Termination Criteria for Solving Concurrent Safety and Reachability Games [*]

Krishnendu Chatterjee[†] Luca de Alfaro[‡] Thomas A. Henzinger[§]

Abstract

We consider concurrent games played on graphs. At every round of a game, each player simultaneously and independently selects a move; the moves jointly determine the transition to a successor state. Two basic objectives are the safety objective to stay forever in a given set of states, and its dual, the reachability objective to reach a given set of states. We present in this paper a strategy improvement algorithm for computing the *value* of a concurrent safety game, that is, the maximal probability with which player 1 can enforce the safety objective. The algorithm yields a sequence of player-1 strategies which ensure probabilities of winning that converge monotonically to the value of the safety game.

Our result is significant because the strategy improvement algorithm provides, for the first time, a way to approximate the value of a concurrent safety game *from below.* Since a value iteration algorithm, or a strategy improvement algorithm for reachability games, can be used to approximate the same value from above, the combination of both algorithms yields a method for computing a converging sequence of upper and lower bounds for the values of concurrent reachability and safety games. Previous methods could approximate the values of these games only from one direction, and as no rates of convergence are known, they did not provide a practical way to solve these games.

1 Introduction

We consider games played between two players on graphs. At every round of the game, each of the two players selects a move; the moves of the players then determine the transition to the successor state. A play of the game gives rise to a path in the graph. We consider the two basic objectives for the players: *reachability* and *safety.* The reachability goal asks player 1 to reach a given set of target states or, if randomization is needed to play the game, to maximize the probability of reaching the target set. The safety goal asks player 2 to ensure that a given set of safe states is never left or, if randomization is required, to minimize the probability of leaving the safe set. The two objectives are dual, and the games are determined: the maximal probability with which player 1 can reach the target set is equal to one minus the maximal probability with which player 2 can confine the game to the complement of the target set [15].

These games on graphs can be divided into two classes: *turn-based* and *concurrent.* In turn-based games, only one player has a choice of moves at each state; in concurrent games, at each state both players choose a move, simultaneously and independently, from a set of available moves. For turn-based games, the solution of games with reachability and safety objectives has long been known. If each move determines a unique successor state, then the games are PTIME-complete and can be solved in linear time in the size of the game graph. If, more generally, each move determines a probability distribution on possible successor states, then the problem of deciding whether a turn-based game can be won with probability greater than a given threshold $p \in [0, 1]$ is in NP $\cap$ co-NP [5], and the exact value of the game can be computed by a strategy improvement algorithm [6], which works well in practice. These results all depend on the fact that in turn-based reachability and safety games, both players have optimal deterministic (i.e., no randomization is required), memoryless strategies. These strategies are functions from states to moves, so they are finite in number, and this guarantees the termination of the strategy improvement algorithm.

[*] A fuller version of the paper with proofs available at [3].

[†] CE, UC Santa Cruz, email: c_krish@eecs.berkeley.edu.

[‡] CE, UC Santa Cruz, email: luca@soe.ucsc.edu.

[§] EECS, UC Berkeley, and EPFL, Switzerland, email: tah@eecs.berkeley.edu.

The situation is very different for concurrent games, where randomization is required even in the special case in which the transition function is deterministic. The player-1 *value* of the game is defined, as usual, as the sup-inf value: the supremum, over all strategies of player 1, of the infimum, over all strategies of player 2, of the probability of achieving the reachability or safety goal. In concurrent reachability games, player 1 is guaranteed only the existence of ε-optimal strategies, which ensure that the value of the game is achieved within a specified tolerance $\varepsilon > 0$ [14]. Moreover, while these strategies (which depend on ε) are memoryless, in general they require randomization [8]. For player 2 (the safety player), *optimal* memoryless strategies exist [9], which again require randomization. All of these strategies are functions from states to probability distributions on moves. The question of deciding whether a concurrent reachability or safety game has a value at least $p \in [0, 1]$ is in PSPACE; this is shown by reduction to the theory of the real-closed fields [11], but no practical algorithms were known.

To summarize: while practical strategy improvement algorithms are available for turn-based reachability and safety games, so far no practical algorithms or even approximation schemes were known for concurrent games. If one wanted to compute the value of a concurrent game within a specified tolerance $\varepsilon > 0$, one was reduced to using a binary search algorithm that approximates the value by iterating queries in the theory of the real-closed fields. Strategy improvement and value iteration schemes were known for such games, but they could be used to approximate the value from one direction only, for reachability goals from below, and for safety goals from above [9, 2]. Neither scheme is guaranteed to terminate. Worse, since no convergence rates are known for these schemes, they provide no termination criteria for approximating a value within ε.

In this paper, we present for the first time a strategy improvement scheme that approximates the value of a concurrent safety game *from below.* Strategy improvement algorithms are generally practical, and together with the known strategy improvement scheme, or the value iteration scheme, to approximate the value of such a game from above, we obtain a termination criterion for computing the value of concurrent reachability and safety games within any given tolerance $\varepsilon > 0$.

Several difficulties had to be overcome in developing our scheme. First, while the strategy improvement algorithm that approximates reachability values from below [2] is based on locally improving a strategy on the basis of the valuation it yields, this approach does not suffice for approximating safety values from below: we would obtain an increasing sequence of values, but they would not necessarily converge to the value of the game (see Example 1). Rather, we introduce a novel, non-local improvement step, which augments the standard valuation-based improvement step. Each non-local step involves the solution of an appropriately constructed turn-based game. Second, as value iteration for safety objectives converges from above, while our sequences of strategies yield values that converge from below, the proof of convergence for our algorithm cannot be derived from a connection with value iteration, as was the case for reachability objectives. We had to develop new proof techniques both to show the monotonicity of the strategy values produced by our algorithm, and to show their convergence to the value of the game.

We also present a detailed analysis of termination criteria for turn-based stochastic games. Our analysis is based on (a) the strategy improvement algorithm for reachability games, and (b) on the bound of the precision of values for turn-based stochastic games. As a consequence of our analysis, we obtain an improved upper bound for termination for turn-based stochastic games.

2 Definitions

Notation. For a countable set A, a *probability distribution* on A is a function $\delta : A \to [0, 1]$ such that $\sum_{a \in A} \delta(a) = 1$. We denote the set of probability distributions on A by $\mathcal{D}(A)$. Given a distribution $\delta \in \mathcal{D}(A)$, we denote by $Supp(\delta) = \{x \in A \mid \delta(x) > 0\}$ the support set of δ.

