Convex Optimization meets Parameter Synthesis for MDPs

Nils Jansen

Radboud University, Nijmegen, The Netherlands

Probabilistic Systems. Many real-world systems exhibit random behavior and stochastic uncertainties. One major example is in the field of *robotics*, where the presence of measurement noise or input disturbances requires special controller synthesis techniques [28] that achieve robustness of robot actions against uncertainties in the robot model and the environment. Formal verification offers methods for rigorously proving or disproving properties about the system behavior, and synthesizing strategies that satisfy these properties. In particular, *model checking* [25] is a well-studied technique that provides guarantees on appropriate behavior for all possible events and scenarios. Model checking can be applied to systems with stochastic uncertainties, including discrete-time Markov chains (MCs), Markov decision processes (MDPs), and their continuous-time counterparts [16]. Probabilistic model checkers are able to verify reachability properties like "the probability of reaching a set of unsafe states is $\leq 10\%$" and expected costs properties like "the expected cost of reaching a goal state is ≤ 20." A rich set of properties, specified by linear- and branching-time logics, reduces to such properties [16]. Efficient implementations are readily available in tools like PRISM [9] or STORM [20].

Introducing Parameters. Key requirements for applying model checking are a reliable system model and formal specifications of desired or undesired behaviors. As a result, most approaches assume that models of the stochastic uncertainties are precisely given. For example, if a system description includes an environmental disturbance, the mean of that disturbance should be known *before* formal statements are made about expected system behavior. However, the desire to treat many applications where uncertainty measures (e.g., faultiness, reliability, reaction rates, packet loss ratio) are not exactly known at design time gives rise to *parametric* probabilistic models [1, 14]. Here, transition probabilities are expressed as functions over system parameters, i.e., *descriptions of uncertainties*. In this setting, *parameter synthesis* addresses the problem of computing parameter instantiations leading to satisfaction of system specifications. More precisely, parameters are mapped to concrete probabilities inducing the resulting *instantiated* model to satisfy specifications. A direct application is *model repair* [8], where a concrete model (without parameters) is changed (repaired) such that specifications *are* satisfied.

Related Work. Dedicated tools like PARAM [7], PRISM [9], or PROPhESY [12] compute rational functions over parameters that express reachability probabilities or expected costs in a parametric Markov chain (pMC). These optimized tools work with millions of states but are restricted to a few parameters, as the necessary computation of greatest common divisors does not scale well with the number of parameters. Equivalently, [15, 21] employ Gaussian elimination for matrices over the field of rational functions. The resulting functions are inherently *nonlinear* and often of high degree. Solving the functions is naturally a SAT-modulo theories (SMT) problem over non-linear arithmetic. However, solving such SMT problems is *exponential in the degree of functions and the number of variables* [21].

Specific approaches to model repair rely on nonlinear programming [8] or particle-swarm optimization (PSO) [10]. Finally, parameter synthesis is equivalent to computing finite-memory strategies for partially observable MDPs (POMDPs) [24]. Such strategies may be obtained, for instance, by employing sequential quadratic programming (SQP) [4]. Exploiting this approach is not practical, though, because SQP for our setting already requires a (feasible) solution satisfying the given specification.

A Nonlinear Programming Problem. We first introduce a general nonlinear programming formulation for the verification of parametric Markov decision processes (pMDPs). The powerful modeling capabilities of nonlinear programs (NLPs) enable incorporating multi-objective properties and penalties on the parameters of the pMDP. However, because of their generality, solving NLPs to find a global optimum is difficult. Even feasible solutions (satisfying the constraints) cannot always be computed efficiently [26, 2]. In contrast, for the class of NLPs called *convex optimization* problems, efficient methods to compute feasible solutions and global optima even for large-scale problems are available [27]. We propose two (automated) methods of utilizing convex optimization for pMDPs.

