European Journal of Control ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

European Journal of Control

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

Quantitative verification and strategy synthesis for stochastic games ☆

Mária Svoreňová, Marta Kwiatkowska *

Department of Computer Science, University of Oxford, Oxford, UK

ARTICLE INFO

Article history:
Received 19 January 2016
Received in revised form
22 April 2016
Accepted 23 April 2016
Recommended by Alessandro Astolfi

Keywords:
Stochastic games
Controller synthesis
Quantitative verification
Temporal logic
Multi-objective properties

ABSTRACT

Design and control of computer systems that operate in uncertain, competitive or adversarial, environments can be facilitated by formal modelling and analysis. In this paper, we focus on analysis of complex computer systems modelled as turn-based $2\frac{1}{2}$-player games, or stochastic games for short, that are able to express both stochastic and non-stochastic uncertainties. We offer a systematic overview of the body of knowledge and algorithmic techniques for verification and strategy synthesis for stochastic games with respect to a broad class of quantitative properties expressible in temporal logic. These include probabilistic linear-time properties, expected total, discounted and average reward properties, and their branching-time extensions and multi-objective combinations. To demonstrate applicability of the framework as well as its practical implementation in a tool called PRISM-games, we describe several case studies that rely on analysis of stochastic games, from areas such as robotics, and networked and distributed systems.

1. Introduction

Since the dawn of the information age, correctness and safety of computer systems have been central to their design and analysis. Computer systems typically operate in uncertain environments. The uncertainty can be stochastic due to, *e.g.,* unreliable communication media, faulty components or simply due to the use of randomisation. Moreover, if components that cannot be controlled are present in the environment, their adversarial or competitive behaviour results in additional, non-stochastic uncertainty. Examples of such systems appear in many domains, from robotics and autonomous transport, to security, networked and distributed systems, and power management.

It is natural to view such complex systems as games between the controllable computer system and its (uncontrollable) environment. In this work, we present a comprehensive overview of techniques used in verification and controller (also called a strategy) synthesis for systems modelled as $2\frac{1}{2}$-player games, or stochastic games for short. In every step of a stochastic game, the two players, Player 1 and Player 2, choose their moves and, based on their choices, the next state of the game is determined, possibly in a probabilistic fashion. Controller synthesis can then be viewed as finding a winning strategy for Player 1, where Player 2 may play adversarially. Stochastic games have been employed, for example, to support decision making and synthesise controllers for aircraft power distribution [6], sensor network management in renewable energy production plants [73], in human-in-the-loop UAV planning [46], and autonomous driving in the presence of hazards such as pedestrians [81,33]. They arise naturally in the context of security and defence, where they have been used in patrol planning [82], port defence [72], infrastructure protection [18], to generate countermeasures for DNS bandwidth attacks [43] and to analyse complex attack-defence scenarios in RFID goods management system [4]. Through the use of abstraction and discretisation, high-level control of hybrid and continuous systems can also be addressed.

Stochastic games were first introduced by Shapley in 1953 [71]. Various classes and modifications of these games have been extensively studied since then and surveyed in, e.g., [47,61,22,48,54,21,62]. In this survey, we focus on turn-based games, where players choose their moves in turns rather than concurrently as in [71,47,61]. More specifically, we restrict our attention to turn-based, finite, complete-observation, stochastic, discrete-time, zero-sum games. We also consider a generalisation of these games to multiple players. Compared to existing surveys of these games such as [22], the distinguishing feature of our survey is a comprehensive coverage of algorithms for temporal logic properties, including reward and multi-objective properties not covered in [22], and an illustration of their practical application on a tutorial-style example. Other surveys typically focus on related classes of games, to mention concurrent games [71,47,61], or only on a subclass of properties, *e.g.,* single-objective [22].

☆This work was supported by ERC Advanced Investigators Grant VERIWARE and EPSRC Mobile Autonomy Programme Grant EP/M019918/1.

* Corresponding author.
E-mail addresses: maria.svorenova@cs.ox.ac.uk (M. Svoreňová), marta.kwiatkowska@cs.ox.ac.uk (M. Kwiatkowska).

http://dx.doi.org/10.1016/j.ejcon.2016.04.009
0947-3580/

We study various classes of properties of stochastic games expressible in temporal logic. First, we consider quantitative probabilistic properties over linear time, expressed as formulas of probabilistic linear temporal logic. Examples of such properties include 'the maximum probability of the airbag failing to deploy within 0.02 s is at most 10^{-6}', or 'the minimum probability of the car to reach its destination without colliding with pedestrians, while obeying traffic rules, is at least $1-10^{-10}$'. Next, properties reasoning about rewards associated with states of the game are introduced. Namely, we consider the expected total and discounted cumulative reward as well as long-run average reward. These can be used to state properties such as 'the minimum expected profit that the investor can guarantee within a year is at least 1000', or 'the expected number of requests served per time unit in the network is at least 5'. Finally, we allow to combine the above linear-time and reward properties to express requirements over branching time, thus allowing analysis of properties such as 'the probability that the network recovers from a bad decision to a state from which a consensus can be reached with probability at least 0.9 is at least 0.95'.

Given a stochastic game and a property, verification and strategy synthesis problems, respectively, focus on the existence and construction of a strategy for Player 1 that guarantees satisfaction of the property against all strategies of Player 2. In this work, we discuss general findings for the two problems and overview the existing algorithmic solutions for various classes of properties. The solutions typically rely on a reduction to simpler games or properties, and the computation of optimal values and strategies, for which a value iteration algorithm is typically utilised. In the multi-player case, a coalition of players aims to cooperatively enforce a property. Intuitively, multi-player stochastic games can be seen as stochastic games with two players, where the coalition acts as Player 1 and the remaining set of players as Player 2. Finally, we analyse stochastic games with respect to multi-objective properties that require simultaneous satisfaction of multiple linear-time and reward properties. Here, the properties can be conflicting and the techniques reduce to the computation of an ε-approximation of the Pareto set of optimal trade-offs between the individual properties.

