Multi-Agent Verification and Control with Probabilistic Model Checking

David Parker

Department of Computer Science, University of Oxford, Oxford, UK
david.parker@cs.ox.ac.uk

Abstract. Probabilistic model checking is a technique for formal automated reasoning about software or hardware systems that operate in the context of uncertainty or stochasticity. It builds upon ideas and techniques from a diverse range of fields, from logic, automata and graph theory, to optimisation, numerical methods and control. In recent years, probabilistic model checking has also been extended to integrate ideas from game theory, notably using models such as stochastic games and solution concepts such as equilibria, to formally verify the interaction of multiple rational agents with distinct objectives. This provides a means to reason flexibly about agents acting in either an adversarial or a collaborative fashion, and opens up opportunities to tackle new problems within, for example, artificial intelligence, robotics and autonomous systems. In this paper, we summarise some of the advances in this area, and highlight applications for which they have already been used. We discuss how the strengths of probabilistic model checking apply, or have the potential to apply, to the multi-agent setting and outline some of the key challenges required to make further progress in this field.

1 Introduction

Probabilistic model checking is a fully automated approach for formal reasoning about systems exhibiting uncertain behaviour, arising for example due to faulty hardware, unpredictable operating environments or the use of randomisation. Probabilistic models, such as Markov chains, Markov decision processes (MDPs) or their variants, are systematically explored and analysed in order to establish whether formal specifications given in temporal logic are satisfied. For models such as MDPs, controllers (policies) can be automatically generated to ensure that such specifications are met, or are optimised for.

Probabilistic model checking builds on concepts and techniques from a wide array of other fields. Its roots lie in formal verification, and it relies heavily on the use of logic and automata. Since models typically need to be solved or optimised numerically, it also adopts methods from Markov chains, control theory, optimisation and, increasingly, from various areas of artificial intelligence.

In the this paper, we discuss the integration of probabilistic model checking with ideas from *game theory*, facilitating the verification or control of multi-agent systems in the context of uncertainty. We focus on *stochastic games*, which model the interaction between multiple agents (players) operating in a dynamic, stochastic environment. They were introduced in the 1950s by Shapley [26], generalising the classic model of Markov decision processes [4] to the case of multiple players, and techniques for their solution have been well studied [12].

In the context of formal verification, game-theoretic modelling has a number of natural applications, in particular when an agent interacts with an *adversary* that has opposing goals, for example a defender and an attacker in the context of a computer security protocol or honest and malicious participants in a distributed consensus algorithm. For verification or control problems in stochastic environments, game-based models also underlie methods for *robust* verification, using worst-case assumptions of epistemic model uncertainty. Furthermore, game theory provides tools for controller synthesis in a more *cooperative* setting, for example via the use of *equilibria* representing strategies for collections of players with differing, but not strictly opposing objectives.

Verification problems for stochastic games have been quite extensively studied (see, e.g., [5,27,6]) and in recent years, probabilistic model checking frameworks and tools have been developed (e.g., [8,19,18]) and applied to a variety of problem domains. In the next section, we summarise some of these advances. We then go on to discuss the particular strengths of probabilistic model checking, the ways in which these are applicable to multi-agent models and some of the remaining challenges that exist in the field.

2 Model Checking for Stochastic Games

2.1 Turn-based stochastic games

Stochastic games comprise a set of n players making a sequence of decisions that have stochastic outcomes. The way in which players interact can be modelled in various ways. The simplest is with a *turn-based stochastic game* (TSG). The state space S of the game is partitioned into disjoint sets $S_1 \uplus \cdots \uplus S_n = S$, where states in S_i are controlled by player i. Players choose between actions from a set A (for simplicity, let us assume that all actions are available to be taken in each state) and the dynamics of the model is captured, like for an MDP, by a probabilistic transition function $P : S \times A \times S \to [0,1]$, with $P(s,a,s')$ giving the probability to move to state s' when action a is taken in state s.

A probabilistic model checking framework for TSGs is presented in [8], which proposes the logic rPATL (and its generalisation rPATL*) for specifying *zero-sum* properties of stochastic games, adding probabilistic and reward operators to the well known game logic ATL (alternating temporal logic) [2]. A simple (numerical) query is $\langle\langle ag_1, ag_2 \rangle\rangle \mathtt{P}_{\max=?}[\,\mathtt{F}\ \mathsf{goal}\,]$, which asks for the maximum probability of reaching a set of states $\mathsf{goal} \subseteq S$ that is achievable by a coalition of the players ag_1 and ag_2, assuming that any other players in the game have the directly opposing objective of minimising the probability of this event.