Definition 1. (Concurrent Games) A (two-player) *concurrent game structure* $G = \langle S, M, \Gamma_1, \Gamma_2, \delta\rangle$ consists of the following components:

- A finite state space S and a finite set M of moves or actions.

- Two move assignments $\Gamma_1, \Gamma_2 : S \to 2^M \setminus \emptyset$. For $i \in \{1, 2\}$, assignment Γ_i associates with each state $s \in S$ a nonempty set $\Gamma_i(s) \subseteq M$ of moves available to player i at state s.

- A probabilistic transition function $\delta : S \times M \times M \to \mathcal{D}(S)$ that gives the probability $\delta(s, a_1, a_2)(t)$ of a transition from s to t when player 1 chooses at state s move a_1 and player 2 chooses move a_2, for all $s, t \in S$ and $a_1 \in \Gamma_1(s)$, $a_2 \in \Gamma_2(s)$.

We denote by $|\delta|$ the size of transition function, i.e., $|\delta| = \sum_{s \in S, a \in \Gamma_1(s), b \in \Gamma_2(s), t \in S} |\delta(s, a, b)(t)|$, where $|\delta(s, a, b)(t)|$ is the number of bits required to specify the transition probability $\delta(s, a, b)(t)$. We denote by $|G|$ the size of the game graph, and $|G| = |\delta| + |S|$. At every state $s \in S$, player 1 chooses a move $a_1 \in \Gamma_1(s)$, and simultaneously and independently player 2 chooses a move $a_2 \in \Gamma_2(s)$. The game then proceeds to the successor state t with probability $\delta(s, a_1, a_2)(t)$, for all $t \in S$. A state s is an *absorbing state* if for all $a_1 \in \Gamma_1(s)$ and $a_2 \in \Gamma_2(s)$, we have $\delta(s, a_1, a_2)(s) = 1$. In other words, at an absorbing state s for all choices of moves of the two players, the successor state is always s.

Definition 2. (Turn-based stochastic games) A *turn-based stochastic game graph* ($2^1/_2$*-player game graph*) $G = \langle (S, E), (S_1, S_2, S_R), \delta \rangle$ consists of a finite directed graph (S, E), a partition $(S_1,\ S_2,\ S_R)$ of the finite set S of states, and a probabilistic transition function δ: $S_R \to \mathcal{D}(S)$, where $\mathcal{D}(S)$ denotes the set of probability distributions over the state space S. The states in S_1 are the *player*-1 states, where player 1 decides the successor state; the states in S_2 are the *player*-2 states, where player 2 decides the successor state; and the states in S_R are the *random or probabilistic* states, where the successor state is chosen according to the probabilistic transition function δ. We assume that for $s \in S_R$ and $t \in S$, we have $(s, t) \in E$ iff $\delta(s)(t) > 0$, and we often write $\delta(s, t)$ for $\delta(s)(t)$. For technical convenience we assume that every state in the graph (S, E) has at least one outgoing edge. For a state $s \in S$, we write $E(s)$ to denote the set $\{t \in S \mid (s, t) \in E\}$ of possible successors. We denote by $|\delta|$ the size of the transition function, i.e., $|\delta| = \sum_{s \in S_R, t \in S} |\delta(s)(t)|$, where $|\delta(s)(t)|$ is the number of bits required to specify the transition probability $\delta(s)(t)$. We denote by $|G|$ the size of the game graph, and $|G| = |\delta| + |S| + |E|$.

Plays. A *play* ω of G is an infinite sequence $\omega = \langle s_0, s_1, s_2, \ldots \rangle$ of states in S such that for all $k \geq 0$, there are moves $a_1^k \in \Gamma_1(s_k)$ and $a_2^k \in \Gamma_2(s_k)$ with $\delta(s_k, a_1^k, a_2^k)(s_{k+1}) > 0$. We denote by Ω the set of all plays, and by Ω_s the set of all plays $\omega = \langle s_0, s_1, s_2, \ldots \rangle$ such that $s_0 = s$, that is, the set of plays starting from state s.

Selectors and strategies. A *selector* ξ for player $i \in \{1, 2\}$ is a function $\xi : S \to \mathcal{D}(M)$ such that for all states $s \in S$ and moves $a \in M$, if $\xi(s)(a) > 0$, then $a \in \Gamma_i(s)$. A selector ξ for player i at a state s is a distribution over moves such that if $\xi(s)(a) > 0$, then $a \in \Gamma_i(s)$. We denote by Λ_i the set of all selectors for player $i \in \{1, 2\}$, and similarly, we denote by $\Lambda_i(s)$ the set of all selectors for player i at a state s. The selector ξ is *pure* if for every state $s \in S$, there is a move $a \in M$ such that $\xi(s)(a) = 1$. A *strategy* for player $i \in \{1, 2\}$ is a function $\pi : S^+ \to \mathcal{D}(M)$ that associates with every finite, nonempty sequence of states, representing the history of the play so far, a selector for player i; that is, for all $w \in S^*$ and $s \in S$, we have $\mathit{Supp}(\pi(w \cdot s)) \subseteq \Gamma_i(s)$. The strategy π is *pure* if it always chooses a pure selector; that is, for all $w \in S^+$, there is a move $a \in M$ such that $\pi(w)(a) = 1$. A *memoryless* strategy is independent of the history of the play and depends only on the current state. Memoryless strategies correspond to selectors; we write $\overline{\xi}$ for the memoryless strategy consisting in playing forever the selector ξ. A strategy is *pure memoryless* if it is both pure and memoryless. In a turn-based stochastic game, a strategy for player 1 is a function $\pi_1 : S^* \cdot S_1 \to \mathcal{D}(S)$, such that for all $w \in S^*$ and for all $s \in S_1$ we have $\mathit{Supp}(\pi_1(w \cdot s)) \subseteq E(s)$. Memoryless strategies and pure memoryless strategies are obtained as the restriction of strategies as in the case of concurrent game graphs. The family of strategies for player 2 are defined analogously. We denote by Π_1 and Π_2 the sets of all strategies for player 1 and player 2, respectively. We denote by Π_i^M and Π_i^{PM} the sets of memoryless strategies and pure memoryless strategies for player i, respectively.

