





Playing Against Fair Adversaries in Stochastic Games with Total Rewards

Pablo F. Castro^{1,3} , Pedro R. D'Argenio^{2,3,4} , Ramiro Demasi^{2,3} ,
and Luciano Putruel^{1,3} 

¹ Departamento de Computación, FCEFQyN, Universidad Nacional de Río Cuarto,
Río Cuarto, Argentina

² FAMAF, Universidad Nacional de Córdoba, Córdoba, Argentina

³ Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET),
Buenos Aires, Argentina

`pcastro@dc.exa.unrc.edu.ar`

⁴ Saarland University, Saarland Informatics Campus, Saarbrücken, Germany

Abstract. We investigate zero-sum turn-based two-player stochastic games in which the objective of one player is to maximize the amount of rewards obtained during a play, while the other aims at minimizing it. We focus on games in which the minimizer plays in a fair way. We believe that these kinds of games enjoy interesting applications in software verification, where the maximizer plays the role of a system intending to maximize the number of “milestones” achieved, and the minimizer represents the behavior of some uncooperative but yet fair environment. Normally, to study total reward properties, games are requested to be stopping (i.e., they reach a terminal state with probability 1). We relax the property to request that the game is stopping only under a fair minimizing player. We prove that these games are determined, i.e., each state of the game has a value defined. Furthermore, we show that both players have memoryless and deterministic optimal strategies, and the game value can be computed by approximating the greatest-fixed point of a set of functional equations. We implemented our approach in a prototype tool, and evaluated it on an illustrating example and an Unmanned Aerial Vehicle case study.

1 Introduction

Game theory [25] admits an elegant and profound mathematical theory. In the last decades, it has received widespread attention from computer scientists because it has important applications to software synthesis and verification. The analogy is appealing, the operation of a system under an uncooperative environment (faulty hardware, malicious agents, unreliable communication channels, etc.) can be modeled as a game between two players (the system and the environment), in which the system tries to fulfill certain goals, whereas the environment tries to prevent this from happening. This view is particularly useful for

This work was supported by ANPCyT PICT-2017-3894 (RAFTSsys), ANPCyT PICT 2019-03134, SeCyT-UNC 33620180100354CB (ARES), and EU Grant agreement ID: 101008233 (MISSION).

© The Author(s) 2022

S. Shoham and Y. VizeI (Eds.): CAV 2022, LNCS 13372, pp. 48–69, 2022.

https://doi.org/10.1007/978-3-031-13188-2_3

controller synthesis, i.e., to automatically generate decision-making policies from high-level specifications. Thus, synthesizing a controller consists of computing optimal strategies for a given game.

In this paper we focus on zero-sum, perfect-information, two-player, turn-based stochastic games with (non-negative) rewards [18]. Intuitively, these games are played in a graph by two players who move a token in turns. Some vertices are probabilistic, in the sense that, if a token is in a probabilistic vertex, then the next vertex is randomly selected. Furthermore, the players select their moves using strategies. Associated with each vertex there is a reward (which, in this paper, is taken to be non-negative). The goal of Player 1 is to maximize the expected amount of collected rewards during the game, whereas Player 2 aims at minimizing this value. This is what [28] calls *total reward objective*. These kinds of games have been shown useful to reason about several classes of systems such as autonomous vehicles, fault-tolerant systems, communication protocols, energy production plants, etc. Particularly, in this paper we consider those games in which one of the players employs fair strategies.

Fairness restrictions, understood as fair resolutions of non-determinism of actions, play an important role in software verification and controller synthesis. Especially, fairness assumptions over environments make possible the verification of liveness properties on open systems. Several authors have indicated the need for fairness assumptions over the environment in the controller synthesis approach, e.g., [2, 16]. As a simple example consider an autonomous vehicle that needs to traverse a field where moving objects may interfere in its path. Though the precise behavior of the objects may be unknown, it is reasonable to assume that they will not continuously obstruct the vehicle attempts to avoid them. In this sense, while stochastic behavior may be a consequence of the vehicle faults, we can only assume a fair behavior of the surrounding moving objects. In this work, we consider stochastic games in which one of the players (the one playing the environment) is assumed to play only with strong fair strategies.

In order to guarantee that the expected value of accumulated rewards is well defined in (perhaps infinite) plays, some kind of stopping criteria is needed. A common way to do this is to force the strategies to decide to stop with some positive probability in every decision. This corresponds to the so-called discounted stochastic games [18, 27], and has the implications that the collected rewards become less important as the game progresses (the “importance reduction” is given by the discount factor). Alternatively, one may be interested in knowing the expected *total* reward, that is, the expected accumulated reward *without* any loss of it as time progresses. For this value to be well defined, the game itself needs to be stopping. That is, no matter the strategies played by the players, the probability of reaching a terminal state needs to be 1 [13, 18]. We focus on this last type of game. However, we study here games that may not be stopping in general (i.e., for every strategy), but instead, require that they become stopping only when the minimizer plays in a fair way. We use a notion of (almost-sure) strong fairness, mostly following the ideas introduced in [7] for Markov decision processes. We show that these kinds of games are determined, i.e., each state of the game has a value defined. Furthermore, we show that memoryless and deterministic optimal

strategies exist for both players. Moreover, the value of the game can be calculated via the greatest fixed point of the corresponding functionals. It is important to remark that most of the properties discussed in this paper hold when the fairness assumptions are made over the minimizer. Similar properties may not hold if the role of players is changed. However, these conditions encompass a large class of scenarios, where the system intends to maximize the total collected reward and the environment has the opposite objective.

In summary, the contributions of this paper are the following: (1) we introduce the notion of stopping under fairness stochastic game, a generalization of stopping game that takes into account fair environments; (2) we prove that it can be decided in polynomial time whether a game is stopping under fairness; (3) we show that these kinds of games are determined and both players possess optimal stationary strategies, which can be computed using Bellman equations; and (4) we implemented these ideas in a prototype tool embedded in the PRISM-games toolset [22], which we used to evaluate the viability of our approach through illustrative case studies.

The paper is structured as follows. Section 2 introduces an illustrating example to motivate the use of having fairness restrictions over the minimizer. Section 3 fixes terminology and introduces background concepts. In Sect. 4 we describe a polynomial procedure to check whether a game stops under fairness assumptions, we also prove that determinacy is preserved in these games as well as the existence of (memoryless and deterministic) optimal strategies. Experimental results are described in Sect. 5. Finally, Sects. 6 and 7 discuss related work and draw some conclusions, respectively.

2 Roborta vs. the Fair Light (A Motivating Example)

Consider the following scenario. Roborta the robot is navigating a grid of 4×4 cells. Roborta's moves respond to a traffic light: if the light is yellow, she must move sideways (at a border cell, Roborta is allowed to wrap around to the other side); if the light is green she ought to move forward; if the light is red, she cannot perform any movement; finally, if the light is off, Roborta is free to move either sideways or forward. The light and Roborta change their states in turns. In addition, a (non-negative) reward is associated with each cell of the grid. Also, some cells restrict the sideways movement to only one direction. Moreover, we consider possible failures on the behavior of the robot and the light. If Roborta fails, she loses her turn to move. If the light fails, it turns itself off. The failures occur with a given probability and are not permanent (they only affect the current play). The goal of Roborta is to collect as many rewards as possible. In opposition, the light aims at minimizing this value.

The specification of this game is captured in Fig. 1 (using PRISM-like notation [23]). In this model, `WIDTH` and `LENGTH` are constants defining the dimension of the grid. `MOVES` is a two-dimensional array modeling the possible sideways movements in the grid (0 allows the robot to move only to the left, 1, to either side, and 2, only to the right). The light plays when it is red (`light=0`) and it

```

module Roberta_vs_the_light
col : [0..WIDTH] init 0;
row : [0..LENGTH] init 0;
light : [0..3] init 0; // current light color
                        // 0: red (light's turn)
                        // 1: yellow (Roberta moves sideways)
                        // 2: green (Roberta moves forward)
                        // 3: off (light fails, any move)

// light moves
[l_y] (light=0) -> (1-Q) : (light'=1) + Q : (light'=3);
[l_g] (light=0) -> (1-Q) : (light'=2) + Q : (light'=3);
// Roberta moves
[r_l] ((light=1) | (light=3) & (MOVES[col,row] <= 1)
      -> (1-P) : (light'=0) & (col'=(col-1)%WIDTH) +
      P : (light'=0) ;
[r_r] ((light=1) | (light=3) & (MOVES[col,row] >= 1)
      -> (1-P) : (light'=0) & (col'=(col+1)%WIDTH) +
      P : (light'=0) ;
[r_f] ((light=2) | (light=3) & (row < LENGTH)
      -> (1-P) : (light'=0) & (row'=row+1) +
      P : (light'=0) ;
endmodule
    
```

Fig. 1. Model for the Game

(`r_r`), provided the grid allows the movements. If the light is green (`light=2`) or off (`light=3`), she can choose to move forward (notice that if `light=2` this is the only possible move). Like the light, each of Roberta's choices has a failure probability of P , in which case, she does not move and only passes the turn to the light (by setting `light'=0`). For completeness, we mention that the rewards are stored in a secondary matrix which is not shown in Fig. 1.