Despite a worse time complexity than MDPs for the core underlying problems (e.g., computing optimal reachability probabilities, for the query above, is in NP ∩ co-NP, rather than PTIME), [8] shows that value iteration (dynamic programming) is in practice an effective and scalable approach. For maximising probabilistic reachability with two players, the values x_s^k defined below converge, as $k \to \infty$, to the required probability for each state s:

$$x_s^k = \begin{cases} 1 & s \in \mathsf{goal} \\ 0 & s \notin \mathsf{goal} \text{ and } k = 0 \\ \max_{a \in A} \sum_{s' \in S} P(s,a,s') \cdot x_{s'}^{k-1} & s \in S_1 \backslash \mathsf{goal} \text{ and } k > 0 \\ \min_{a \in A} \sum_{s' \in S} P(s,a,s') \cdot x_{s'}^{k-1} & s \in S_2 \backslash \mathsf{goal} \text{ and } k > 0 \end{cases}$$

The computation yields optimal strategies for players, which are deterministic (i.e., pick a single action in each state) and memoryless (i.e., do so regardless of history). The model checking algorithm [8] extends to many other temporal operators including a variety of reward (or cost) based measures.

Subsequently, various other aspects of TSG model checking have been explored, including the performance of different game solution techniques [17], the use of interval iteration methods to improve accuracy and convergence [10], trade-offs between multiple objectives [3,7] and the development of symbolic (decision diagram based) implementations to improve scalability [21].

Despite the relative simplicity of TSGs as a modelling formalism, they have been shown to be appropriate for various scenarios in which there is natural turn-based alternation between agents; examples include human-robot control systems [11,15] and self-adaptive software systems interacting with their environment [9].

2.2 Concurrent stochastic games

To provide a more realistic model of the concurrent execution of agents, we can move to the more classic view of player interaction in stochastic games, where players make their decisions simultaneously and independently. To highlight the distinction with the turn-based model variant discussed above, we call these *concurrent stochastic games* (CSGs); the same model is referred to elsewhere as, for example, Markov games or multi-agent Markov decision processes. In a CSG, each player i has a separate set of actions A_i and the probabilistic transition function $P : S \times (A_1 \times \cdots \times A_n) \times S \to [0,1]$ now models the resulting stochastic state update that arises for each possible joint player action.

Probabilistic model checking of CSGs against zero-sum objectives, again using the logic rPATL, is proposed in [19]. Crucially, optimal strategies for players are now randomised, i.e., can choose a probability of selecting each action in each state, however, a value iteration approach can again be adopted. Consider again maximal probabilistic reachability for two players: instead of simply picking the highest value action for one player in each state, a one-shot matrix game Z, indexed over action sets A_1 and A_2, is solved at each state:

$$x_s^k = \begin{cases} 1 & s \in \mathsf{goal} \\ 0 & s \notin \mathsf{goal} \text{ and } k = 0 \\ \mathrm{val}(Z) & s \notin \mathsf{goal} \text{ and } k > 0 \end{cases} \quad \text{where } Z_{a,b} = \sum_{s' \in S} P(s,(a,b),s') \cdot x_{s'}^{k-1}$$

The matrix game Z contains payoff values for player 1 and the value $\mathrm{val}(Z)$ of Z is the optimal achievable value when player 2 minimises. This is solved via a (small) linear programming problem over variables $\{p_a \,|\, a \in A_1\}$ which yields the optimal probabilities p_a for player 1 to pick each action a. Despite the increase in solution complexity with respect to TSGs, results in [19] show the feasibility of building and solving large CSGs that model examples taken from robotics, computer security and communication protocols. These also highlight deficiencies when the same examples are modelled with TSGs.

2.3 Equilibria for stochastic games

Zero-sum objectives, for example specified in rPATL, allow synthesis of optimal controllers in the context of both stochasticity and adversarial behaviour. But there are many instances where agents do not have directly opposing goals. The CSG probabilistic model checking framework has been extended to incorporate non-zero-sum objectives such as Nash equilibria (NE) [19]. Informally, strategies for a set of players with distinct, individual objectives form an NE when there is no benefit to any of them of unilaterally changing their strategy.