Sequential Convex Programming [19]. Many NLP problems for pMDPs belong to the class of *signomial programs* (SGPs), a certain class of nonconvex optimization problems. For instance, all benchmarks available at the PARAM–webpage [13] belong to this class. Restricting the general pMDP problem accordingly yields a direct and efficient synthesis method for a large class of pMDP problems. The two main technical results are:

1. We relax nonconvex constraints in SGPs and apply a simple transformation to the parameter functions. The resulting programs are *geometric programs* (GPs) [5], a class of *convex programs*. We show that a solution to the relaxed GP induces feasibility (satisfaction of all specifications) in the original pMDP problem. Note that solving GPs is *polynomial* in the number of variables.
2. Given an initial feasible solution, we use a technique called *sequential convex programming* [5] to improve a signomial objective. This local optimization method for nonconvex problems leverages convex optimization by solving a sequence of convex approximations (GPs) of the original SGP.

Sequential convex programming is known to efficiently find a feasible solution with good, though not necessarily globally optimal, objective values [5, 6]. We initialize the sequence with a feasible solution (obtained from the GP) of the original problem and compute a *trust region*. Inside this region, the optimal value of the approximation of the SGP is at least as good as the objective value at the feasible solution of the GP. The optimal solution of the approximation is then the initial point of the next iteration with a new trust region. This procedure is iterated to approximate a local optimum of the original problem.

The Convex-concave Procedure [23]. In another direction, we transform the NLP formulation into a *quadratically-constrained quadratic program* (QCQP). As such an optimization problem is nonconvex in general, we cannot resort to polynomial-time algorithms for convex QCQPs [3]. Instead, to solve our NP-hard problem, we massage the QCQP formulation into a *difference-of-convex* (DC) problem. The convex-concave procedure (CCP) [17] yields local optima of a DC problem by a convexification towards a convex quadratic program, which is amenable for state-of-the-art solvers such as Gurobi [11].

Yet, blackbox CCP solvers [22, 18] suffer from severe numerical issues and can only solve very small problems. We integrate the procedure with a probabilistic model checker, creating a method that — realized in the open-source tool PROPhESY [12] — yields (a) an improvement of multiple orders of magnitude compared to just using CCP as a black box and (b) ensures the correctness of the solution. In particular, we use probabilistic model checking to:

- rule out feasible solutions that may be spurious due to numerical errors,
- check if intermediate solutions are already feasible for earlier termination,
- compute concrete probabilities from intermediate parameter instantiations to avoid potential numerical instabilities.

Experiments. Empirical evaluation on a large range of benchmarks shows that our approaches can solve the parameter synthesis problem for models with large state spaces and up to thousands of parameters, and is superior to all existing parameter synthesis tools [7, 9, 12] and an efficient re-implementation of an approach based on particle-swarm optimization (PSO) [10] that we created to deliver a better comparison.

We include the standard benchmarks from the PARAM or the PRISM website, and a rich selection of strategy synthesis problems obtained from partially observable MDPs (POMDPs), cf. [24].

Acknowledgements. The methods presented here are the result of fruitful and enjoyable collaborations with Murat Cubuktepe, Sebastian Junges, Joost-Pieter Katoen, Hasan A. Poonawala, Ivan Papusha, and Ufuk Topcu.