Destinations of moves and selectors. For all states $s \in S$ and moves $a_1 \in \Gamma_1(s)$ and $a_2 \in \Gamma_2(s)$, we indicate by $\mathit{Dest}(s, a_1, a_2) = \mathit{Supp}(\delta(s, a_1, a_2))$ the set of possible successors of s when the moves a_1 and a_2 are chosen. Given a state s, and selectors ξ_1 and ξ_2 for the two players, we denote by

$$\mathit{Dest}(s, \xi_1, \xi_2) = \bigcup_{\substack{a_1 \in \mathit{Supp}(\xi_1(s)),\\ a_2 \in \mathit{Supp}(\xi_2(s))}} \mathit{Dest}(s, a_1, a_2)$$

the set of possible successors of s with respect to the selectors ξ_1 and ξ_2.

Once a starting state s and strategies π_1 and π_2 for the two players are fixed, the game is reduced to an ordinary stochastic process. Hence, the probabilities of events are uniquely defined, where an *event* $\mathcal{A} \subseteq \Omega_s$ is a measurable set of plays. For an event $\mathcal{A} \subseteq \Omega_s$, we denote by $\Pr_s^{\pi_1,\pi_2}(\mathcal{A})$ the probability that a play belongs to $\mathcal{A}$ when the game starts from s and the players follow the strategies π_1 and π_2. Similarly, for a measurable function $f : \Omega_s \to \mathrm{I\!R}$, we denote by $\mathrm{E}_s^{\pi_1,\pi_2}(f)$ the expected value of f when the game starts from s and the players follow the strategies π_1 and π_2. For $i \geq 0$, we denote by $\Theta_i : \Omega \to S$ the random variable denoting the i-th state along a play.

Valuations. A *valuation* is a mapping $v : S \to [0, 1]$ associating a real number $v(s) \in [0, 1]$ with each state s. Given two valuations $v, w : S \to \mathrm{I\!R}$, we write $v \leq w$ when $v(s) \leq w(s)$ for all states $s \in S$. For an event $\mathcal{A}$, we denote by $\Pr^{\pi_1,\pi_2}(\mathcal{A})$ the valuation $S \to [0, 1]$ defined for all states $s \in S$ by $\big(\Pr^{\pi_1,\pi_2}(\mathcal{A})\big)(s) = \Pr_s^{\pi_1,\pi_2}(\mathcal{A})$. Similarly, for a measurable function $f : \Omega_s \to [0, 1]$, we denote by $\mathrm{E}^{\pi_1,\pi_2}(f)$ the valuation $S \to [0, 1]$ defined for all $s \in S$ by $\big(\mathrm{E}^{\pi_1,\pi_2}(f)\big)(s) = \mathrm{E}_s^{\pi_1,\pi_2}(f)$.

Reachability and safety objectives. Given a set $F \subseteq S$ of *safe* states, the objective of a safety game consists in never leaving F. Therefore, we define the set of winning plays as the set $\mathrm{Safe}(F) = \{\langle s_0, s_1, s_2, \ldots \rangle \in \Omega \mid s_k \in F$ for all $k \geq 0\}$. Given a subset $T \subseteq S$ of *target* states, the objective of a reachability game consists in reaching T. Correspondingly, the set winning plays is $\mathrm{Reach}(T) = \{\langle s_0, s_1, s_2, \ldots \rangle \in \Omega \mid s_k \in T$ for some $k \geq 0\}$ of plays that visit T. For all $F \subseteq S$ and $T \subseteq S$, the sets $\mathrm{Safe}(F)$ and $\mathrm{Reach}(T)$ is measurable. An objective in general is a measurable set, and in this paper we would consider only reachability and safety objectives. For an objective Φ, the probability of satisfying Φ from a state $s \in S$ under strategies π_1 and π_2 for players 1 and 2, respectively, is $\mathrm{Pr}_s^{\pi_1,\pi_2}(\Phi)$. We define the *value* for player 1 of game with objective Φ from the state $s \in S$ as: $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}(\Phi)(s) = \sup_{\pi_1 \in \Pi_1} \inf_{\pi_2 \in \Pi_2} \mathrm{Pr}_s^{\pi_1,\pi_2}(\Phi)$; i.e., the value is the maximal probability with which player 1 can guarantee the satisfaction of Φ against all player 2 strategies. Given a player-1 strategy π_1, we use the notation $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\pi_1}(\Phi)(s) = \inf_{\pi_2 \in \Pi_2} \mathrm{Pr}_s^{\pi_1,\pi_2}(\Phi)$. A strategy π_1 for player 1 is *optimal* for an objective Φ if for all states $s \in S$, we have $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\pi_1}(\Phi)(s) = \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}(\Phi)(s)$. For $\varepsilon > 0$, a strategy π_1 for player 1 is *ε-optimal* if for all states $s \in S$, we have $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\pi_1}(\Phi)(s) \geq \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}(\Phi)(s) - \varepsilon$. The notion of values and optimal strategies for player 2 are defined analogously. Reachability and safety objectives are dual, i.e., we have $\mathrm{Reach}(T) = \Omega \setminus \mathrm{Safe}(S \setminus T)$. The quantitative determinacy result of [15] ensures that for all states $s \in S$, we have $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}(\mathrm{Safe}(F))(s) + \langle\!\langle 2 \rangle\!\rangle_{\mathsf{val}}(\mathrm{Reach}(S \setminus F))(s) = 1$.

Theorem 2.1. (Memoryless determinacy) *For all concurrent game graphs G, for all $F \subseteq S$ and all $T \subseteq S$, if $F = S \setminus T$, then the following assertions hold.*

1. *[12] Memoryless optimal strategies exist for the safety objective Safe(F).*

2. *[2, 11] For all $\varepsilon > 0$, memoryless ε-optimal strategies exist for the reachability objectives Reach(T).*

3. *[5] If G is a turn-based stochastic game graph, then pure memoryless optimal strategies exist for the reachability objective Reach(T) and the safety objectives Safe(F).*

3 Markov Decision Processes

We present some facts about one-player versions of concurrent stochastic games, known as *Markov decision processes* (MDPs) [10, 1]. For $i \in \{1, 2\}$, a *player-i MDP* (for short, i-MDP) is a concurrent game where, for all states $s \in S$, we have $|\Gamma_{3-i}(s)| = 1$. Given a concurrent game G, if we fix a memoryless strategy corresponding to selector ξ_1 for player 1, the game is equivalent to a 2-MDP G_{ξ_1} with the transition function $\delta_{\xi_1}(s, a_2)(t) = \sum_{a_1 \in \Gamma_1(s)} \delta(s, a_1, a_2)(t) \cdot \xi_1(s)(a_1)$, for all $s \in S$ and $a_2 \in \Gamma_2(s)$. Similarly, if we fix selectors ξ_1 and ξ_2 for both players in a concurrent game G, we obtain a Markov chain, which we denote by G_{ξ_1,ξ_2}.