While a number of software tools exist with partial support for stochastic games, see Section 5 for a summary, they only allow a subclass of such games, *e.g.,* with one player or without stochasticity, or perform analysis of stochastic games against single-objective properties only. On the other hand, most of the overviewed algorithms have been implemented within the tool called PRISM-games [36,55] for modelling, verification, synthesis and simulation of stochastic games, an extension of the PRISM model checker [56]. We briefly overview the features and functionality of the tool and offer a number of case studies, where complex computer systems have been modelled as stochastic games and their properties analysed in PRISM-games.

Contributions of this paper can be summarised as follows:

- we present a comprehensive framework for analysis of stochastic games, focusing on high-level temporal logic specifications;
- we overview the existing body of knowledge and algorithmic solutions for verification and strategy synthesis problems for stochastic games, and identify open problems;
- we offer a list of case studies of control systems that are modelled and analysed through stochastic games.

The remainder of this paper is organised as follows. In Section 2, we introduce stochastic games and define a specification language for linear-time and reward properties. In Section 3, we formulate the verification and strategy synthesis problems for single-objective properties, present general findings for the problems, as well as algorithmic solutions, and their extensions to branching-time properties and multi-player games. Multi-objective combinations of properties are then discussed in Section 4. We overview existing tools for games in Section 5 and briefly describe the functionality of PRISM-games, which currently provides the most comprehensive support for stochastic games. Finally, we list several case studies that rely on stochastic games in Section 6. We finish with concluding remarks in Section 7. To demonstrate the framework, an illustrative example modelled and analysed in PRISM-games is used throughout the paper.

2. Preliminaries

2.1. Notation

We use $\mathcal{D}(X)$ to denote the set of all probability distributions over a set X. Given a finite or infinite sequence λ of elements of X, we use λ_i to denote its ith element for $i \geq 0$. For a finite sequence $\lambda = x_0 x_1 \ldots x_k$ of elements of X, we use $|\lambda| = k+1$ to denote the length of the sequence and $\text{last}(\lambda) = x_k$ denotes its last element.

2.2. Stochastic games

Definition 1 (*Stochastic game*). A turn-based $2\frac{1}{2}$-player game or simply a stochastic game is a tuple $\mathcal{G} = (S, (S_1, S_2, S_p), \Delta)$, where S is a finite set of states partitioned into sets S_1, S_2 and S_p of Player 1, Player 2 and probabilistic states, respectively, and $\Delta : S \times S \to [0,1]$ is a probabilistic transition function such that, for states $s \in S_1 \cup S_2$, it holds that $\Delta(s,s') \in \{0,1\}$ for every $s' \in S$, where we assume that $\Delta(s,s') = 1$ for at least one $s' \in S$, and for states $s \in S_p$, it holds $\sum_{s' \in S} \Delta(s,s') = 1$.

Intuitively, the game is played as follows. The state of the game is always determined uniquely and, in every step, the next state is chosen according to the transition function. When the current state of the game is a Player 1 state, *i.e.,* $s \in S_1$, then Player 1 chooses the next state $s' \in S$ such that $\Delta(s,s') = 1$, and similarly for Player 2. When the current state is probabilistic, *i.e.,* $s \in S_p$, the next state of the game is sampled according to the distribution $\Delta(s,\cdot)$.

Formally, a *path* of a game $\mathcal{G}$ is an infinite sequence $\lambda = s_0 s_1 \ldots$ such that $\Delta(s_i, s_{i+1}) > 0$ for all $i \geq 0$. A finite path of $\mathcal{G}$ is a finite prefix of a path. We use $\text{Path}_{\mathcal{G},s}$ to denote the set of all paths originating in a state $s \in S$ and $\text{Path}_{\mathcal{G}} = \cup_{s \in S} \text{Path}_{\mathcal{G},s}$. The sets $\text{FPath}_{\mathcal{G},s}$, $\text{FPath}_{\mathcal{G}}$ of finite paths are defined analogously. Given two states $s, s' \in S$, we say that s' is *reachable* from s if and only if there exists a finite path λ such that $\lambda_0 = s$ and $\text{last}(\lambda) = s'$.

Definition 2 (*Labelling function*). Given a finite set of atomic propositions AP, a labelling function $L : S \to 2^{\text{AP}}$ assigns to each state $s \in S$ of the game a set of atomic propositions that hold true in s.

Definition 3 (*Reward structure*). Given a game $\mathcal{G} = (S, (S_1, S_2, S_p), \Delta)$, a reward structure for $\mathcal{G}$ is a function $r : S \to \mathbb{R}_{\geq 0}$ or $r : S \to \mathbb{R}_{\leq 0}$.

While the term reward intuitively suggests that the goal will be to maximise functions over these values, we use a reward structure as a general value assignment and consider minimisation problems as well. In such a case, the values are often referred to as costs rather than rewards. By inverting the signature of all rewards, the resulting function is again a reward structure, and this will allow us to translate minimisation problems to maximisation. Note that, unless stated otherwise, in this work we do not consider reward structures that assign both negative and positive values.

Please cite this article as: M. Svoreňová, M. Kwiatkowska, Quantitative verification and strategy synthesis for stochastic games, European Journal of Control (2016), http://dx.doi.org/10.1016/j.ejcon.2016.04.009