Figure 2 shows the assignment of rewards to each cell of the 4×4 grid as well as the sideways movement restrictions (shown on the bottom-right of each cell with white arrows). The game starts at the cell $(0, 0)$ and it stops when Roberta escapes through the end of the grid (i.e., `row = LENGTH`).

A possible scenario in this game is as follows. Roberta starts in cell $(0, 0)$ and, in an attempt to minimize the rewards accumulated by the robot, the environment switches the yellow light on. For the sake of simplicity, we assume no failures on the light, i.e., $Q = 0$. Notice that, if the environment plays always in this way (signaling a yellow light), then Roberta does not collect rewards (since all rewards in the first row are 0) but also she will never reach the goal and the game never stops. This scenario occurs when the light plays in an unfair way, i.e., an action (the one that turns the green light on) is enabled infinitely often, but it is not executed infinitely often. Assuming fairness over the environment, we can ensure that a green light will be eventually switched on, allowing the robot to move forward.

For the case in which $Q = 0$, the best strategy for Roberta when the light is yellow is shown in black arrows on the top-right of each cells with no movement

can choose whether to turn on the yellow light (transition labelled with `l_y`) or green (transition labelled `l_g`). Notice that with any choice, the light may fail with probability Q , in which case it turns itself off (`light'=3`). If the light is not red, then it is Roberta's turn to play. If the light is yellow (`light=1`) or off (`light=3`), Roberta can chose whether to move left (`r_l`) or right

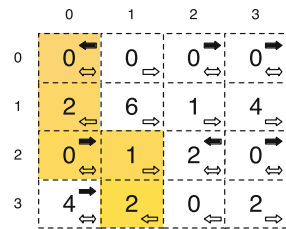


Fig. 2. A robot on a 4×4 grid

restrictions (restricting cells provide only one choice). As a result, when both players play their optimal strategies, the path taken by Roborta to achieve the goal can be observed in the yellow-highlighted portion of the grid in Fig. 2. In Sect. 5, we evaluate this problem experimentally with different configurations of the game.

3 Preliminaries

We introduce some basic definitions and results on stochastic games that will be necessary across the paper.

A (discrete) *probability distribution* μ over a denumerable set S is a function $\mu : S \rightarrow [0, 1]$ such that $\mu(S) = \sum_{s \in S} \mu(s) = 1$. Let $\mathcal{D}(S)$ denote the set of all probability distributions on S . $\Delta_s \in \mathcal{D}(S)$ denotes the Dirac distribution for $s \in S$, i.e., $\Delta_s(s) = 1$ and $\Delta_s(s') = 0$ for all $s' \in S$ such that $s' \neq s$. The *support* set of μ is defined by $Supp(\mu) = \{s \mid \mu(s) > 0\}$.

Given a set V , V^* (resp. V^∞) denotes the set of all finite (resp. infinite) sequences of elements of V . Concatenation is represented using juxtaposition. We use variables $\omega, \omega', \dots \in V^\infty$ as ranging over infinite sequences, and variables $\hat{\omega}, \hat{\omega}', \dots \in V^*$ as ranging over finite sequences. The i -th element of a finite (resp. infinite) sequence $\hat{\omega}$ (resp. ω) is denoted $\hat{\omega}_i$ (resp. ω_i). Furthermore, for any finite sequence $\hat{\omega}$, $|\hat{\omega}|$ denotes its length. For $\omega \in V^\infty$, $\text{inf}(\omega)$ denotes the set of items appearing infinitely often in ω . Given $S \subseteq V^*$, S^k is the set obtained by concatenating k times the sequences in S .

A *stochastic game* [11, 28] is a tuple $\mathcal{G} = (V, (V_1, V_2, V_P), \delta)$, where V is a finite set of vertices (or states) with $V_1, V_2, V_P \subseteq V$ being a partition of V , and $\delta : V \times V \rightarrow [0, 1]$ is a probabilistic transition function, such that for every $v \in V_1 \cup V_2$, $\delta(v, v') \in \{0, 1\}$, for any $v' \in V$; and $\delta(v, \cdot) \in \mathcal{D}(V)$ for $v \in V_P$. If $V_P = \emptyset$, then \mathcal{G} is called a two-player game graph. Moreover, if $V_1 = \emptyset$ or $V_2 = \emptyset$, then \mathcal{G} is a *Markov decision process* (or MDP). Finally, in case that $V_1 = \emptyset$ and $V_2 = \emptyset$, \mathcal{G} is a *Markov chain* (or MC). For all states $v \in V$ we define $\text{post}^\delta(v) = \{v' \in V \mid \delta(v, v') > 0\}$, the set of successors of v . Similarly, $\text{pre}^\delta(v') = \{v \in V \mid \delta(v, v') > 0\}$ as the set of predecessors of v' , we omit the index δ when it is clear from context. Also, when useful, we fix an initial state for a game, in such a case we use the notation \mathcal{G}_v to indicate that the game starts from v . Furthermore, we assume that $\text{post}(v) \neq \emptyset$ for every $v \in V$. A vertex $v \in V$ is said to be *terminal* if $\delta(v, v) = 1$, and $\delta(v, v') = 0$ for all $v \neq v'$. Most results on MDPs rely on the notion of *end component* [5], we straightforwardly extend this notion to two-player games: an end component of \mathcal{G} is a pair (V', δ') such that (a) $V' \subseteq V$; (b) $\delta'(v) = \delta(v)$ for $v \in V_P$; (c) $\emptyset \neq \text{post}^{\delta'}(v) \subseteq \text{post}^\delta(v)$ for $v \in V_1 \cup V_2$; (d) $\text{post}^{\delta'}(v) \subseteq V'$ for all $v \in V'$; (e) the underlying graph of (V', δ') is strongly connected. Note that an end component can also be considered as being a game. The set of end components of \mathcal{G} is denoted $EC(\mathcal{G})$.

A *path* in the game \mathcal{G} is an infinite sequence of vertices $v_0 v_1 \dots$ such that $\delta(v_k, v_{k+1}) > 0$ for every $k \in \mathbb{N}$. $\text{Paths}_{\mathcal{G}}$ denotes the set of all paths, and $\text{FPaths}_{\mathcal{G}}$ denotes the set of finite prefixes of paths. Similarly, $\text{Paths}_{\mathcal{G}, v}$ and $\text{FPaths}_{\mathcal{G}, v}$ denote the set of paths and the set of finite paths starting at vertex v .

A *strategy* for Player i (for $i \in \{1, 2\}$) in a game \mathcal{G} is a function $\pi_i : V^*V_i \rightarrow \mathcal{D}(V)$ that assigns a probabilistic distribution to each finite sequence of states such that $\pi_i(\hat{\omega}v)(v') > 0$ only if $v' \in \text{post}(v)$. The set of all the strategies for Player i is named Π_i . A strategy π_i is said to be *pure* or *deterministic* if, for every $\hat{\omega}v \in V^*V_i$, $\pi_i(\hat{\omega}v)$ is a Dirac distribution, and it is called *memoryless* if $\pi_i(\hat{\omega}v) = \pi_i(v)$, for every $\hat{\omega} \in V^*$. Let Π_i^M and Π_i^D be respectively the set of all memoryless strategies and the set of all deterministic strategies for Player i . $\Pi_i^{MD} = \Pi_i^M \cap \Pi_i^D$ is the set of all its deterministic and memoryless strategies.

Given two strategies $\pi_1 \in \Pi_1$, $\pi_2 \in \Pi_2$ and an initial vertex v , the *result* of the game is a Markov chain [11], denoted $\mathcal{G}_v^{\pi_1, \pi_2}$. An event \mathcal{A} is a measurable set in the Borel σ -algebra generated by the cones of $\text{Paths}_{\mathcal{G}}$. The *cone* or *cylinder* spanned by the finite path $\hat{\omega} \in \text{FPaths}_{\mathcal{G}}$ is the set $\text{cyl}(\hat{\omega}) = \{\omega \in \text{Paths}_{\mathcal{G}} \mid \forall 0 \leq i < |\hat{\omega}| : \omega_i = \hat{\omega}_i\}$. $\text{Prob}_{\mathcal{G}, v}^{\pi_1, \pi_2}$ is the associated probability measure obtained when fixing strategies π_1 , π_2 , and an initial vertex v [11]. Intuitively, $\text{Prob}_{\mathcal{G}, v}^{\pi_1, \pi_2}(\mathcal{A})$ is the probability that strategies π_1 and π_2 generates a path belonging to the set \mathcal{A} when the game \mathcal{G} starts in v . When no confusion is possible, we just write $\text{Prob}_{\mathcal{G}, v}^{\pi_1, \pi_2}(\hat{\omega})$ instead of $\text{Prob}_{\mathcal{G}, v}^{\pi_1, \pi_2}(\text{cyl}(\hat{\omega}))$. Similar notations are used for MDPs and MCs. A stochastic game (defined as above) is said to be *stopping* [14] if for all pair of strategies π_1, π_2 the probability of reaching a terminal state is 1. We use LTL notation to represent specific set of paths, e.g., $\diamond T = \{\omega \in \text{Paths}_{\mathcal{G}} \mid \exists i \geq 0 : \omega_i \in T\}$ is the set of all the plays in the game that reach vertices in T .