It is shown in [19], that by focusing on *social welfare* NE, which also maximise the sum of all players' objectives, value iteration can again be applied. Extending rPATL, we can write for example $\langle\langle ag_1{:}ag_2\rangle\rangle_{\max=?}(\mathtt{P}[\,\mathtt{F}\ \mathsf{goal}_1\,]{+}\mathtt{P}[\,\mathtt{F}\ \mathsf{goal}_2\,])$ to ask for the social welfare NE in which two players maximise the probabilities of reaching distinct sets of state goal_1 and goal_2. Value iteration becomes:

$$x_s^k = \begin{cases} (1,1) & s \in \mathsf{goal}_1 \cap \mathsf{goal}_2 \\ (1, P^{\max}_{s,\mathsf{goal}_2}) & s \in \mathsf{goal}_1 \backslash \mathsf{goal}_2 \\ (P^{\max}_{s,\mathsf{goal}_1}, 1) & s \in \mathsf{goal}_2 \backslash \mathsf{goal}_1 \\ (0,0) & s \notin (\mathsf{goal}_1 \cup \mathsf{goal}_2) \text{ and } k = 0 \\ \mathrm{val}(Z^1, Z^2) & s \notin (\mathsf{goal}_1 \cup \mathsf{goal}_2) \text{ and } k > 0 \end{cases}$$

$$\text{where } Z^i_{a,b} = \sum_{s' \in S} P(s,(a,b),s') \cdot x^{k-1}_{s'}(i),$$

$\mathrm{val}(Z^1, Z^2)$ is the value of a *bimatrix game* and $P^{\max}_{s,\mathsf{goal}_i}$ is the maximum probability of reaching goal_i from state s, which can be computed independently by treating the stochastic game as an MDP. The value of the (one-shot) game defined by payoff matrices Z^i for player i is a (social welfare) NE, computed in [19] using an approach called labelled polytopes [22] and a reduction to SMT. Optimal strategies (in fact, ϵ-optimal strategies) can be extracted. They are, as for zero-sum CSGs, randomised but also require memory.

The move towards concurrent decision making over distinct objectives opens up a variety of interesting directions for exploration. Equilibria-based model checking of CSG is extended in several ways in [20]. Firstly, *correlated* equilibria allow players to coordinate through the use of a (probabilistic) public signal, which then dictates their individual strategies. These are shown to be cheaper to compute and potentially more equitable in the sense that they improve joint outcomes. Secondly, *social fairness* is presented as an alternative optimality criterion to social welfare, which minimises the *difference* between players' objectives, something that is ignored by by the latter criterion.

3 Opportunities and Challenges

Probabilistic model checking is a flexible technique, which already applies to many different types of stochastic models and temporal logic specifications. On the one hand, the thread of research described above represents a further evolution of the approach towards a new class of models and solution methods. On the other, it represents an opportunity to apply the strengths of probabilistic model checking to a variety of problem domains in which multi-agent approaches to modelling are applicable and where guarantees on safety or reliability may be essential; examples include multi-robot coordination, warehouse logistics, autonomous vehicles or robotic swarms. We now discuss some of the key benefits of probabilistic model checking and their relevance to the multi-agent setting. We also summarise some of the challenges that arise as a result.

Temporal logic. Key to the success of model checking based techniques is the ability to precisely and unambiguously specify desired system properties in a formal language that is expressive enough to be practically useful, but constrained enough that verification or control problems are practical.

Temporal logics such as rPATL and its extensions show that it is feasible to combine quantitative aspects (probability, costs, rewards) with reasoning about the strategies and objectives of multiple agents, for both zero-sum optimality and equilibria of various types. This combines naturally with the specification of temporal behaviour using logics such as LTL, from simple reachability or reach-avoid goals, to more complex sequences of events and long-run specifications. The latter have been of increasing interest in, for example, task specification in robotics [16] or reinforcement learning [14]. Another key benefit here is the continual advances in translations from such logics to automata, facilitating algorithmic analysis. From a multi-agent perspective, specifically, challenges include more expressive reasoning about dependencies between strategies or epistemic aspects, where logic extensions exist but model checking is challenging.

Tool support and modelling languages. The main functionality for model checking of stochastic games described in Section 2 is implemented within the PRISM-games tool [18], which has been developed over the past 10 years. However, interest in verification for this class of models is growing and, for the simpler model variant of TSGs, support now exists in multiple actively developed probabilistic model checking tools [25,24,17,13].

