Synthesis of Safe, Optimal and Compact Strategies for Stochastic Hybrid Games

Kim G. Larsen
Department of Computer Science, Aalborg University, DK
kgl@cs.aau.dk

Abstract

Uppaal-Stratego is a recent branch of the verification tool Uppaal allowing for synthesis of safe and optimal strategies for stochastic timed (hybrid) games. We describe newly developed learning methods, allowing for synthesis of significantly better strategies and with much improved convergence behaviour. Also, we describe novel use of decision trees for learning orders-of-magnitude more compact strategy representation. In both cases, the seek for optimality does not compromise safety.

2012 ACM Subject Classification Theory of computation → Timed and hybrid models; Theory of computation → Verification by model checking; Theory of computation → Online learning theory; Theory of computation → Reinforcement learning

Keywords and phrases Timed automata, Stochastic hybrid grame, Symbolic synthesis, Reinforcement learning, Q-learning, M-learning

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2019.2

Category Invited Paper

Funding Kim G. Larsen is partly funded by the ERC-Project LASSO.

1 UPPAAL Stratego

Cyber-physical systems are often safety-critical and hence strong guarantees on their safety are paramount. Besides, resource efficiency and the quality of the delivered service are strong requirements and the behavior needs also to be optimized with respect to these objectives, of course, within the bounds of what is still safe. In order to achieve this, controllers of such systems can be either implemented manually or automatically synthesized. In the former case, due to the complexity of the system, coming up with a controller that is safe is difficult, even more so with the additional optimization requirement. In the latter case, the synthesis may succeed with significantly less effort, though the requirement on both safety and optimality is still a challenge for current synthesis methods. However, due to the size of the systems, the produced controllers may be very complex, hard to understand, implement, modify, or even just output. Indeed, even for moderately sized systems, we can easily end up with gigabytes-long descriptions of their controllers (in the algorithmic context called strategies).

In [5, 6], we introduced Uppaal-Stratego, a branch of Uppaal allowing for synthesis of safe and optimal strategies for stochastic (priced) timed games (STG). The process of using Uppaal-Stratego is depicted in Fig. 1. First, the STG $\mathcal{G}$ is abstracted into a 2-player (non-stochastic) timed game $\mathcal{TG}$, ignoring any stochasticity of the behaviour. Next, the Uppaal-Tiga [3] is used to synthesize a safe strategy σ_{safe} for $\mathcal{TG}$ and the safety specification φ. After that, the safe strategy is applied on $\mathcal{G}$ to obtain $\mathcal{G} \upharpoonright \sigma_{safe}$. It is now possible to perform reinforcement learning on $\mathcal{G} \upharpoonright \sigma_{safe}$ in order to iteratively learn a sub-strategy σ_{fast} that will optimize the expectated value of given quantitative cost, given as a run-based expressions (formally defining a random variable over runs).

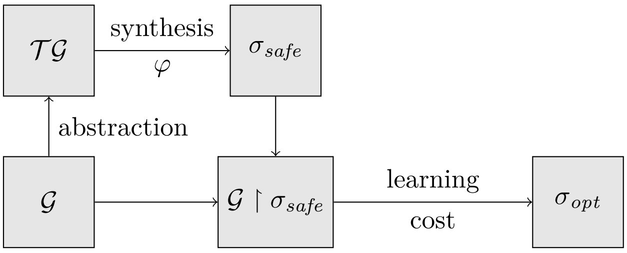

Figure 1 Uppaal-Stratego original workflow.

2 Better and Faster Learning

Though Uppaal-Stratego on a number of practical examples [10, 11, 13, 12, 7] has already demonstrated its ability to learn *near-optimal* strategies, we have recently improved Uppaal-Stratego in a number of ways. Firstly, the run-based reinforcement learning method used in Uppaal-Stratego is a continuous-time extension of the method in [8]. This method is known to be possibly caught in local optima and not necessarily converge towards the overall optimal strategy. In recent work [9], we show that we can significantly improve with respect to this existing method of Uppaal-Stratego both in terms of the quality of the the learned strategy, as well as in obtaining overall improved convergence characteristics as a function of the data size. The new learning methods in [9] are refinement based learning methods for continuous models: one based on Q-learning [15] and one related to Real Time Dynamic Programming [14, 2].

3 Compact Strategies

Aiming at using the synthesized strategies as control programs to be executed on small embedded platforms an important issue is how to encode compactly the synthesized strategies. For this purpose algorithmic methods have been devised to take into account both compositionality [4] as well as partial observability. Although neural network representations of strategies are attractive from a memory foot-print point of view, they may easily destroy the guarantee of safety. In [1], we introduce a new alternative method for learning compact representations of strategies in the form of *decision trees*. These decision trees are much smaller, more understandable, and can easily be exported as code that can be loaded into embedded systems. Despite the size compression and actual differences to the original strategy, we provide guarantees on both safety and optimality of the decision-tree strategy. On the top, we showed how to obtain yet smaller representations, which are still guaranteed safe, but achieve a desired trade off between size and optimality. Finally, we consider two case studies, one of them the cruise control from [13, 12], showing size reductions of two orders of magnitude, and quantify the additional size-performance trade-off.