A *quantitative objective* or *payoff function* is a measurable function $f : V^\infty \rightarrow \mathbb{R}$. Let $\mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[f]$ be the expectation of measurable function f under probability $\text{Prob}_{\mathcal{G}, v}^{\pi_1, \pi_2}$. The goal of Player 1 is to maximize this value whereas the goal of Player 2 is to minimize it. Sometimes quantitative objective functions can be defined via *rewards*. These are assigned by a *reward function* $r : V \rightarrow \mathbb{R}^+$. We usually consider stochastic games augmented with a reward function. Moreover, we assume that for every terminal vertex v , $r(v) = 0$.

The value of the game for Player 1 at vertex v under strategy π_1 is defined as the infimum over all the values resulting from Player 2 strategies in that vertex, i.e., $\inf_{\pi_2 \in \Pi_2} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[f]$. The *value of the game* for Player 1 is defined as the supremum of the values of all Player 1 strategies, i.e., $\sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[f]$. Similarly, the value of the game for a Player 2 under strategy π_2 and the value of the game for Player 2 are defined as $\sup_{\pi_1 \in \Pi_1} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[f]$ and $\inf_{\pi_2 \in \Pi_2} \sup_{\pi_1 \in \Pi_1} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[f]$, respectively. We say that a game is *determined* if both values are the same, that is, $\sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[f] = \inf_{\pi_2 \in \Pi_2} \sup_{\pi_1 \in \Pi_1} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[f]$. Martin [24] proved the determinacy of stochastic games for Borel and bounded objective functions.

In this paper we focus on the *total accumulated reward payoff* function, i.e., $\text{rew}(\omega) = \sum_{i=0}^{\infty} r(\omega_i)$. Since rew is unbounded, the results of Martin [24] do not apply to this function. In this paper we restrict ourselves to non-negative rewards, as shown in the next sections, non-negative rewards are enough to deal with interesting case studies, we briefly discuss in Sect. 7 the possible extension of the results presented here to games having negative rewards.

4 Stopping Games and Fair Strategies

We begin this section by introducing the notions of (*almost sure*) *fair strategy* and *stopping games under fairness*. From now on, we assume that Player 2 represents the environment, which tries to minimize the amount of rewards obtained by the system, thus fairness restrictions will be applied to this player.

Definition 1. *Given a stochastic game $\mathcal{G} = (V, (V_1, V_2, V_P), \delta)$. The set of fair plays for Player 2 (denoted FP^2) is defined as follows:*

$$FP^2 = \{\omega \in Paths_{\mathcal{G}} \mid \forall v' \in V_2 : v' \in \text{inf}(\omega) \Rightarrow \text{post}(v') \subseteq \text{inf}(\omega)\}$$

Alternatively, if we consider each vertex as a proposition, FP^2 can be written using LTL notation as: $\bigwedge_{v \in V_2} \bigwedge_{v' \in \text{post}(v)} (\Box \diamond v \Rightarrow \Box \diamond v')$. This property is ω -regular, thus it is measurable in the σ -algebra generated by the cones of $Paths_{\mathcal{G}}$ (see e.g., [5, p.804]). This is a state-based notion of fairness, but it can be straightforwardly extended to settings where transitions are considered. For the sake of simplicity we do not do so in this paper.

Next, we introduce the notion of (almost-sure) *fair strategies* for Player 2.

Definition 2. *Given a stochastic game $\mathcal{G} = (V, (V_1, V_2, V_P), \delta)$, a strategy $\pi_2 \in \Pi_2$ is said to be almost-sure fair (or simply fair) iff it holds that: $Prob_{\mathcal{G},v}^{\pi_1, \pi_2}(FP^2) = 1$, for every $\pi_1 \in \Pi_1$ and $v \in V$.*

The set of all the fair strategies for Player 2 is denoted by $\Pi_2^{\mathcal{F}}$. We combine this notation with the notation introduced in Sect. 3, e.g., Π_2^{MF} refers to the set of all memoryless and fair strategies for Player 2. The previous definition is based on the notion of fair scheduler as introduced for Markov decision processes [5, 7].

Note that for stopping games, every strategy is fair, because the probability of visiting a vertex infinitely often is 0. Also notice that there are games which are not stopping, but they become stopping if Player 2 uses only fair strategies. This is the main idea behind the notion of *stopping under fairness* as introduced in the following definition.

Definition 3. *A stochastic game $\mathcal{G} = (V, (V_1, V_2, V_P), \delta)$ is said to be stopping under fairness iff for all strategies $\pi_1 \in \Pi_1, \pi_2 \in \Pi_2^{\mathcal{F}}$ and vertex $v \in V$, it holds that $Prob_{\mathcal{G},v}^{\pi_1, \pi_2}(\diamond T) = 1$, where T is the set of terminal vertices of \mathcal{G} .*

Checking stopping criteria. This section is devoted to the effective characterization of games that are stopping under fairness. The following lemma states that, for every game that is not stopping under fairness, there is a *memoryless deterministic* strategy for Player 1 and a fair strategy for Player 2 that witnesses it.

Lemma 1. *Let $\mathcal{G} = (V, (V_1, V_2, V_P), \delta)$ be a stochastic game, $v \in V$, and T the set of terminal states of \mathcal{G} . If $Prob_{\mathcal{G},v}^{\pi_1, \pi_2}(\diamond T) < 1$ for some $\pi_1 \in \Pi_1$ and $\pi_2 \in \Pi_2^{\mathcal{F}}$, then, for some memoryless and deterministic strategy $\pi'_1 \in \Pi_1^{MD}$ and fair strategy $\pi'_2 \in \Pi_2^{\mathcal{F}}$, $Prob_{\mathcal{G},v}^{\pi'_1, \pi'_2}(\diamond T) < 1$.*

The proof of this lemma follows by noticing that, if $Prob_{\mathcal{G},v}^{\pi_1,\pi_2}(\diamond T) < 1$, there must be a finite path that leads with some probability to an end component not containing a terminal state and which is a trap for the fair strategy π_2 . This part of the game enables the construction of a memoryless deterministic strategy for Player 1 by ensuring that it follows the same finite path (but skipping loops) and that it traps Player 2 in the same end component.

The next theorem states that checking stopping under fairness in a stochastic game \mathcal{G} can be reduced to check the stopping criteria in a MDP, which is obtained from \mathcal{G} by fixing a strategy in Player 2 that selects among the output transitions according to a uniform distribution. Thus, this theorem enables a graph solution to determine stopping under fairness.

Theorem 1. *Let $\mathcal{G} = (V, (V_1, V_2, V_P), \delta)$ be a stochastic game and T its set of terminal states. Consider the Player 2 (memoryless) strategy $\pi_2^u : V_2 \rightarrow \mathcal{D}(V)$ defined by $\pi_2^u(v)(v') = \frac{1}{\#post(v)}$, for all $v \in V_2$ and $v' \in post(v)$. Then, \mathcal{G} is stopping under fairness iff $Prob_{\mathcal{G},v}^{\pi_1,\pi_2^u}(\diamond T) = 1$ for every $v \in V$ and $\pi_1 \in \Pi_1$.*

While the “only if” part of the theorem is direct, the “if” part is proved by contraposition using Lemma 1.

Theorem 1 introduces an algorithm to check if the stochastic game \mathcal{G} is stopping under fairness: transform \mathcal{G} into the MDP $\mathcal{G}^{\pi_2^u}$ by fixing π_2^u in \mathcal{G} and check whether $Prob_{\mathcal{G}^{\pi_2^u},v}^{\pi_1}(\diamond T) = 1$ for all $v \in V$. As a consequence, we have the following theorem.

Theorem 2. *Checking whether the stochastic game \mathcal{G} is stopping under fairness or not is in $O(\text{poly}(\text{size}(\mathcal{G})))$.*

Alternatively, we can use Theorem 1 to provide a direct algorithm on \mathcal{G} and avoiding the construction of the intermediate MDP. The main idea is to use a modification of the standard *pre* operator, as shown in the following definition:

$$\exists Pre_f(C) = \{v \in V \mid \delta(v, C) > 0\}$$

$$\forall Pre_f(C) = \{v \in V_2 \cup V_P \mid \delta(v, C) > 0\} \cup \{v \in V_1 \mid \forall v' \in V : \delta(v, v') > 0 \Rightarrow v' \in C\}$$

As usual we consider the transitive closures of these operators denoted $\exists Pre_f^*$ and $\forall Pre_f^*$, respectively.