Currently, these tools share a common formalism for specifying models, namely PRISM-games's extension to stochastic games of the widely used PRISM modelling language. Although relatively simple from a programming language perspective, this has proved to be expressive enough for modelling across a large range of application domains. Key modelling features include the ability to describe systems as the parallel composition of multiple, interacting (sometimes duplicated) components and the use of parameters for easy reconfiguration of models. It also provides a common language for many different types of probabilistic models through various features that can be combined, e.g., clocks (for *real-time* modelling), observations (for *partially observable* models) and model uncertainty such as transition probability intervals (for *epistemic uncertainty*).

Component-based modelling is of particular benefit for the multi-agent setting, but challenges remain as the modelling language evolves. Examples include dealing with the subtleties that arise regarding how components communicate and synchronise, particularly under partial observability, and the specification of particular strategies for some agents.

Exhaustive analysis. Traditionally, a strength of model checking based techniques is their focus on an exhaustive model analysis, allowing them to identify (or prove the absence of) corner cases representing erroneous or anomalous behaviour. In the stochastic setting this remains true, in particular for models that combine probabilistic and nondeterministic behaviour. The subtle interaction between these aspects can be difficult to reason about without automated tools, and exhaustive approaches can in these cases be preferable to more approximate model solution methods, such as those based on simulation.

Adding multiple players only strengthens the case for these techniques. For example, [8] identifies weaknesses in a distributed, randomised energy management protocol arising when some participants behave selfishly; a simple incentive scheme is then shown to help alleviate this issue. Multi-agent models allow a combination of *control* and *verification*, for example synthesising a controller (strategy) for one player, or coalition of players, which can be verified to perform robustly in the context of adversarial behaviour by other players.

A natural direction is to then deploy verification to controllers synthesised by more widely applicable and more scalable approaches such as multi-agent reinforcement learning [1]. This brings challenges in terms of, for example, extending probabilistic model checking to continuous state spaces, and tighter integration with machine learning methods. Progress in this direction includes the extension of CSGs to a *neuro-symbolic* setting [29,28], which incorporates neural networks for specific model components, e.g. for perception tasks.

Further challenges. In addition to those highlighted above, numerous other challenges exist in the field. One perennial issue with model checking approaches is their *scalability* to large systems. Symbolic approaches, e.g., based on decision diagrams, have proved to be valuable for probabilistic model checking, and also extended to TSGs [21]. However it is unclear how to adapt these to CSGs: while value iteration is often amenable to a symbolic implementation, methods such as linear programming or bimatrix game solution are typically not.

The use of modelling formalisms like the PRISM-games language should also encourage the development of *compositional* analysis techniques, such as counter abstraction [23] or assume-guarantee methods, but progress remains limited in these directions within probabilistic model checking. On a related note, while human-created models naturally exhibit high-level structure, strategies synthesised by model checking tools typically do not. This hinders comprehension and explainability, which becomes more important when strategies, as here, are more complex due to randomisation, memory and distribution across agents.

There are also major *algorithmic* challenges which arise as the techniques are applied to new problems. For example, there have been steady advances in verification techniques for *partially observable* MDPs, but much less work on this topic for stochastic games. Finally, there are potential benefits from further exploration of ideas from game theory, e.g., other equilibria, such as Stackelberg (with applications, for instance, to security or automotive settings) or the inclusion of epistemic aspects into logics and model checking algorithms.

Acknowledgements. This work was funded in part by the ERC under the European Union's Horizon 2020 research and innovation programme (FUN2MODEL, grant agreement No. 834115).

References

1. Albrecht, S.V., Christianos, F., Schäfer, L.: Multi-Agent Reinforcement Learning: Foundations and Modern Approaches. MIT Press (2023)
2. Alur, R., Henzinger, T., Kupferman, O.: Alternating-time temporal logic. J. ACM **49**(5), 672–713 (2002)
3. Basset, N., Kwiatkowska, M., Wiltsche, C.: Compositional strategy synthesis for stochastic games with multiple objectives. IC **261**(3), 536–587 (2018)
4. Bellman, R.: Dynamic Programming. Princeton University Press (1957)
5. Chatterjee, K.: Stochastic ω-Regular Games. Ph.D. thesis, University of California at Berkeley (2007)
6. Chatterjee, K., Henzinger, T.: A survey of stochastic ω-regular games. J. CSS **78**(2), 394–413 (Mar 2012)
7. Chatterjee, K., Katoen, J.P., Weininger, M., Winkler, T.: Stochastic games with lexicographic reachability-safety objectives. In: Proc. 32nd International Conference on Computer Aided Verification (CAV'20). LNCS, vol. 12225, pp. 398–420. Springer (2020)
8. Chen, T., Forejt, V., Kwiatkowska, M., Parker, D., Simaitis, A.: Automatic verification of competitive stochastic systems. FMSD **43**(1), 61–92 (2013)
9. Cámara, J., Garlan, D., Schmerl, B., Pandey, A.: Optimal planning for architecture-based self-adaptation via model checking of stochastic games. In: Proc. SAC'15. pp. 428–435. ACM (2015)
10. Eisentraut, J., Kelmendi, E., Kretínský, J., Weininger, M.: Value iteration for simple stochastic games: Stopping criterion and learning algorithm. Information and Computation **285**(Part), 104886 (2022)
11. Feng, L., Wiltsche, C., Humphrey, L., Topcu, U.: Controller synthesis for autonomous systems interacting with human operators. In: Proc. ICCPS'15. pp. 70–79. ACM (2015)
12. Filar, J., Vrieze, K.: Competitive Markov Decision Processes. Springer, New York, NY, USA (1996)