We summarize the end-to-end work flow of Uppaal-Stratego+ for obtaining a safe, optimal and compact strategy from the model, a safety specification and an optimization query, see Fig. 2. Uppaal-Tiga is used to generate the most-permissive safety strategy σ_{safe} for the given safety specification φ. Now we can either use the standard Uppaal-Stratego workflow to generate the optimal strategy σ_{opt} and then learn a decision tree for this, as depicted on the right path of Fig. 2; or take the new approach following the left branch in Fig. 2. Here, we first learn a DT $\mathcal{T}^{k,p}_{\sigma_{safe}}$ from σ_{safe} using so-called minimum splitting size k and p rounds of safe pruning. This DT is smaller than the one representing σ_{safe} exactly,

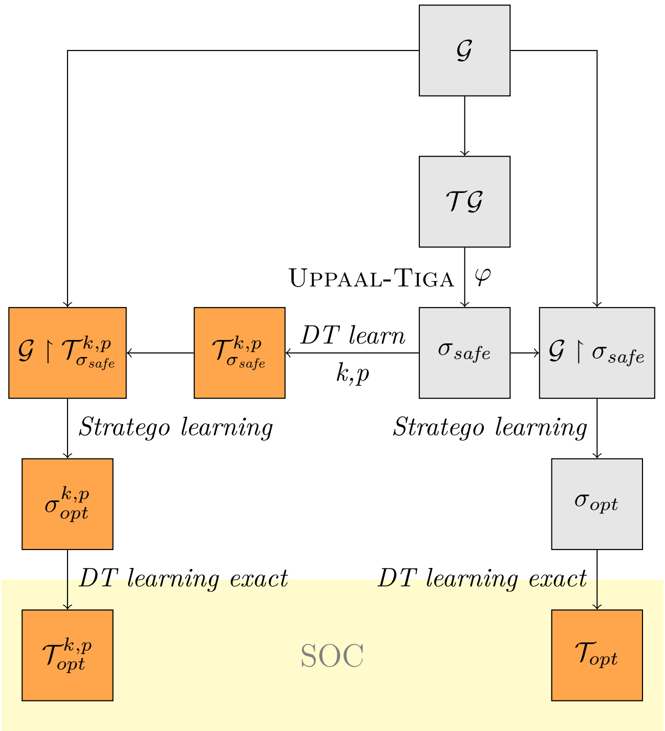

Figure 2 UPPAAL-STRATEGO+workflow. The dark orange nodes are the additions to the original workflow, which now involve DT learning, the yellow-shaded area delimits the desired safe, optimal, and compact strategy representations.

and the described strategy is less permissive. By restricting the game to this strategy and using UPPAAL-STRATEGO to get the optimal strategy, we get a smaller, but less performant strategy $\sigma^{k,p}_{opt}$ that is then output as DT $\mathcal{T}^{k,p}_{opt}$. In both cases, the resulting DT is safe by construction since we allow the DT to predict only pure actions (actions allowed by all configurations in a leaf). We convert these trees into a nested-if-statements-code, which can easily be loaded onto embedded systems.

4 On-line Learning

Finally, on-line methods for strategy synthesis/learning has been successfully applied in diverse domains such as heating systems [11] and intelligent traffic control [7]. The on-line method has the distinct advantages of not needing to store any strategy (as it is constantly computed during operation) but may be too slow to meet response-frequency of a given domain (e.g. in the order of milli-seconds for switched controllers for power electronics or adaptive cruice controls). Thus, we are investigating ways of making the on-line computations more efficient.

References

1 Pranav Ashok, Jan Kretinsky, Kim Guldstrand Larsen, Adrien Le Coent, Jakob Haahr Taankvist, and Maximilian Weininger. SOS: Safe, Optimal and Small Strategies for Stochastic Hybrid Games. In *To appear in Proceedings of QEST*, 2019.

2 Andrew G. Barto, Steven J. Bradtke, and Satinder P. Singh. Learning to Act Using Real-time Dynamic Programming. *Artif. Intell.*, 72(1-2):81–138, January 1995. doi:10.1016/0004-3702(94)00011-O.

3 Gerd Behrmann, Agnès Cougnard, Alexandre David, Emmanuel Fleury, Kim Guldstrand Larsen, and Didier Lime. UPPAAL-Tiga: Time for Playing Games! In Werner Damm and Holger Hermanns, editors, *Computer Aided Verification, 19th International Conference, CAV 2007, Berlin, Germany, July 3-7, 2007, Proceedings*, volume 4590 of *Lecture Notes in Computer Science*, pages 121–125. Springer, 2007. doi:10.1007/978-3-540-73368-3_14.