Theorem 3. *Let $\mathcal{G} = (V, (V_1, V_2, V_P), \delta)$, be a stochastic game and let T be the set of its terminal states. Then, (1) $Prob_{\mathcal{G},v}^{\pi_1,\pi_2}(\diamond T) = 1$ for every $\pi_1 \in \Pi_1$ and $\pi_2 \in \Pi_2^f$ iff $v \in V \setminus \exists Pre_f^*(V \setminus \forall Pre_f^*(T))$, and (2) \mathcal{G} is stopping under fairness iff $\exists Pre_f^*(V \setminus \forall Pre_f^*(T)) = \emptyset$.*

Determinacy of Stopping Games under Fairness. The determinacy of stochastic games with Borel and bounded payoff functions follows from Martin’s results [24]. The function *rew* is unbounded, so Martin’s theorems do not apply to it. In [18], the determinacy of a general class of stopping stochastic games (called *transient*) with total rewards is proven. However, note that we restrict Player 2 to only play

with fair strategies and hence, the last result does not apply either. In [26] the authors classify Player 2’s strategies into proper (those ensuring termination) and improper (those prolonging the game indefinitely). For proving determinacy, the authors assume that the value of the game for Player 2’s improper strategies is ∞ . It is worth noting that, for proving the results below, we do not make any assumption about unfair strategies. In the following we prove that the restriction to fair plays does not affect the determinacy of the games.

Figure 3 shows the dependencies of the lemmas that eventually lead to our main results, namely, Theorem 4, which states that the general problem can be limited to only memoryless and deterministic strategies, and Theorem 5, which establishes determinacy and the correctness of the algorithmic solution through the Bellman equations. To prove Theorem 4 we use the intermediate notion of *semi-Markov* strategies [18] and a first step to this reduction is presented in Lemma 2. Lemmas 3 and 4 ensure the transient characteristics of stopping under fairness problems. They are essential to prove that every possible total reward play yields a solution (Lemma 5). Already approaching Theorem 4, Lemma 6 states that there is always a minimizing fair strategy that is memoryless and deterministic, and Lemma 7 helps to reduce the problem from the domain of semi-Markov strategies to the domain of memoryless deterministic strategies. Using Theorem 4 and Proposition 1, which states that the Bellman equations are well behaved in the lattice of solutions, Theorem 5 is finally proved.

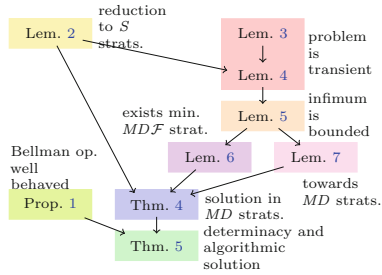


Fig. 3. A roadmap to proving Theorems 4 and 5

Intuitively, a semi-Markov strategy only takes into account the length of a play, the initial state, and the current state to select the next step in the play.

Definition 4. Let $\mathcal{G} = (V, (V_1, V_2, V_P), \delta)$ be a stochastic game. A strategy $\pi_i \in \Pi_i$ is called *semi-Markov* if: $\pi_i(v\hat{\omega}v') = \pi_i(v\hat{\omega}'v')$, for every $v \in V$ and $\hat{\omega}, \hat{\omega}' \in V^*$ such that $|\hat{\omega}| = |\hat{\omega}'|$.

Notice that, by fixing an initial state v , a semi-Markov strategy π_i can be thought of as a sequence of memoryless strategies $\pi_i^{0,v} \pi_i^{1,v} \pi_i^{2,v} \dots$ where $\pi_i(v) = \pi_i^{0,v}(v)$ and $\pi_i(v\hat{\omega}v') = \pi_i^{|\hat{\omega}|+1,v}(v')$. The set of all semi-Markov (resp. semi-Markov fair) strategies for player i is denoted Π_i^S (resp. Π_i^{SF}).

The importance of semi-Markov strategies lies in the fact that, when Player 2 plays a semi-Markov strategy, any Player 1’s strategy can be mimicked by a semi-Markov strategy as stated in the following lemma.

Lemma 2. Let \mathcal{G} be a stopping under fairness stochastic game, and let $\pi_2 \in \Pi_2^{SF}$ be a fair and semi-Markov strategy. Then, for any $\pi_1 \in \Pi_1$, there is a semi-Markov strategy $\pi_1^* \in \Pi_1^S$ such that $\mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2} [rew] = \mathbb{E}_{\mathcal{G},v}^{\pi_1^*, \pi_2} [rew]$.

Proof (Sketch). The proof follows the arguments of Theorem 4.2.7 in [18] adapted to our setting.

Consider the event $\diamond^k v' = \{\omega \in \text{Paths}_{\mathcal{G}} \mid \omega_k = v'\}$, for $k \geq 0$. That is, the set of runs in which v' is reached after exactly k steps. We define π_1^* as follows. For v' with $\text{Prob}_{\mathcal{G},v}^{\pi_1,\pi_2}(\diamond^k v') > 0$ and $|\hat{\omega}v'| = k$,

$$\pi_1^*(\hat{\omega}v')(v'') = \text{Prob}_{\mathcal{G},v}^{\pi_1,\pi_2}(\diamond^{k+1}v'' \mid \diamond^k v').$$

For v' with $\text{Prob}_{\mathcal{G},v}^{\pi_1,\pi_2}(\diamond^k v') = 0$ and $|\hat{\omega}v'| = k$ we define $\pi_1^*(\hat{\omega}v')$ to be the uniform distribution on $\text{post}(v')$. Notice that π_1^* is a semi-Markov strategy. We prove that π_1^* is the strategy that satisfies the conclusion of the lemma. For this, we first show that $\text{Prob}_{\mathcal{G},v}^{\pi_1,\pi_2}(\diamond^k v') = \text{Prob}_{\mathcal{G},v}^{\pi_1^*,\pi_2}(\diamond^k v')$ by induction on k , and use it to conclude the following.

$$\begin{aligned} \mathbb{E}_{\mathcal{G},v}^{\pi_1,\pi_2}[\text{rew}] &= \sum_{N=0}^{\infty} \sum_{\hat{\omega} \in V^{N+1}} \text{Prob}_{\mathcal{G},v}^{\pi_1,\pi_2}(\hat{\omega}) r(\hat{\omega}_N) = \sum_{N=0}^{\infty} \sum_{v' \in V} \text{Prob}_{\mathcal{G},v}^{\pi_1,\pi_2}(\diamond^N v') r(v') \\ &= \sum_{N=0}^{\infty} \sum_{v' \in V} \text{Prob}_{\mathcal{G},v}^{\pi_1^*,\pi_2}(\diamond^N v') r(v') = \mathbb{E}_{\mathcal{G},v}^{\pi_1^*,\pi_2}[\text{rew}] \quad \square \end{aligned}$$

In a stopping game, all non-terminal states are transient (a state is transient if the expected time that both players spend in it is finite). In fact, [18] defines a stopping game with terminal states in T as a *transient game*, i.e., a game in which $\sum_{N=1}^{\infty} \sum_{\hat{\omega} \in (V \setminus T)^N} \text{Prob}_{\mathcal{G},v}^{\pi_1,\pi_2}(\hat{\omega}) < \infty$ for all strategies $\pi_1 \in \Pi_1$ and $\pi_2 \in \Pi_2$. Obviously, this generality does not hold in our case since unfair strategies make the game dwell infinitely on a set of non-terminal states. Therefore, we prove a weaker property in our setting. Roughly speaking, the next lemma states that, in games that stop under fairness, non-terminal states are transient, provided that the two players play memoryless strategies, and in particular, that Player 2 plays only fair.

Lemma 3. *Let $\mathcal{G} = (V, (V_1, V_2, V_P), \delta)$ be a stochastic game that is stopping under fairness with T being the set of terminal states. Let $\pi_1 \in \Pi_1^M$ be a memoryless strategy for Player 1 and $\pi_2 \in \Pi_2^{MF}$ a memoryless fair strategy for Player 2. Then $\sum_{N=1}^{\infty} \sum_{\hat{\omega} \in (V \setminus T)^N} \text{Prob}_{\mathcal{G},v}^{\pi_1,\pi_2}(\hat{\omega}) < \infty$.*

This result can be extended to all the strategies of Player 1. The main idea behind the proof is to fix a stationary fair strategy for Player 2 (e.g., a uniform distributed strategy). This yields an MDP that stops for every strategy of Player 1, and furthermore, it can be seen as a one-player *transient game* (as defined in [18]). Hence, the result follows from Lemma 3 and Theorem 4.2.12 in [18].

Lemma 4. *Let \mathcal{G} be a stochastic game that is stopping under fairness and let T be the set of terminal states. In addition, let $\pi_1 \in \Pi_1$ be a strategy for Player 1 and $\pi_2 \in \Pi_2^{MF}$ be a fair and memoryless strategy for Player 2. Then $\sum_{N=0}^{\infty} \sum_{\hat{\omega} \in v(V \setminus T)^N} \text{Prob}_{\mathcal{G},v}^{\pi_1,\pi_2}(\hat{\omega}) < \infty$.*

Using the previous lemma, some fairly simple calculations lead to the fact that the value of the total accumulated reward payoff game is well-defined for any strategy of the players. As a consequence, the value of the game is bounded from above for any Player 1's strategy. This is stated in the next lemma.