MDPs with reachability objectives. Given a 2-MDP with a reachability objective $\mathrm{Reach}(T)$ for player 2, where $T \subseteq S$, the values can be obtained as the solution of a linear program [12]. The linear program has a variable $x(s)$ for all states $s \in S$. The objective function is $\min \sum_{s \in S} x(s)$ subject to the following constraints:

$$\begin{aligned} x(s) \geq \sum_{t \in S} x(t) \cdot \delta(s, a_2)(t) \quad & \text{for all} \quad s \in S \text{ and } a_2 \in \Gamma_2(s) \\ x(s) = 1 \quad & \text{for all} \quad s \in T \\ 0 \leq x(s) \leq 1 \quad & \text{for all} \quad s \in S \end{aligned}$$

The correctness of the above linear program to compute the values follows from [10, 12].

4 Strategy Improvement for Safety Games

In this section we present a strategy improvement algorithm for concurrent games with safety objectives. The algorithm will produce a sequence of selectors $\gamma_0, \gamma_1, \gamma_2, \ldots$ for player 1, such that:

1. for all $i \geq 0$, we have $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(\mathrm{Safe}(F)) \leq \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_{i+1}}(\mathrm{Safe}(F))$;

2. if there is $i \geq 0$ such that $\gamma_i = \gamma_{i+1}$, then $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(\mathrm{Safe}(F)) = \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}(\mathrm{Safe}(F))$; and

3. $\lim_{i \to \infty} \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(\mathrm{Safe}(F)) = \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}(\mathrm{Safe}(F))$.

Condition 1 guarantees that the algorithm computes a sequence of monotonically improving selectors. Condition 2 guarantees that if a selector cannot be improved, then it is optimal. Condition 3 guarantees that the value guaranteed by the selectors converges to the value of the game, or equivalently, that for all $\varepsilon > 0$, there is a number i of iterations such that the memoryless player-1 strategy $\overline{\gamma}_i$ is ε-optimal. Note that for concurrent safety games, there may be no $i \geq 0$ such that $\gamma_i = \gamma_{i+1}$, that is, the algorithm may fail to generate an optimal selector. This is because there are concurrent safety games such that the values are irrational [9]. We start with a few notations

The *Pre* operator and optimal selectors. Given a valuation v, and two selectors $\xi_1 \in \Lambda_1$ and $\xi_2 \in \Lambda_2$, we define the valuations $Pre_{\xi_1,\xi_2}(v)$, $Pre_{1:\xi_1}(v)$, and

$Pre_1(v)$ as follows, for all states $s \in S$:

$$\begin{aligned}
&Pre_{\xi_1,\xi_2}(v)(s) \\
&\quad = \sum_{a,b \in M} \sum_{t \in S} v(t) \cdot \delta(s,a,b)(t) \cdot \xi_1(s)(a) \cdot \xi_2(s)(b) \\
&Pre_{1:\xi_1}(v)(s) = \inf_{\xi_2 \in \Lambda_2} Pre_{\xi_1,\xi_2}(v)(s) \\
&Pre_1(v)(s) = \sup_{\xi_1 \in \Lambda_1} \inf_{\xi_2 \in \Lambda_2} Pre_{\xi_1,\xi_2}(v)(s)
\end{aligned}$$

Intuitively, $Pre_1(v)(s)$ is the greatest expectation of v that player 1 can guarantee at a successor state of s. Also note that given a valuation v, the computation of $Pre_1(v)$ reduces to the solution of a zero-sum one-shot matrix game, and can be solved by linear programming. Similarly, $Pre_{1:\xi_1}(v)(s)$ is the greatest expectation of v that player 1 can guarantee at a successor state of s by playing the selector ξ_1. Note that all of these operators on valuations are monotonic: for two valuations v, w, if $v \leq w$, then for all selectors $\xi_1 \in \Lambda_1$ and $\xi_2 \in \Lambda_2$, we have $Pre_{\xi_1,\xi_2}(v) \leq Pre_{\xi_1,\xi_2}(w)$, $Pre_{1:\xi_1}(v) \leq Pre_{1:\xi_1}(w)$, and $Pre_1(v) \leq Pre_1(w)$. Given a valuation v and a state s, we define by $\mathsf{OptSel}(v,s) = \{\xi_1 \in \Lambda_1(s) \mid Pre_{1:\xi_1}(v)(s) = Pre_1(v)(s)\}$ the set of optimal selectors for v at state s. For an optimal selector $\xi_1 \in \mathsf{OptSel}(v,s)$, we define the set of counter-optimal actions as follows: $\mathsf{CountOpt}(v,s,\xi_1) = \{b \in \Gamma_2(s) \mid Pre_{\xi_1,b}(v)(s) = Pre_1(v)(s)\}$. Observe that for $\xi_1 \in \mathsf{OptSel}(v,s)$, for all $b \in \Gamma_2(s) \setminus \mathsf{CountOpt}(v,s,\xi_1)$ we have $Pre_{\xi_1,b}(v)(s) > Pre_1(v)(s)$. We define the set of optimal selector support and the counter-optimal action set as follows:

$$\begin{aligned}
&\mathsf{OptSelCount}(v,s) \\
&= \{(A,B) \subseteq \Gamma_1(s) \times \Gamma_2(s) \mid \exists \xi_1 \in \Lambda_1(s). \\
&\qquad \xi_1 \in \mathsf{OptSel}(v,s) \ \wedge \ \mathit{Supp}(\xi_1) = A \\
&\qquad \wedge \ \mathsf{CountOpt}(v,s,\xi_1) = B\};
\end{aligned}$$

i.e., it consists of pairs (A,B) of actions of player 1 and player 2, such that there is an optimal selector ξ_1 with support A, and B is the set of counter-optimal actions to ξ_1.