4 Adrien Le Coënt, Laurent Fribourg, Nicolas Markey, Florian De Vuyst, and Ludovic Chamoin. Compositional synthesis of state-dependent switching control. *Theor. Comput. Sci.*, 750:53–68, 2018. doi:10.1016/j.tcs.2018.01.021.

5 Alexandre David, Peter Gjøl Jensen, Kim Guldstrand Larsen, Axel Legay, Didier Lime, Mathias Grund Sørensen, and Jakob Haahr Taankvist. On Time with Minimal Expected Cost! In Franck Cassez and Jean-François Raskin, editors, *Automated Technology for Verification and Analysis - 12th International Symposium, ATVA 2014, Sydney, NSW, Australia, November 3-7, 2014, Proceedings*, volume 8837 of *Lecture Notes in Computer Science*, pages 129–145. Springer, 2014. doi:10.1007/978-3-319-11936-6_10.

6 Alexandre David, Peter Gjøl Jensen, Kim Guldstrand Larsen, Marius Mikucionis, and Jakob Haahr Taankvist. Uppaal Stratego. In Christel Baier and Cesare Tinelli, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 21st International Conference, TACAS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9035 of *Lecture Notes in Computer Science*, pages 206–211. Springer, 2015. doi:10.1007/978-3-662-46681-0_16.

7 Andreas Berre Eriksen, Harry Lahrmann, and Kim Guldstrand Larsen andJakob Haahr Taankvist. Controlling Signalized Intersections using Machine Learning. In *World Conference on Transport Research*, 2019.

8 David Henriques, João Martins, Paolo Zuliani, André Platzer, and Edmund M. Clarke. Statistical Model Checking for Markov Decision Processes. In *Ninth International Conference on Quantitative Evaluation of Systems, QEST 2012, London, United Kingdom, September 17-20, 2012*, pages 84–93. IEEE Computer Society, 2012. doi:10.1109/QEST.2012.19.

9 Manfred Jaeger, Peter G. Jensen, Kim G. Larsen, Axel Legay, Sean Sedwards, and Jakob H. Taankvist. Teaching Stratego to Play Ball> Optimal Synthesis fo Continuous Space MDPs. In *To appear in Proceedings of ATVA*, 2019.

10 Shyam Lal Karra, Kim Guldstrand Larsen, Florian Lorber, and Jirí Srba. Safe and Time-Optimal Control for Railway Games. In Simon Collart Dutilleul, Thierry Lecomte, and Alexander B. Romanovsky, editors, *Reliability, Safety, and Security of Railway Systems. Modelling, Analysis, Verification, and Certification - Third International Conference, RSSRail 2019, Lille, France, June 4-6, 2019, Proceedings*, volume 11495 of *Lecture Notes in Computer Science*, pages 106–122. Springer, 2019. doi:10.1007/978-3-030-18744-6_7.

11 Kim G. Larsen, Marius Mikucionis, Marco Muñiz, Jirí Srba, and Jakob Haahr Taankvist. Online and Compositional Learning of Controllers with Application to Floor Heating. In Marsha Chechik and Jean-François Raskin, editors, *Tools and Algorithms for the Construction and Analysis of Systems - 22nd International Conference, TACAS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016, Proceedings*, volume 9636 of *Lecture Notes in Computer Science*, pages 244–259. Springer, 2016. doi:10.1007/978-3-662-49674-9_14.

12 Kim Guldstrand Larsen, Adrien Le Coënt, Marius Mikucionis, and Jakob Haahr Taankvist. Guaranteed Control Synthesis for Continuous Systems in Uppaal Tiga. In Roger D. Chamberlain, Walid Taha, and Martin Törngren, editors, *Cyber Physical Systems. Model-Based Design - 8th International Workshop, CyPhy 2018, and 14th International Workshop, WESE 2018, Turin, Italy, October 4-5, 2018, Revised Selected Papers*, volume 11615 of *Lecture Notes in Computer Science*, pages 113–133. Springer, 2018. doi:10.1007/978-3-030-23703-5_6.

13 Kim Guldstrand Larsen, Marius Mikucionis, and Jakob Haahr Taankvist. Safe and Optimal Adaptive Cruise Control. In Roland Meyer, André Platzer, and Heike Wehrheim, editors, *Correct System Design - Symposium in Honor of Ernst-Rüdiger Olderog on the Occasion of His 60th Birthday, Oldenburg, Germany, September 8-9, 2015. Proceedings*, volume 9360 of *Lecture Notes in Computer Science*, pages 260–277. Springer, 2015. doi:10.1007/978-3-319-23506-6_17.

14 Alexander L. Strehl, Lihong Li, and Michael L. Littman. Incremental Model-based Learners With Formal Learning-Time Guarantees. *CoRR*, 2012. arXiv:1206.6870.

15 Christopher John Cornish Hellaby Watkins. *Learning from delayed rewards.* PhD thesis, King's College, Cambridge, 1989.