Lemma 5. *Let $\mathcal{G} = (V, (V_1, V_2, V_P), \delta, r)$ be a stochastic game that is stopping under fairness, $\pi_1 \in \Pi_1$ a strategy for Player 1. Then, for all memoryless fair strategy $\pi_2 \in \Pi_2^{MF}$ for Player 2 and all $v \in V$, $\mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[\text{rew}] < \infty$. Moreover, for every vertex $v \in V$, $\inf_{\pi_2 \in \Pi_2^F} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[\text{rew}] < \infty$.*

The following lemma is crucial and plays an important role in the rest of the paper. Intuitively, it states that, when Player 1 plays with a memoryless strategy, Player 2 has an optimal deterministic memoryless fair strategy. This lemma is the guarantee of the eventual existence of a minimizing memoryless deterministic fair strategy for Player 2 in general.

Lemma 6. *Let $\mathcal{G} = (V, (V_1, V_2, V_P), \delta, r)$ be a stochastic game that is stopping under fairness and let $\pi_1 \in \Pi_1^M$ be a memoryless strategy for Player 1. There exists a deterministic memoryless fair strategy $\pi_2^* \in \Pi_2^{MDF}$ such that $\inf_{\pi_2 \in \Pi_2^F} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[\text{rew}] = \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2^*}[\text{rew}]$, for every $v \in V$.*

Proof (Sketch). Though it differs in the details, the proof strategy is inspired by the proof of Lemma 10.102 in [5]. We first construct a reduced MDP $\mathcal{G}_{\min}^{\pi_1}$ which preserves exactly the optimizing part of the MDP \mathcal{G}^{π_1} . Thus $\delta_{\min}^{\pi_1}(v, v') = \delta^{\pi_1}(v, v')$ if $v \in V_1 \cup V_P$, or $v \in V_2$ and $x_v = r(v) + x_{v'}$, where, for every $v \in V$, $x_v = \inf_{\pi_2 \in \Pi_2^F} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[\text{rew}]$ (which exists due to Lemma 5). Otherwise, $\delta_{\min}^{\pi_1}(v, v') = 0$. $\mathcal{G}_{\min}^{\pi_1}$ can be proved to be stopping under fairness.

Then, the strategy π_2^* for $\mathcal{G}_{\min}^{\pi_1}$ is constructed as follows. For every $v \in V$, let $\|v\|$ be the length of the shortest path fragment to some terminal vertex in T in the MDP $\mathcal{G}_{\min}^{\pi_1}$. Define $\pi_2^*(v)(v') = 1$ for some v' such that $\delta_{\min}^{\pi_1}(v, v') = 1$ and $\|v\| = \|v'\| + 1$. By definition, π_2^* is memoryless. We prove first that π_2^* yields the optimal solution of \mathcal{G}^{π_1} by showing that the vector $(x_v)_{v \in V}$ (i.e., the optimal values of \mathcal{G}^{π_1}) is a solution to the set of equations for expected rewards of the Markov chain $\mathcal{G}^{\pi_1, \pi_2^*}$. Being the solution unique, we have that $x_v = \mathbb{E}_{\mathcal{G}^{\pi_1, \pi_2^*}, v}[\text{rew}]$ for all $v \in V$ and hence the optimality of π_2^* . To conclude the proof we show by contradiction that π_2^* is fair. \square

As already noted, semi-Markov strategies can be thought of as sequences of memoryless strategies. The next lemma uses this fact to show that, when Player 2 plays a memoryless and fair strategy, semi-Markov strategies do not improve the value that Player 1 can obtain via memoryless deterministic strategies. The proof of the following lemma adapts the ideas of Theorem 4.2.9 in [18] to our games.

Lemma 7. *For any stochastic game \mathcal{G} that is stopping under fairness, and vertex v , it holds that:*

$$\sup_{\pi_1 \in \Pi_1^S} \inf_{\pi_2 \in \Pi_2^{MDF}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[\text{rew}] = \sup_{\pi_1 \in \Pi_1^{MD}} \inf_{\pi_2 \in \Pi_2^{MDF}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[\text{rew}]$$

Using the previous lemma, we can conclude that the problem of finding $\sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \mathbb{E}^{\pi_1, \pi_2}[rew]$, for any vertex v , can be solve by only focusing on deterministic memoryless strategies as stated and proved in the following theorem.

Theorem 4. *For any stochastic game \mathcal{G} that is stopping under fairness we have:*

$$\sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[rew] = \sup_{\pi_1 \in \Pi_1^{MD}} \inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[rew]$$

Proof. First, we prove that the left-hand term is less than or equal to the right-hand one:

$$\begin{aligned} \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[rew] &\leq \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[rew] \\ &\leq \sup_{\pi_1 \in \Pi_1^S} \inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[rew] \\ &\leq \sup_{\pi_1 \in \Pi_1^{MD}} \inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[rew]. \end{aligned}$$

The first inequality follows from $\Pi_2^{MD\mathcal{F}} \subseteq \Pi_2^{\mathcal{F}}$, the second inequality is due to Lemma 2 and the fact that memoryless strategies are semi-Markov, and the last inequality is obtained by applying Lemma 7.

To prove the other inequality, we calculate:

$$\begin{aligned} \sup_{\pi_1 \in \Pi_1^{MD}} \inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[rew] &= \sup_{\pi_1 \in \Pi_1^{MD}} \inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[rew] \\ &\leq \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \mathbb{E}_{\mathcal{G}, v}^{\pi_1, \pi_2}[rew]. \end{aligned}$$

The first equality is a consequence of Lemma 6 and the second inequality is due to properties of suprema. \square

The standard technique to prove the determinacy of stopping games is by showing that the Bellman operator

$$\Gamma(f)(v) = \begin{cases} r(v) + \sum_{v' \in \text{post}(v)} \delta(v, v') f(v') & \text{if } v \in V_P \setminus T \\ \max\{r(v) + f(v') \mid v' \in \text{post}(v)\} & \text{if } v \in V_1 \setminus T, \\ \min\{r(v) + f(v') \mid v' \in \text{post}(v)\} & \text{if } v \in V_2 \setminus T, \\ 0 & \text{if } v \in T. \end{cases}$$

has a unique fixpoint. However, in the case of games stopping under fairness, Γ has several fixpoints as shown by the next example.

Example 1. Consider the (one-player) game in Fig. 4, where Player 1's vertices are drawn as boxes, Player 2's vertices are drawn as diamonds, and probabilistic vertices are depicted as circles. Note that, in that game, the greatest fixpoint is $(1, 1, 1, 0)$. Yet, $(0.5, 0.5, 1, 0)$ is also a

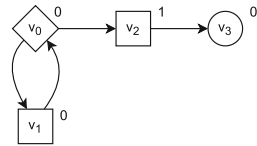


Fig. 4. A game with infinite fixpoints

fixpoint as $\Gamma(0.5, 0.5, 1, 0) = (0.5, 0.5, 1, 0)$. In fact, the Bellman operator for this game has infinite fixpoints: any f of the form $(x, x, 1, 0)$ with $x \in [0, 1]$.

Thus, the standard approach cannot be used here. Instead, we use the greatest fixpoint for proving determinacy, but this cannot be done directly on Γ . A main difficulty is that the Knaster-Tarski theorem does not apply for Γ since (\mathbb{R}^V, \leq) is not a complete lattice. Using instead the extended reals $(\mathbb{R} \cup \{\infty\})^V$ is not a solution, as in some cases the greatest fixpoint will assign ∞ to some vertices (e.g., $(\infty, \infty, 0)$ would be the greatest fixpoint in the Markov chain of Fig. 5). One possible approach is to approximate the greatest fixpoint from an estimated upper bound via value iteration. Unfortunately, there may not be an order relation between f and $\Gamma(f)$ and it may turn out that for some vertex v , $\Gamma(f)(v) > f(v)$ before converging to the fixpoint. This is shown in the next example.

Example 2. Consider the game depicted in Fig. 5. The (unique) fixpoint in this case is $(100, 90, 0)$. Observe that, we have that $\Gamma(120, 100, 0) = (110, 108, 0)$, thus the value at v_1 increases after one iteration. Several iterations are needed then to reach the greatest fixpoint. Thus, in general, starting value iteration from an estimated upper bound does not guarantee a monotone convergence to the greatest fixpoint.

We overcome the aforementioned issues by using a modified version of Γ . Roughly speaking, we modify the Bellman operator in such a way that it operates over a complete lattice.

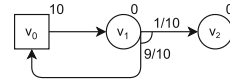


Fig. 5. A game where value iteration may go up

Notice that, by Lemma 5, the value $\mathbb{E}_{\mathcal{G},v}^{\pi_1,\pi_2}[rew]$ is finite for every stopping game under fairness \mathcal{G} and strategies $\pi_1 \in \Pi_1^{MD}$, $\pi_2 \in \Pi_2^{MD\mathcal{F}}$. Furthermore, because the number of deterministic memoryless strategies is finite, we also have that the number $\max\{\inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \sup_{\pi_1 \in \Pi_1^{MD}} \mathbb{E}_{\mathcal{G},v}^{\pi_1,\pi_2}[rew] \mid v \in V\}$ is well defined. From now on, fix a number $\mathbf{U} \geq \max\{\inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \sup_{\pi_1 \in \Pi_1^{MD}} \mathbb{E}_{\mathcal{G},v}^{\pi_1,\pi_2}[rew] \mid v \in V\}$. We define a modified Bellman operator $\Gamma^* : [0, \mathbf{U}]^V \rightarrow [0, \mathbf{U}]^V$ as follows.

$$\Gamma^*(f)(v) = \begin{cases} \min(r(v) + \sum_{v' \in post(v)} \delta(v, v')f(v'), \mathbf{U}) & \text{if } v \in V_P \setminus T \\ \min(\max\{r(v) + f(v') \mid v' \in post(v)\}, \mathbf{U}) & \text{if } v \in V_1 \setminus T, \\ \min(\min\{r(v) + f(v') \mid v' \in post(v)\}, \mathbf{U}) & \text{if } v \in V_2 \setminus T, \\ 0 & \text{if } v \in T. \end{cases}$$

Note that Γ^* is monotone, which can be proven by observing that maxima, minima and convex combinations are all monotone operators. Furthermore, Γ^* is also Scott continuous (it preserves suprema of directed sets), this can be proven similarly as in [10]. The following proposition formalizes these properties.