13. Fu, C., Hahn, E.M., Li, Y., Schewe, S., Sun, M., Turrini, A., Zhang, L.: EPMC gets knowledge in multi-agent systems. In: Proc. VMCAI'22. LNCS, vol. 13182, pp. 93–107. Springer (2022)
14. Hammond, L., Abate, A., Gutierrez, J., Wooldridge, M.J.: Multi-agent reinforcement learning with temporal logic specifications. In: Proc. 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS'21). pp. 583–592. ACM (2021)
15. Junges, S., Jansen, N., Katoen, J.P., Topcu, U., Zhang, R.: Model checking for safe navigation among humans. In: Proc. QEST'18. LNCS, vol. 11024, pp. 207–222. Springer (2018)
16. Kress-Gazit, H., Fainekos, G.E., Pappas, G.J.: Temporal logic-based reactive mission and motion planning. IEEE Transactions on Robotics **25**(6), 1370–1381 (2009)
17. Kretínský, J., Ramneantu, E., Slivinskiy, A., Weininger, M.: Comparison of algorithms for simple stochastic games. Information and Computation **289**(Part), 104885 (2022)
18. Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: PRISM-games 3.0: stochastic game verification with concurrency, equilibria and time. In: Proc. CAV'20. LNCS, vol. 12225, pp. 475–487. Springer (2020), prismmodelchecker.org/games/
19. Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: Automatic verification of concurrent stochastic systems. Formal Methods in System Design **58**, 188–250 (2021)
20. Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: Correlated equilibria and fairness in concurrent stochastic games. In: Proc. 28th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'22). LNCS, vol. 13244, p. 60–78. Springer (2022)
21. Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: Symbolic verification and strategy synthesis for turn-based stochastic games. In: Principles of Systems Design: Essays Dedicated to Thomas A. Henzinger on the Occasion of His 60th Birthday. LNCS, vol. 13660. Springer (2022)
22. Lemke, C., J. Howson, J.: Equilibrium points of bimatrix games. J. Society for Industrial and Applied Mathematics **12**(2), 413–423 (1964)
23. Lomuscio, A., Pirovano, E.: A counter abstraction technique for verifying properties of probabilistic swarm systems. Artificial Intelligence **305**, 103666 (2022)
24. Meggendorfer, T.: PET - A partial exploration tool for probabilistic verification. In: Proc. 20th International Symposium on Automated Technology for Verification and Analysis (ATVA'22). LNCS, vol. 13505, pp. 320–326. Springer (2022)
25. Pranger, S., Könighofer, B., Posch, L., Bloem, R.: TEMPEST - synthesis tool for reactive systems and shields in probabilistic environments. In: ATVA. Lecture Notes in Computer Science, vol. 12971, pp. 222–228. Springer (2021)
26. Shapley, L.: Stochastic games. PNAS **39**, 1095–1100 (1953)
27. Ummels, M.: Stochastic Multiplayer Games: Theory and Algorithms. Ph.D. thesis, RWTH Aachen University (2010)
28. Yan, R., Santos, G., Norman, G., Parker, D., Kwiatkowska, M.: Strategy synthesis for zero-sum neuro-symbolic concurrent stochastic games. arXiv.2202.06255 (2022)
29. Yan, R., Santos, G., Duan, X., Parker, D., Kwiatkowska, M.: Finite-horizon equilibria for neuro-symbolic concurrent stochastic games. In: Proc. 38th Conference on Uncertainty in Artificial Intelligence (UAI'22). AUAI Press (2022)