RODES to a replicated file system used by Google's search engine, a cluster availability management system, and a producer-consumer system.

These contributions significantly extend our conference paper on robust model synthesis (Calinescu et al., 2017a) and the prototype probabilistic model synthesis tool (Calinescu et al., 2017b) in several ways. First, we provide a more detailed description of our solution, including a running example and new experimental results. Second, we greatly improve the scalability of RODES by integrating the GPU-accelerated analysis of candidate designs into our prototype tool (Calinescu et al., 2017b). Third, we extend the experimental evaluation to demonstrate the impact of the GPU acceleration. Finally, we present an additional case study in which we apply RODES to a producer-consumer system, and we use the systems and models from our experiments to assemble a repository of case studies available on our project website.

The remainder of the paper is organised as follows. Section 2 introduces the RODES design-space modelling language and the formalism to specify quality constraints and optimisation criteria. Section 3 defines the sensitivity-aware dominance relation and introduces the parametric Markov chain synthesis problem. We then present our method for synthesising robust designs in the form of a sensitivity-aware Pareto set, and the GPU-accelerated tool RODES implementing the method in Sections 4 and 5, respectively. Finally, we evaluate our method within three case studies in Section 6, discuss related work in Section 7, and conclude the paper with a summary and future work in Section 8.

2. Modelling and specification language for probabilistic systems

This section formalises three key elements underpinning the formulation of the robust design problem: 1) the modelling of the design space of a SUD, 2) the specification of quality attributes and requirements, and 3) the sensitivity of a design.

2.1. Design space modelling

We use a *parametric continuous-time Markov chain* (*p*CTMC) to define the design space of a SUD. To this end, we extend the original *p*CTMC definition (Han et al., 2008), where only real-valued parameters determining the transition rates of the Markov chain are considered, and assume that a *p*CTMC also includes discrete parameters affecting its state space. Our definition captures the need for both discrete parameters encoding architectural structural information (e.g. by selecting between alternative implementations of a software component) and continuous parameters encoding configurable aspects of the system (e.g. network latency or throughput). As such, a candidate system design corresponds to a fixed discrete parameter valuation and to continuous parameter values from a (small) region.

Definition 1 (*(pCTMC)*). Let K be a finite set of real-valued parameters such that the domain of each parameter $k \in K$ is a closed interval $[k^{\perp}, k^{\top}] \subset \mathbb{R}$, and D a finite set of discrete parameters such that the domain of each parameter $d \in D$ is a set $T^d \subset \mathbb{Z}$. Let also $\mathscr{P} = \times_{k \in K} [k^{\perp}, k^{\top}]$ and $\mathscr{Q} = \times_{d \in D} T^d$ be the continuous and the discrete *parameter spaces* induced by K and D, respectively. A *p*CTMC over K and D is a tuple

$$\mathscr{C}(\mathscr{P}, \mathscr{Q}) = (\mathscr{D}_S, \mathscr{D}_{init}, \mathscr{D}_{\mathbf{R}}, L), \tag{1}$$

where, for any discrete parameter valuation $q \in \mathscr{Q}$:

- $\mathscr{D}_S(q) = S$ is a finite set of *states*, and $\mathscr{D}_{init}(q) \in S$ is the initial state;
- $\mathscr{D}_{\mathbf{R}}(q)$: $S \times S \to \mathbb{R}[K]$ is a parametric rate matrix, where $\mathbb{R}[K]$ denotes the set of polynomials over the reals with variables in K;
- $L(q)$: $S \to 2^{AP}$ is a labelling function mapping each state $s \in S$ to the set $L(q)(s) \subseteq AP$ of atomic propositions that hold true in s.

A *p*CTMC $C(\mathscr{P}, \mathscr{Q})$ describes the uncountable set of continuous-time Markov chains (CTMCs) $\{\mathscr{C}(p, q) \mid p \in \mathscr{P} \wedge q \in \mathscr{Q}\}$, where each $\mathscr{C}(p, q) = (\mathscr{D}_S(q), \mathscr{D}_{init}(q), \mathbf{R}(p, q), L(q))$ is the instantiated CTMC with transition matrix $\mathbf{R}(p, q)$ obtained by replacing the real-valued parameters in $\mathscr{D}_{\mathbf{R}}(q)$ with their valuation in p.

In our approach we operate with *p*CTMCs expressed in a high-level modelling language extending the PRISM language (Kwiatkowska et al., 2011) which models a system as the parallel composition of a set of