Proposition 1. *Γ^* is monotone and Scott-continuous.*

Note that $([0, \mathbf{U}]^V, \leq)$ is a complete lattice. Thus by Proposition 1 and the Knaster-Tarski theorem [15], the (non-empty) set of fixed points of Γ^* forms a complete lattice, and the greatest fixpoint of the operator can be approximated

by successive applications of Γ^* to the top element (i.e., \mathbf{U}) [15]. In the following we denote by $\nu\Gamma^*$ the greatest fixed point of Γ^* .

The following theorem states that games restricted to fair strategies on Player 2 are determinate. Furthermore, the value of the game is given by the greatest fixpoint of Γ^* .

Theorem 5. *Let \mathcal{G} be a stochastic game that is stopping under fairness. It holds that:*

$$\inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \sup_{\pi_1 \in \Pi_1} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}] = \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}] = \nu\Gamma^*(v)$$

Proof. First, note that $\inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \sup_{\pi_1 \in \Pi_1^{MD}} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}]$ is a fixed point of Γ^* . Thus we have:

$$\begin{aligned} \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}] &\leq \inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \sup_{\pi_1 \in \Pi_1} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}] \\ &\leq \inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \sup_{\pi_1 \in \Pi_1^{MD}} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}] \leq \nu\Gamma^*(v) \end{aligned}$$

for any v . The first inequality is a standard property of suprema and infima [21], the second inequality holds because $\Pi_2^{MD\mathcal{F}} \subseteq \Pi_2^{\mathcal{F}}$ and standard properties of MDPs: by fixing a deterministic memoryless fair strategy for Player 2 we obtain a transient MDP, the optimal strategy for Player 1 in this MDP is obtained via a deterministic memoryless strategy [20]. The last inequality holds because $\inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \sup_{\pi_1 \in \Pi_1^{MD}} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}]$ is a fixpoint of Γ^* .

Rest to prove that $\sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}] \geq \nu\Gamma^*(v)$. Note that, if there is $\pi_1 \in \Pi_1$ such that $\inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}] \geq \nu\Gamma^*(v)$ the property above follows by properties of supremum. Consider the strategy π_1^* defined as follows: $\pi_1^*(v) \in \text{argmax}\{\nu\Gamma^*(v') + r(v) \mid v' \in \text{post}(v)\}$. Note that π_1^* is a memoryless and deterministic strategy. For any memoryless, deterministic and fair strategy $\pi_2 \in \Pi_2^{MD\mathcal{F}}$ we have $\nu\Gamma^*(v) \leq \mathbb{E}_{\mathcal{G},v}^{\pi_1^*, \pi_2}[\text{rew}]$ (by definition of Γ^*). Thus, $\nu\Gamma^*(v) \leq \inf_{\pi_2 \in \Pi_2^{MD\mathcal{F}}} \mathbb{E}_{\mathcal{G},v}^{\pi_1^*, \pi_2}[\text{rew}]$ and then: $\nu\Gamma^*(v) \leq \sup_{\pi_1 \in \Pi_1^{MD}} \inf_{\pi_2 \in \Pi_1^{MD\mathcal{F}}} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}]$. Finally, by Theorem 4 we get: $\nu\Gamma^*(v) \leq \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2^{\mathcal{F}}} \mathbb{E}_{\mathcal{G},v}^{\pi_1, \pi_2}[\text{rew}]$. \square

Considerations for an algorithmic solution. Value iteration [9] has been used to compute maximum/minimum expected accumulated reward in MDPs, e.g., in the PRISM model checker. Usually, the value is computed by approximating the least fixpoint from below using the Bellman equations [9]. In [6], the authors propose to approach these values from both a lower and an upper bound (known as interval iteration [19]). To do so, [6] shows a technique for computing upper bounds for the expected total rewards for MDPs. This approach is based on the fact that, given a stopping MDP \mathcal{G} , $\mathbb{E}_{\mathcal{G},v}^{\pi_1}[\text{rew}] = \sum_{v' \in R(v)} \zeta_v^{\pi_1}(v') * r(v')$, where $R(v)$ denotes the set of reachable states from v , and $\zeta_v^{\pi_1}(v')$ denotes the expected number of times to visit v' in the Markov chain induced by π_1 when starting at v . [6] describes how to compute a value $\zeta_v^*(v')$, such that $\zeta_v^*(v') \geq \sup_{\pi_1 \in \Pi_1} \zeta_v^{\pi_1}(v')$. Thus, $\sum_{v' \in R(v)} \zeta_v^*(v') * r(v')$ gives an upper bound for $\sup_{\pi_1} \mathbb{E}_{\mathcal{G},v}^{\pi_1}[\text{rew}]$. Our algorithm uses these ideas to provide an upper bound for two-player games. Roughly

speaking, the above defined functional Γ^* presents a form of Bellman equations that enables a value iteration algorithm to solve these games. We need to start with some value vector larger than such a fixpoint. Given a stopping under fairness game, we fix a (memoryless) fair strategy for the environment, thus obtaining an MDP. We then use the techniques described above to find an upper bound for this MDP, which in turn is an upper bound in the original game. The obvious fair strategy to use is the one based on the uniform distribution (as in Theorem 1). This idea is described in Algorithm 1. It is worth noting that, instead of using a unique upper bound for every vertex (as in the definition of Γ^*), the algorithm may use a different upper bound for each component of the value vector, this improves the number of iterations performed by the algorithm. We have implemented Algorithm 1 as a prototype embedded in the PRISM-games toolset [22], as described in the next section.

Algorithm 1 Algorithm for computing GFP

Require: \mathcal{G} is a stopping under fairness game

$$\delta' \leftarrow \lambda(v, v').(v \in V_1 \cup V_P) ? \delta(v, v') : \frac{1}{|\text{post}^{\mathcal{G}}(v)|}$$

$$\mathcal{G}' \leftarrow (V, (V_1, \emptyset, V_2 \cup V_P), \delta')$$

$$x' \leftarrow \lambda v : \sum_{v' \in R(v)} : \zeta_v^*(v') * r(v')$$

repeat

$$x \leftarrow x'$$

$$x' \leftarrow \Gamma^*(x)$$

until $\|x - x'\| \leq \varepsilon$

return x'

5 Experimental Validation

In order to evaluate the viability of our approach we have extended the model checker PRISM [22, 23] with an operator to compute the expected rewards for stochastic games that stop under fairness. The prototype also allows one to check whether a game is stopping under fairness. The tool takes as input a model describing the game in PRISM notation and returns as output the optimal expected total reward for a given initial state as well as the synthesized optimal controller strategy (under fairness assumptions). The experimental evaluation shows that our approach can cope with non-trivial case studies. For computing these values we set a relative error of at most $\varepsilon = 10^{-6}$.

Roborta vs. the Fair Light. Table 1 shows the results of the example introduced in Sect. 2 for multiple configurations. We considered three variants of the case study: version A (the light does not fail), version B (the light can only fail when trying to signal a green light), and version C (the light can fail when trying to signal any kind of light). We assumed that, when Roborta fails, she cannot move (this is beneficial to Roborta since she can re-collect the reward); when the light fails, the robot can freely move into any allowed direction. The grid configuration (movement restrictions and rewards) are randomly generated. For each setting, Table 1 describes the results for three different scenarios generated starting at different seeds. For the grid configuration shown in Sect. 2 with parameters $P = 0.1$ and $Q = 0$, the tool derived the optimal strategy depicted in Fig. 2 and reports an expected total reward of 5.55.

Autonomous UAV vs. Human Operator. We adapted the case study analyzed in [17]. A remotely controlled Unmanned Aerial Vehicle (UAV) is used to perform intelligence, surveillance, and reconnaissance (ISR) missions over a road network. The UAV performs piloting functions autonomously (selecting a path to fly between *waypoints*). The human operator (environment) controls the onboard sensor to capture imagery at a waypoint as well as the piloting functions on certain waypoints (called checkpoints). Note that an operator can continuously try to get a better image by making the UAV loiter around a certain waypoint, this may lead to an unfair behavior. Each successful capture from an unvisited waypoint grants a reward. Figure 6 shows an example of road network consisting of six surveillance waypoints labeled w_0, w_2, \dots, w_5 , the edges represent paths, a red-dashed line means that the path is dangerous enough to make the UAV stop working with probability 1, while on any other path, this probability is S . Checkpoints are depicted as pink nodes, therein the operator can still delegate the piloting task to the UAV with probability D . Each node is annotated with three possible rewards. For instance, for $S = 0.3$ and $D = 0.5$ and the leftmost reward values in each triple, the synthesized strategy for the UAV tries to follow the optimal circuit $w_0, w_1, w_2, w_3, w_4, w_5$. While for the middle and rightmost reward values, the optimal circuits to follow are $w_0, w_5, w_0, w_1, w_2, w_3, w_4$ and $w_0, w_5, w_4, w_1, w_2, w_3$, respectively. Table 2 shows the results obtained for this game for several randomly generated road networks.