References

1. Jay K. Satia and Roy E. Lave Jr. Markovian decision processes with uncertain transition probabilities. *Operations Research*, 21(3):728–740, 1973.
2. Jean B. Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.
3. Farid Alizadeh and Donald Goldfarb. Second-order cone programming. *Math Program.*, 95(1):3–51, 2003.
4. Christopher Amato, Daniel S Bernstein, and Shlomo Zilberstein. Solving POMDPs using quadratically constrained linear programs. In *AAMAS*, pages 341–343. ACM, 2006.
5. Stephen Boyd, Seung-Jean Kim, Lieven Vandenberghe, and Arash Hassibi. A tutorial on geometric programming. *Optimization and Engineering*, 8(1), 2007.
6. Stephen Boyd. Sequential convex programming. Lecture Notes, 2008.
7. Ernst Moritz Hahn, Holger Hermanns, Björn Wachter, and Lijun Zhang. PARAM: A model checker for parametric Markov models. In *CAV*, volume 6174 of *LNCS*, pages 660–664. Springer, 2010.
8. Ezio Bartocci, Radu Grosu, Panagiotis Katsaros, CR Ramakrishnan, and Scott A Smolka. Model repair for probabilistic systems. In *TACAS*, volume 6605 of *LNCS*, pages 326–340. Springer, 2011.
9. Marta Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *CAV*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.
10. Taolue Chen, Ernst Moritz Hahn, Tingting Han, Marta Kwiatkowska, Hongyang Qu, and Lijun Zhang. Model repair for Markov decision processes. In *TASE*, pages 85–92. IEEE CS, 2013.
11. Gurobi Optimization, Inc. Gurobi optimizer reference manual. http://www.gurobi.com, 2013.
12. Christian Dehnert, Sebastian Junges, Nils Jansen, Florian Corzilius, Matthias Volk, Harold Bruintjes, Joost-Pieter Katoen, and Erika Ábrahám. Prophesy: A probabilistic parameter synthesis tool. In *CAV (1)*, volume 9206 of *LNCS*, pages 214–231. Springer, 2015.
13. PARAM Website, 2015. `http://depend.cs.uni-sb.de/tools/param/`.
14. Karina Valdivia Delgado, Leliane N. de Barros, Daniel B. Dias, and Scott Sanner. Real-time dynamic programming for Markov decision processes with imprecise probabilities. *Artif. Intell.*, 230:192–223, 2016.
15. Antonio Filieri, Giordano Tamburrelli, and Carlo Ghezzi. Supporting self-adaptation via quantitative verification and sensitivity analysis at run time. *IEEE Trans. Software Eng.*, 42(1):75–99, 2016.
16. Joost-Pieter Katoen. The probabilistic model checking landscape. In *IEEE Symposium on Logic In Computer Science (LICS)*. ACM, 2016.
17. Thomas Lipp and Stephen Boyd. Variations and extension of the convex–concave procedure. *Optimization and Engineering*, 17(2):263–287, 2016.
18. Xinyue Shen, Steven Diamond, Yuantao Gu, and Stephen Boyd. Disciplined convex-concave programming. In *CDC*, pages 1009–1014. IEEE, 2016.
19. Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, Ivan Papusha, Hasan A. Poonawala, and Ufuk Topcu. Sequential convex programming for the efficient verification of parametric mdps. In *TACAS (2)*, volume 10206 of *Lecture Notes in Computer Science*, pages 133–150, 2017.

20. Christian Dehnert, Sebastian Junges, Joost-Pieter Katoen, and Matthias Volk. A storm is coming: A modern probabilistic model checker. In *CAV (2)*, volume 10427 of *Lecture Notes in Computer Science*, pages 592–600. Springer, 2017.
21. Lisa Hutschenreiter, Christel Baier, and Joachim Klein. Parametric Markov chains: PCTL complexity and fraction-free Gaussian elimination. In *GandALF*, volume 256 of *EPTCS*, pages 16–30, 2017.
22. Jaehyun Park and Stephen Boyd. General heuristics for nonconvex quadratically constrained quadratic programming. *arXiv preprint arXiv:1703.07870*, 2017.
23. Murat Cubuktepe, Nils Jansen, Sebastian Junges, Joost-Pieter Katoen, and Ufuk Topcu. Synthesis in pmdps: A tale of 1001 parameters. In *ATVA*, volume 11138 of *Lecture Notes in Computer Science*, pages 160–176. Springer, 2018.
24. Sebastian Junges, Nils Jansen, Ralf Wimmer, Tim Quatmann, Leonore Winterer, Joost-Pieter Katoen, and Bernd Becker. Finite-state controllers of pomdps using parameter synthesis. In *UAI*, pages 519–529. AUAI Press, 2018.
25. Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008.
26. Dimitri P. Bertsekas. *Nonlinear Programming*. Athena Scientific Belmont, 1999.
27. Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.
28. Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.