Turn-based reduction. Given a concurrent game $G = \langle S, M, \Gamma_1, \Gamma_2, \delta \rangle$ and a valuation v we construct a turn-based stochastic game $\overline{G}_v = \langle (\overline{S}, \overline{E}), (\overline{S}_1, \overline{S}_2, \overline{S}_R), \overline{\delta} \rangle$ as follows:

1. The set of states is as follows:

$$\begin{aligned}
&\overline{S} = S \cup \\
&\{(s,A,B) \mid s \in S, (A,B) \in \mathsf{OptSelCount}(v,s)\} \cup \\
&\{(s,A,b) \mid s \in S, (A,B) \in \mathsf{OptSelCount}(v,s), b \in B\}.
\end{aligned}$$

2. The state space partition is as follows: $\overline{S}_1 = S$; $\overline{S}_2 = \{(s,A,B) \mid s \in S, (A,B) \in \mathsf{OptSelCount}(v,s)\}$; and $\overline{S}_R = \overline{S} \setminus (\overline{S}_1 \cup \overline{S}_2)$.

3. The set of edges is as follows:

$$\begin{aligned}
&\overline{E} \\
&= \{(s,(s,A,B)) \mid s \in S, (A,B) \in \mathsf{OptSelCount}(v,s)\} \\
&\cup \{((s,A,B),(s,A,b)) \mid b \in B\} \\
&\cup \{((s,A,b),t) \mid t \in \bigcup_{a \in A} \mathit{Dest}(s,a,b)\}.
\end{aligned}$$

4. The transition function $\overline{\delta}$ for all states in $\overline{S}_R$ is uniform over its successors.

Intuitively, the reduction is as follows. Given the valuation v, state s is a player 1 state where player 1 can select a pair (A,B) (and move to state (s,A,B)) with $A \subseteq \Gamma_1(s)$ and $B \subseteq \Gamma_2(s)$ such that there is an optimal selector ξ_1 with support exactly A and the set of counter-optimal actions to ξ_1 is the set B. From a player 2 state (s,A,B), player 2 can choose any action b from the set B, and move to state (s,A,b). A state (s,A,b) is a probabilistic state where all the states in $\bigcup_{a \in A} \mathit{Dest}(s,a,b)$ are chosen uniformly at random. Given a set $F \subseteq S$ we denote by $\overline{F} = F \cup \{(s,A,B) \in \overline{S} \mid s \in F\} \cup \{(s,A,b) \in \overline{S} \mid s \in F\}$. We refer to the above reduction as TB, i.e., $(\overline{G}_v, \overline{F}) = \mathsf{TB}(G,v,F)$.

Value class of a valuation. Given a valuation v and a real $0 \leq r \leq 1$, the *value class* $U_r(v)$ of value r is the set of states with valuation r, i.e., $U_r(v) = \{s \in S \mid v(s) = r\}$

Ordering of strategies. We now define the notion of ordering of strategies. Let G be a concurrent game and F be the set of safe states. Let $T = S \setminus F$. Given a concurrent game graph G with a safety objective $\mathrm{Safe}(F)$, the set of *almost-sure winning* states is the set of states s such that the value at s is 1, i.e., $W_1 = \{s \in S \mid \langle\langle 1 \rangle\rangle_{\mathsf{val}}(\mathrm{Safe}(F)) = 1\}$ is the set of almost-sure winning states. An optimal strategy from W_1 is referred as an almost-sure winning strategy. The set W_1 and an almost-sure winning strategy can be computed in linear time by the algorithm given in [7]. We assume without loss of generality that all states in $W_1 \cup T$ are absorbing. We define a preorder $\prec$ on the strategies for player 1 as follows: given two player 1 strategies π_1 and π_1', let $\pi_1 \prec \pi_1'$ if the following two conditions hold: (i) $\langle\langle 1 \rangle\rangle_{\mathsf{val}}^{\pi_1}(\mathrm{Safe}(F)) \leq \langle\langle 1 \rangle\rangle_{\mathsf{val}}^{\pi_1'}(\mathrm{Safe}(F))$; and (ii) $\langle\langle 1 \rangle\rangle_{\mathsf{val}}^{\pi_1}(\mathrm{Safe}(F))(s) < \langle\langle 1 \rangle\rangle_{\mathsf{val}}^{\pi_1'}(\mathrm{Safe}(F))(s)$ for some state $s \in S$. Furthermore, we write $\pi_1 \preceq \pi_1'$ if either $\pi_1 \prec \pi_1'$ or $\pi_1 = \pi_1'$. We first present an

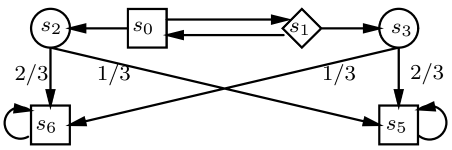

Figure 1: A turn-based stochastic safety game.

example that shows the improvements based only on Pre_1 operators are not sufficient for safety games, even on turn-based games and then present our algorithm.

Example 1. Consider the turn-based stochastic game shown in Fig 1, where the □ states are player 1 states, the ◇ states are player 2 states, and ○ states are random states with probabilities labeled on edges. The safety goal is to avoid the state s_6. Consider a memoryless strategy π_1 for player 1 that chooses the successor $s_0 \to s_2$, and the counter-strategy π_2 for player 2 chooses $s_1 \to s_0$. Given the strategies π_1 and π_2, the value at s_0, s_1 and s_2 is 1/3, and since all successors of s_0 have value 1/3, the value cannot be improved by Pre_1. However, note that if player 2 is restricted to choose only value optimal selectors for the value 1/3, then player 1 can switch to the strategy $s_0 \to s_1$ and ensure that the game stays in the value class 1/3 with probability 1. Hence switching to $s_0 \to s_1$ would force player 2 to select a counter-strategy that switches to the strategy $s_1 \to s_3$, and thus player 1 can get a value 2/3. ∎