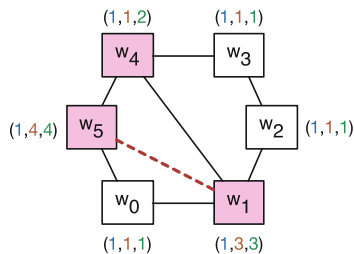


Fig. 6. UAV Network for ISR missions adapted from [17]

Tables 1 and 2 do not report the time taken to compute the results, but in all cases the output was computed in less than 400s. All the experiments were run on a MacBook Air with Intel Core i5 at 1.3 GHz and 4 Gb of RAM.

6 Related Work

Stochastic games with payoff functions have been extensively investigated in the literature. In [18], several results are presented about *transient games*, a generalized version of stopping stochastic games with total reward payoff. In transient games, both players possess optimal (memoryless and deterministic) strategies. Most importantly, the games are determined and their value can be computed as the least fixed point of a set of equations. Most of these results are based on the fact that the Γ functional (see Sect. 4) for transient games has a unique fixed point. Notice that in this paper we have dealt with games that are stopping only under fairness assumptions. Thus, the corresponding functional may have several fixed points. Hence, the main results presented in [18] do not apply to our setting.

[12] and [28] present logical frameworks for the verification and synthesis of systems. While [12] provides a solution for a probabilistic branching temporal

Table 1. Results for Roborta vs. Light Game. First column describes the grid size. Second column indicates the fault probability for the robot (P) and light (Q). The other columns describe the size of the model, the expected total reward for the optimal strategy, and the number of iterations performed, respectively, for three different randomly generated grid configurations.

Version	Fault prob.		Size (States/Transitions)			Opt. Expect. Total Rew.			Iterations		
	P	Q	s. 1	s. 2	s. 3	s. 1	s. 2	s. 3	s. 1	s. 2	s. 3
A 60×8	0.1	—	st. 1448	st. 1418	st. 1421	26.66	31.11	27.77	711	681	252
	0.5	—	tr. 3220	tr. 3112	tr. 3132	48	56	50	2253	2225	475
A 120×16	0.1	—	st. 5686	st. 5716	st. 5716	62.22	55.55	48.88	687	700	685
	0.5	—	tr. 12586	tr. 12658	tr. 12722	112	100	88	2231	2265	2229
B 60×8	0.1	0.1	st. 1928	st. 1888	st. 1892	42.6	44.59	42.23	479	335	388
		0.5				130.14	127.7	136.22	772	689	824
	0.5	0.1	tr. 5952	tr. 5746	tr. 5785	76.68	80.26	76.02	873	764	909
		0.5				234.26	229.87	245.21	1263	1139	1341
B 120×16	0.1	0.1	st. 7576	st. 7616	st. 7616	91.19	87.27	80.07	538	544	616
		0.5				281.83	281.48	265.33	1076	1118	1252
	0.5	0.1	tr. 23266	tr. 23400	tr. 23528	164.15	157.1	144.13	1147	1223	1373
		0.5				507.30	506.67	477.6	1850	1865	2088
C 60×8	0.1	0.1	st. 1928	st. 1888	st. 1892	46.32	47.07	44.87	379	336	390
		0.5				143.35	146.41	153.98	742	658	774
	0.5	0.1	tr. 6432	tr. 6216	tr. 6256	83.37	84.73	80.77	879	769	914
		0.5				258.04	263.53	277.17	1202	1076	1246
C 120×16	0.1	0.1	st. 7576	st. 7616	st. 7616	98.25	93.74	88.33	533	544	606
		0.5				321.18	317.61	311.62	1002	1068	1188
	0.5	0.1	tr. 25156	tr. 25300	tr. 25428	176.85	168.73	158.99	1147	1227	1365
		0.5				578.13	571.71	560.92	1700	1760	1956

Table 2. Results for the UAV vs. Operator Game. First column describes the number of waypoints used. Second column indicates probability of delegation (D), and the probability that the UAV stops working (S). The other columns show the size of the model, the expected total reward for the optimal strategy, and the number of iterations performed, respectively, for three different randomly generated roadmap configurations.

Version	Prob.		Size(States/Transitions)			Opt. Expect. Total Rew.			Iterations		
	D	S	s. 1	s. 2	s. 3	s. 1	s. 2	s. 3	s. 1	s. 2	s. 3
UAV $6w.$	0.1	0.05	st. 213	st. 508	st. 136	16.72	12.47	13.14	142	248	22
		0.1				15.73	11.15	12.63	73	188	22
	0.5	0.05	tr. 504	tr. 1368	tr. 312	20.49	12.77	17.05	103	133	22
		0.1				18.87	11.67	15.95	55	70	22
UAV $8w.$	0.1	0.05	st. 2177	st. 3591	st. 1426	17.88	40.59	24.6	407	332	779
		0.1				17.11	34.3	21.48	280	233	437
	0.5	0.05	tr. 5959	tr. 9991	tr. 3604	26	42.21	30.87	128	214	257
		0.1				23.44	36.08	24.72	116	113	194
UAV $10w.$	0.1	0.05	st. 6631	st. 5072	st. 8272	39.76	28.7	19.76	256	377	356
		0.1				35.43	23.36	16.2	136	260	154
	0.5	0.05	tr. 17306	tr. 13052	tr. 24376	42.13	30.77	24.56	250	247	292
		0.1				37.11	26.08	19.27	130	134	151

logic extended with expected total, discounted, and average reward objective functions, [28] does the same in a similar extension of a probabilistic linear temporal logic. Both frameworks were implemented in the tool PRISM [22, 23]. Although a vast class of properties can be expressed in these frameworks, none of them are presented under fair environments. In fact, these works are on stochastic multiplayer games in which each player is treated equally.

However, of all the operators in [12, 22, 28], $\langle\langle p_1 \rangle\rangle R_{\max=?}[F^{\infty}T]$ is the closest to our proposal and it deserves a deeper comparison. $\langle\langle p_1 \rangle\rangle R_{\max=?}[F^{\infty}T]$ returns the expected accumulated reward until reaching T in which infinite plays receive an infinite value [12, 22]. PRISM approximates this value by computing a greatest fixpoint. It uses a two-phase algorithm to do so: (i) it first replaces zero rewards with a small positive value and applies value iteration on this modification to get an estimated upper bound, and (ii) this upper bound is used to start another value iteration process aimed to compute the greatest fixpoint.

This heuristic could return erroneous approximations of the greatest fixpoint. We illustrate this with a simple example. Consider the game depicted in Fig. 7. For any p , the value of the greatest fixpoint in vertex v_0 is 2. However, by taking $p = 0.99$ and tolerance $\epsilon = 10^{-6}$, PRISM returns a value close to 39608. This occurs because PRISM changes 0 to the value 0.02, which results in an extremely large upper bound. Obviously, it also returns an incorrect strategy for vertex v_0 . We have checked this example with our tool, and it returned the correct value for vertex v_0 in 2 iterations, regardless of the value of p . We have chosen a large value for p to make the difference noticeable. Small values also may produce different values in, e.g., v_1 only that it could be blamed on approximation errors. We have also run this operator on our case studies and observed small differences in many of them (particularly on Robortia) that get larger when the fault probabilities get larger as well.

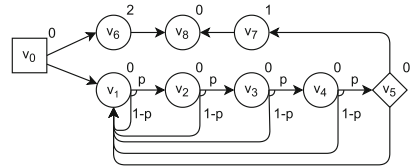


Fig. 7. A simple two-player game: only probability less than 1 are shown

Stochastic shortest path games [26] are two-player stochastic games with (negative or positive) rewards in which the minimizer’s strategies are classified into *proper* and *improper*, proper strategies are those ensuring termination. As proven in [26], these games are determined, and both players possess memoryless optimal strategies. To prove these results, the authors assume that the expected game value for improper strategies is ∞ , this ensures that the corresponding functional is a contraction and thus it has a unique fixpoint. In contrast, we restrict ourselves to non-negative rewards but we do not make any assumptions over unfair strategies, as mentioned above the corresponding functional for our games may have several fixpoints. Furthermore, we proved that the value of the game is given by the greatest fixpoint of T . In recent years, several authors have investigated stochastic shortest path problems for MDPs (i.e., one-player games),

where the assumption over improper strategies is relaxed (e.g., [3]); to the best of our knowledge, these results have not been extended to two-player games.