Informal description of Algorithm 1. We now present the strategy improvement algorithm (Algorithm 1) for computing the values for all states in $S \setminus W_1$. The algorithm iteratively improves player-1 strategies according to the preorder $\prec$. The algorithm starts with the random selector $\gamma_0 = \overline{\xi}_1^{unif}$ that plays at all states all actions uniformly at random. At iteration $i+1$, the algorithm considers the memoryless player-1 strategy $\overline{\gamma}_i$ and computes the value $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(\text{Safe}(F))$. Observe that since $\overline{\gamma}_i$ is a memoryless strategy, the computation of $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(\text{Safe}(F))$ involves solving the 2-MDP G_{γ_i}. The valuation $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(\text{Safe}(F))$ is named v_i. For all states s such that $Pre_1(v_i)(s) > v_i(s)$, the memoryless strategy at s is modified to a selector that is value-optimal for v_i. The algorithm then proceeds to the next iteration. If $Pre_1(v_i) = v_i$, then the algorithm constructs the game $(\overline{G}_{v_i}, \overline{F}) = \mathsf{TB}(G, v_i, F)$, and computes $\overline{A}_i$ as the set of almost-sure winning states in $\overline{G}_{v_i}$ for the objective Safe$(\overline{F})$. Let $U = (\overline{A}_i \cap S) \setminus W_1$. If U is non-empty, then a selector γ_{i+1} is obtained at U from an pure memoryless optimal strategy (i.e., an almost-sure winning strategy) in $\overline{G}_{v_i}$, and the algorithm proceeds to iteration $i+1$. If $Pre_1(v_i) = v_i$ and U is empty, then the algorithm stops and returns the memoryless strategy $\overline{\gamma}_i$ for player 1. Unlike strategy improvement algorithms for turn-based games (see [6] for a survey), Algorithm 1 is not guaranteed to terminate, because the value of a safety game may not be rational. Proofs omitted due to lack of space are available in [3].

Lemma 4.1. *Let γ_i and γ_{i+1} be the player-1 selectors obtained at iterations i and $i+1$ of Algorithm 1. Let $I = \{s \in S \setminus (W_1 \cup T) \mid Pre_1(v_i)(s) > v_i(s)\}$. Let $v_i = \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(Safe(F))$ and $v_{i+1} = \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_{i+1}}(Safe(F))$. Then $v_{i+1}(s) \geq Pre_1(v_i)(s)$ for all states $s \in S$; and therefore $v_{i+1}(s) \geq v_i(s)$ for all states $s \in S$, and $v_{i+1}(s) > v_i(s)$ for all states $s \in I$.*

Recall that by Example 1 it follows that improvement by only step 3.2 is not sufficient to guarantee convergence to optimal values. Lemma 4.2 shows that step 3.3 also leads to an improvement. Finally, Theorem 4.2 shows that if improvements by step 3.2 and step 3.3 are not possible, then the optimal value and an optimal strategy is obtained.

Lemma 4.2. *Let γ_i and γ_{i+1} be the player-1 selectors obtained at iterations i and $i+1$ of Algorithm 1. Let $I = \{s \in S \setminus (W_1 \cup T) \mid Pre_1(v_i)(s) > v_i(s)\} = \emptyset$, and $(\overline{A}_i \cap S) \setminus W_1 \neq \emptyset$. Let $v_i = \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(Safe(F))$ and $v_{i+1} = \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_{i+1}}(Safe(F))$. Then $v_{i+1}(s) \geq v_i(s)$ for all states $s \in S$, and $v_{i+1}(s) > v_i(s)$ for some state $s \in (\overline{A}_i \cap S) \setminus W_1$.*

We obtain the following theorem from Lemma 4.1 and Lemma 4.2 that shows that the sequences of values we obtain is monotonically non-decreasing.

Theorem 4.1. (Monotonicity of values) *For $i \geq 0$, let γ_i and γ_{i+1} be the player-1 selectors obtained at iterations i and $i+1$ of Algorithm 1. If $\gamma_i \neq \gamma_{i+1}$, then $\langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(Safe(F)) < \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_{i+1}}(Safe(F))$.*

Theorem 4.2. (Optimality on termination) *For $i \geq 0$, let v_i be the valuation at iteration i of Algorithm 1 such that $v_i = \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(Safe(F))$. If $I = \{s \in S \setminus (W_1 \cup T) \mid Pre_1(v_i)(s) > v_i(s)\} = \emptyset$, and $(\overline{A}_i \cap S) \setminus W_1 = \emptyset$, then $\overline{\gamma}_i$ is an optimal strategy for player 1 for the objective Safe(F) and $v_i = \langle\!\langle 1 \rangle\!\rangle_{\mathsf{val}}(Safe(F))$.*

Convergence. We first observe that since pure memoryless optimal strategies exist for turn-based stochastic games with safety objectives (Theorem 2.1), for turn-based stochastic games it suffices to iterate over

Algorithm 1 Strategy-Improvement Algorithm for Safety Objective

```
Input: a concurrent game structure $G$ with safe set $F$.
Output: a strategy $\overline{\gamma}$ for player 1.
0. Compute $W_1 = \{s \in S \mid \langle\langle 1 \rangle\rangle_{\mathsf{val}}(\text{Safe}(F))(s) = 1\}$.
1. Let $\gamma_0 = \xi_1^{unif}$ and $i = 0$.
2. Compute $v_0 = \langle\langle 1 \rangle\rangle_{\mathsf{val}}^{\overline{\gamma}_0}(\text{Safe}(F))$.
3. do {
    3.1. Let $I = \{s \in S \setminus (W_1 \cup T) \mid Pre_1(v_i)(s) > v_i(s)\}$.
    3.2 if $I \neq \emptyset$, then
        3.2.1 Let $\xi_1$ be a player-1 selector such that for all states $s \in I$,
            we have $Pre_{1:\xi_1}(v_i)(s) = Pre_1(v_i)(s) > v_i(s)$.
        3.2.2 The player-1 selector $\gamma_{i+1}$ is defined as follows: for each state $s \in S$, let
            $\gamma_{i+1}(s) = \begin{cases} \gamma_i(s) & \text{if } s \notin I; \\ \xi_1(s) & \text{if } s \in I. \end{cases}$
    3.3 else
        3.3.1 let$(\overline{G}_{v_i}, \overline{F}) = \mathsf{TB}(G, v_i, F)$
        3.3.2 let $\overline{A}_i$ be the set of almost-sure winning states in $\overline{G}_{v_i}$ for Safe$(\overline{F})$ and
            $\overline{\pi}_1$ be a pure memoryless almost-sure winning strategy from the set $\overline{A}_i$.
        3.3.3 if $((\overline{A}_i \cap S) \setminus W_1 \neq \emptyset)$
            3.3.3.1 let $U = (\overline{A}_i \cap S) \setminus W_1$
            3.3.3.2 The player-1 selector $\gamma_{i+1}$ is defined as follows: for $s \in S$, let
                $\gamma_{i+1}(s) = \begin{cases} \gamma_i(s) & \text{if } s \notin U; \\ \xi_1(s) & \text{if } s \in U, \xi_1(s) \in \mathsf{OptSel}(v_i, s), \\ & \overline{\pi}_1(s) = (s, A, B), B = \mathsf{OptSelCount}(s, v, \xi_1). \end{cases}$
    3.4. Compute $v_{i+1} = \langle\langle 1 \rangle\rangle_{\mathsf{val}}^{\overline{\gamma}_{i+1}}(\text{Safe}(F))$.
    3.5. Let $i = i + 1$.
} until $I = \emptyset$ and $(\overline{A}_{i-1} \cap S) \setminus W_1 = \emptyset$.
4. return $\overline{\gamma}_i$.
```

pure memoryless selectors. Since the number of pure memoryless strategies is finite, it follows for turn-based stochastic games Algorithm 1 always terminates and yields an optimal strategy. For concurrent games, we will use the result that for $\varepsilon > 0$, there is a *k-uniform memoryless* strategy that achieves the value of a safety objective with in ε. We first define k-uniform memoryless strategies. For a positive integer $k > 0$, a selector ξ for player 1 is *k-uniform* if for all $s \in S \setminus (T \cup W_1)$ and all $a \in Supp(\pi_1(s))$ there exists $i, j \in \mathbb{N}$ such that $0 \leq i \leq j \leq k$ and $\xi(s)(a) = \frac{i}{j}$, i.e., the moves in the support are played with probability that are multiples of $\frac{1}{\ell}$ with $\ell \leq k$. A memoryless strategy is k-uniform if it is obtained from a k-uniform selector.

Lemma 4.3. *For all concurrent game graphs G, for all safety objectives Safe(F), for $F \subseteq S$, for all $\varepsilon > 0$, there exist $k > 0$ and k-uniform selectors ξ such that $\overline{\xi}$ is an ε-optimal strategy.*

Strategy improvement with k-uniform selectors. We first argue that if we restrict Algorithm 1 such that every iteration yields a k-uniform selector, for $k > 0$, then the algorithm terminates. For $k > 0$, the restriction of Algorithm 1 to k-uniform selectors means that instead of considering all possible selectors for player 1, the algorithm restricts player 1 to select among the k-uniform selectors. The basic argument that if Algorithm 1 is restricted to k-uniform selectors for player 1, for $k > 0$, then the algorithm terminates, follows from the fact that the number of k-uniform selectors for a given k is finite. A more formal argument is as follows: if we restrict player 1 to chose between k-uniform selectors, then a concurrent game graph G can be converted to a turn-based stochastic game graph, where player 1 first chooses a k-uniform selector, then player 2 chooses an action, and then the transition is determined by the chosen k-uniform selector of player 1, the action of player 2 and the transition function δ of the game graph G. Then by termination of turn-based stochastic games it follows that the algorithm will terminate. Given $k > 0$, let us denote by z_i^k the valuation of Algorithm 1 at iteration i, where the selectors for player 1 are restricted to be k-uniform. This gives us the following lemma.

LEMMA 4.4. *For all $k > 0$, there exists $i \geq 0$ such that $z_i^k = z_{i+1}^k$.*

LEMMA 4.5. *For all concurrent game graphs G, for all safety objectives $Safe(F)$, for $F \subseteq S$, for all $\varepsilon > 0$, there exist $k > 0$ and $i \geq 0$ such that for all $s \in S$ we have $z_i^k(s) \geq \langle\langle 1 \rangle\rangle_{\mathsf{val}}(Safe(F))(s) - \varepsilon$.*

THEOREM 4.3. (CONVERGENCE) *For $i \geq 0$, let v_i be the valuation obtained at iteration i of Algorithm 1. Then the following assertions hold.*

1. *For all $\varepsilon > 0$, there exists i such that for all s we have $v_i(s) \geq \langle\langle 1 \rangle\rangle_{\mathsf{val}}(Safe(F))(s) - \varepsilon$.*

2. *$\lim_{i \to \infty} v_i = \langle\langle 1 \rangle\rangle_{\mathsf{val}}(Safe(F))$.*

Complexity. Algorithm 1 may not terminate in general; we describe the complexity of each iteration. For a valuation v_i, the computation of $Pre_1(v_i)$ involves solution of matrix games with rewards v_i; this can be done in polynomial time using linear programming. Given v_i, if $Pre_1(v_i) = v_i$, the sets $\mathsf{OptSel}(v_i, s)$ and $\mathsf{OptSelCount}(v_i, s)$ can be computed by enumerating the subsets of available actions at s and then using linear-programming. For example, to check whether $(A, B) \in \mathsf{OptSelCount}(v_i, s)$ it suffices to check both of these facts:

1. *(A is the support of an optimal selector ξ_1).* there is an selector ξ_1 such that (i) ξ_1 is optimal (i.e. for all actions $b \in \Gamma_2(s)$ we have $Pre_{\xi_1, b}(v_i)(s) \geq v_i(s)$); (ii) for all $a \in A$ we have $\xi_1(a) > 0$, and for all $a \notin A$ we have $\xi_1(a) = 0$;

2. *(B is the set of counter-optimal actions against ξ_1).* for all $b \in B$ we have $Pre_{\xi_1, b}(v_i)(s) = v_i(s)$, and for all $b \notin B$ we have $Pre_{\xi_1, b}(v_i)(s) > v_i(s)$.

All the above checks can be performed by checking feasibility of sets of linear equalities and inequalities. Hence, $\mathsf{TB}(G, v_i, F)$ can be computed in time polynomial in size of G and v_i and exponential in the number of moves. We observe that the construction is exponential only in the number of moves at a state, and not in the number of states. The number of moves at a state is typically much smaller than the size of the state space. We also observe that the improvement step 3.3.2 requires the computation of the set of almost-sure winning states of a turn-based stochastic safety game: this can be done both via linear-time discrete graph-theoretic algorithms [4], and via symbolic algorithms [8]. Both of these methods are more efficient than the basic step 3.4 of the improvement algorithm, where the quantitative values of an MDP must be computed. Thus, the improvement step 3.3 of Algorithm 1 is in practice not inefficient, compared with the standard steps 3.2 and 3.4.