In [4] the authors tackle the problem of synthesizing a controller that maximizes the probability of satisfying an LTL property. Fairness strategies are used to reduce this problem to the synthesis of a controller maximizing a PCTL property over a product game. However, this article does not address expected rewards and game determinacy under fairness assumptions.

Interestingly, in [2] the authors consider the problem of winning a (non-stochastic) two-player game with fairness assumptions over the environment. The objective of the system is to guarantee an ω -regular property. The authors show that winning in these games is equivalent to almost-sure winning in a Markov decision process. It must be noted that this work only considers non-stochastic games. Furthermore, payoff functions are not considered therein.

Finally, we remark that in *qualitative* ω -regular stochastic games [1] strong fairness can easily be considered by properly transforming the original ω -regular objective. Notably, in this setting, [8] shows that qualitative Rabin conditions on stochastic games can be solved by translating this problem into a two-player (non-stochastic) game with the same Rabin condition under extreme fairness following a somewhat inverse direction to that we used to prove Theorem 2.

7 Concluding Remarks

In this paper, we have investigated the properties of stochastic games with total reward payoff under the assumption that the minimizer (i.e., the environment) plays only with fair strategies. We have shown that, in this scenario, determinacy is preserved and both players have optimal memoryless and deterministic strategies; furthermore, the value of the game can be calculated by approximating a greatest fixed point of a Bellman operator. We have only considered non-negative rewards in this paper. A possible way of extending the results presented here to games with negative rewards is to adapt the techniques presented in [3] for MDPs with negative costs, we leave this as a further work.

In order to show the applicability of our technique, we have presented two examples of applications and an experimental validation over diverse instances of these case studies using our prototype tool. We believe that fairness assumptions allow one to consider more realistic behavior of the environment.

We have not investigated other common payoff functions such as discounted payoff or limiting-average payoff. A benefit of these classes of functions is that the value of games are well-defined even when the games are not stopping. At first sight, the notion of fairness is little relevant for games with discounted payoff, since these kinds of payoff functions take most of their value from the initial parts of runs. For limiting-average the situation is different, and fairness assumptions may be relevant as they could change the value of games, we leave this as further work.

References

1. de Alfaro, L., Henzinger, T.A.: Concurrent omega-regular games. In: 15th Annual IEEE Symposium on Logic in Computer Science, pp. 141–154. IEEE Computer Society (2000). <https://doi.org/10.1109/LICS.2000.855763>
2. Asarin, E., Chane-Yack-Fa, R., Varacca, D.: Fair adversaries and randomization in two-player games. In: Ong, L. (ed.) FoSSaCS 2010. LNCS, vol. 6014, pp. 64–78. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12032-9_6
3. Baier, C., Bertrand, N., Dubslaff, C., Gburek, D., Sankur, O.: Stochastic shortest paths and weight-bounded properties in Markov decision processes. In: Dawar, A., Grädel, E. (eds.) Proceedings of the 33rd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2018, pp. 86–94. ACM (2018). <https://doi.org/10.1145/3209108.3209184>
4. Baier, C., Größer, M., Leucker, M., Bollig, B., Ciesinski, F.: Controller synthesis for probabilistic systems. In: Lévy, J., Mayr, E.W., Mitchell, J.C. (eds.) Exploring New Frontiers of Theoretical Informatics, IFIP 18th World Computer Congress, TC1 3rd International Conference on Theoretical Computer Science (TCS2004). IFIP, vol. 155, pp. 493–506. Kluwer/Springer (2004). https://doi.org/10.1007/1-4020-8141-3_38
5. Baier, C., Katoen, J.P.: Principles of Model Checking. The MIT Press (2008)
6. Baier, C., Klein, J., Leuschner, L., Parker, D., Wunderlich, S.: Ensuring the reliability of your model checker: interval iteration for Markov decision processes. In: Majumdar, R., Kunčák, V. (eds.) CAV 2017. LNCS, vol. 10426, pp. 160–180. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-63387-9_8
7. Baier, C., Kwiatkowska, M.Z.: Model checking for a probabilistic branching time logic with fairness. *Distrib. Comput.* **11**(3), 125–155 (1998). <https://doi.org/10.1007/s004460050046>
8. Banerjee, T., Majumdar, R., Mallik, K., Schmuck, A., Soudjani, S.: A direct symbolic algorithm for solving stochastic Rabin games. In: Fisman, D., Rosu, G. (eds.) Tools and Algorithms for the Construction and Analysis of Systems - 28th International Conference, TACAS 2022, Proceedings, Part II. LNCS, vol. 13244, pp. 81–98. Springer, Cham (2022). https://doi.org/10.1007/978-3-030-99527-0_5
9. Bellman, R.: Dynamic Programming, 1st edn. Princeton University Press, Princeton (1957)
10. Brázdil, T., Kučera, A., Novotný, P.: Determinacy in stochastic games with unbounded payoff functions. In: Kučera, A., Henzinger, T.A., Nešetřil, J., Vojnar, T., Antoš, D. (eds.) MEMICS 2012. LNCS, vol. 7721, pp. 94–105. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-36046-6_10
11. Chatterjee, K., Henzinger, T.A.: A survey of stochastic ω -regular games. *J. Comput. Syst. Sci.* **78**(2), 394–413 (2012). <https://doi.org/10.1016/j.jcss.2011.05.002>
12. Chen, T., Forejt, V., Kwiatkowska, M.Z., Parker, D., Simaitis, A.: Automatic verification of competitive stochastic systems. *Formal Methods Syst. Des.* **43**(1), 61–92 (2013). <https://doi.org/10.1007/s10703-013-0183-7>
13. Condon, A.: On algorithms for simple stochastic games. In: Cai, J. (ed.) Advances in Computational Complexity Theory, Proceedings of a DIMACS Workshop. DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 13, pp. 51–71. DIMACS/AMS (1990)
14. Condon, A.: The complexity of stochastic games. *Inf. Comput.* **96**(2), 203–224 (1992). [https://doi.org/10.1016/0890-5401\(92\)90048-K](https://doi.org/10.1016/0890-5401(92)90048-K)

15. Davey, B.A., Priestley, H.A.: *Introduction to Lattices and Order*. Cambridge University Press, Cambridge (1990)
16. D'Ippolito, N., Braberman, V.A., Piterman, N., Uchitel, S.: Synthesis of live behaviour models for fallible domains. In: Taylor, R.N., Gall, H.C., Medvidovic, N. (eds.) *Proceedings of the 33rd International Conference on Software Engineering, ICSE 2011*. pp. 211–220. ACM (2011). <https://doi.org/10.1145/1985793.1985823>
17. Feng, L., Wiltsche, C., Humphrey, L.R., Topcu, U.: Controller synthesis for autonomous systems interacting with human operators. In: Bayen, A.M., Branicky, M.S. (eds.) *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems, ICCPS 2015*, pp. 70–79. ACM (2015). <https://doi.org/10.1145/2735960.2735973>
18. Filar, J., Vrieze, K.: *Competitive Markov Decision Processes*. Springer, Heidelberg (1996). <https://doi.org/10.1007/978-1-4612-4054-9>
19. Haddad, S., Monmege, B.: Interval iteration algorithm for MDPs and IMDPs. *Theor. Comput. Sci.* **735**, 111–131 (2018). <https://doi.org/10.1016/j.tcs.2016.12.003>
20. Kallenberg, L.: *Linear Programming and Finite Markovian Control Problems*. Mathematisch Centrum, Amsterdam (1983)
21. Kučera, A.: Turn-based stochastic games. In: Apt, K.R., Grädel, E. (eds.) *Lectures in Game Theory for Computer Scientists*, pp. 146–184. Cambridge University Press (2011). <https://doi.org/10.1017/CBO9780511973468.006>
22. Kwiatkowska, M., Norman, G., Parker, D., Santos, G.: PRISM-games 3.0: stochastic game verification with concurrency, equilibria and time. In: Lahiri, S.K., Wang, C. (eds.) *CAV 2020*. LNCS, vol. 12225, pp. 475–487. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-53291-8_25
23. Kwiatkowska, M., Norman, G., Parker, D.: PRISM 4.0: verification of probabilistic real-time systems. In: Gopalakrishnan, G., Qadeer, S. (eds.) *CAV 2011*. LNCS, vol. 6806, pp. 585–591. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-22110-1_47
24. Martin, D.A.: The determinacy of Blackwell games. *J. Symb. Log.* **63**(4), 1565–1581 (1998). <https://doi.org/10.2307/2586667>
25. Morgenstern, O., von Neumann, J.: *Theory of Games and Economic Behavior*, 1st edn. Princeton University Press (1942)
26. Patek, S.D., Bertsekas, D.P.: Stochastic shortest path games. *SIAM J. Control Optimiz.* **37**, 804–824 (1999)
27. Shapley, L.: Stochastic games. *Proc. Natl. Acad. Sci.* **39**(10), 1095–1100 (1953). <https://doi.org/10.1073/pnas.39.10.1095>
28. Svorenová, M., Kwiatkowska, M.: Quantitative verification and strategy synthesis for stochastic games. *Eur. J. Control* **30**, 15–30 (2016). <https://doi.org/10.1016/j.ejcon.2016.04.009>

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