5 Termination Criteria

In this section we present termination criteria for strategy improvement algorithms for concurrent games for ε-approximation, and then present an improved termination condition for turn-based games.

Strategy improvement algorithm for reachability objectives. A strategy improvement algorithm for concurrent reachability games was presented in [2]. We refer to the strategy improvement algorithm of [2] as Algorithm 2. Algorithm 2 is simpler than Algorithm 1: it is similar to Algorithm 1 and in every iteration only Step 3.2 is executed (and Step 3.3 need not be executed).

Termination for concurrent games. We now present termination criteria for concurrent games. Applying Algorithm 2 (of [2]) for player 2, for a reachability objective Reach(T), we obtain a sequence of valuations $(u_i)_{i \geq 0}$ such that (a) $u_{i+1} \geq u_i$; (b) if $u_{i+1} = u_i$, then $u_i = \langle\langle 2 \rangle\rangle_{\mathsf{val}}(\text{Reach}(T))$; and (c) $\lim_{i \to \infty} u_i = \langle\langle 2 \rangle\rangle_{\mathsf{val}}(\text{Reach}(T))$. Given a concurrent game G with $F \subseteq S$ and $T = S \setminus F$, we apply Algorithm 2 to obtain the sequence of valuation $(u_i)_{i \geq 0}$ as above, and we apply Algorithm 1 to obtain a sequence of valuation $(v_i)_{i \geq 0}$. The termination criteria are as follows:

1. if for some i we have $u_{i+1} = u_i$, then we have $u_i = \langle\langle 2 \rangle\rangle_{\mathsf{val}}(\text{Reach}(T))$, and $1 - u_i = \langle\langle 1 \rangle\rangle_{\mathsf{val}}(\text{Safe}(F))$, and we obtain the values of the game;

2. if for some i we have $v_{i+1} = v_i$, then we have $1 - v_i = \langle\langle 2 \rangle\rangle_{\mathsf{val}}(\text{Reach}(T))$, and $v_i = \langle\langle 1 \rangle\rangle_{\mathsf{val}}(\text{Safe}(F))$, and we obtain the values of the game; and

3. for $\varepsilon > 0$, if for some $i \geq 0$, we have $u_i + v_i \geq 1 - \varepsilon$, then for all $s \in S$ we have $v_i(s) \geq \langle\langle 1 \rangle\rangle_{\mathsf{val}}(\text{Safe}(F))(s) - \varepsilon$ and $u_i(s) \geq \langle\langle 2 \rangle\rangle_{\mathsf{val}}(\text{Reach}(T))(s) - \varepsilon$ (i.e., the algorithm can stop for ε-approximation).

Observe that since $(u_i)_{i \geq 0}$ and $(v_i)_{i \geq 0}$ are both monotonically non-decreasing and $\langle\langle 1 \rangle\rangle_{\mathsf{val}}(\text{Safe}(F)) + \langle\langle 2 \rangle\rangle_{\mathsf{val}}(\text{Reach}(T)) = 1$, it follows that if $u_i + v_i \geq 1 - \varepsilon$, then forall $j \geq i$ we have $u_i \geq u_j - \varepsilon$ and $v_i \geq v_j - \varepsilon$. This establishes that $u_i \geq \langle\langle 1 \rangle\rangle_{\mathsf{val}}(\text{Safe}(F)) - \varepsilon$ and $v_i \geq \langle\langle 2 \rangle\rangle_{\mathsf{val}}(\text{Reach}(T)) - \varepsilon$; and the correctness of the stopping criteria (3) for ε-approximation follows. We also note that instead of applying Algorithm 2, a value-iteration algorithm can be applied for reachability games to obtain a sequence of valuation with properties similar to $(u_i)_{i\geq 0}$ and the above termination criteria can be applied.

THEOREM 5.1. *Let G be a concurrent game graph with a safety objective $Safe(F)$. Algorithm 1 and Algorithm 2 for player 2 for the reachability objective $Reach(S \setminus F)$ yield two sequences of valuations $(v_i)_{i\geq 0}$ and $(u_i)_{i\geq 0}$, respectively, such that (a) for all $i \geq 0$, we have $v_i \leq \langle\langle 1\rangle\rangle_{\mathsf{val}}(Safe(F)) \leq 1 - u_i$; and (b) $\lim_{i\to\infty} v_i = \lim_{i\to\infty} 1 - u_i = \langle\langle 1\rangle\rangle_{\mathsf{val}}(Safe(F))$.*

Termination for turn-based games. For turn-based stochastic games Algorithm 1 and as well as Algorithm 2 terminates. Each iteration of the Algorithm 2 of [2] is computable in polynomial time, and here we present a termination guarantee for Algorithm 2. To apply Algorithm 2 we assume the objective of player 1 to be a reachability objective Reach(T), and the correctness of the algorithm relies on the notion of *proper strategies.* Let $W_2 = \{s \in S \mid \langle\langle 1\rangle\rangle_{\mathsf{val}}(\text{Reach}(T))(s) = 0\}$. Then the notion of proper strategies and its properties are as follows.

DEFINITION 3. (PROPER STRATEGIES AND SELECTORS) A player-1 strategy π_1 is *proper* if for all player-2 strategies π_2, and for all states $s \in S\setminus(T\cup W_2)$, we have $\Pr_s^{\pi_1,\pi_2}(\text{Reach}(T \cup W_2)) = 1$. A player-1 selector ξ_1 is *proper* if the memoryless player-1 strategy $\overline{\xi}_1$ is proper.

LEMMA 5.1. ([2]) *Given a selector ξ_1 for player 1, the memoryless player-1 strategy $\overline{\xi}_1$ is proper iff for every pure selector ξ_2 for player 2, and for all states $s \in S$, we have $\Pr_s^{\overline{\xi}_1,\overline{\xi}_2}(Reach(T \cup W_2)) = 1$.*

LEMMA 5.2. *Let G be a turn-based stochastic game with reachability objective $Reach(T)$ for player 1. Let γ_0 be the initial selector, and γ_i be the selector obtained at iteration i of Algorithm 2. If γ_i is a pure, proper selector, then the following assertions hold: (a) for all $i \geq 0$, we have γ_i is a pure, proper selector; (b) for all $i \geq 0$, we have $u_{i+1} \geq u_i$, where $u_i = \langle\langle 1\rangle\rangle_{\mathsf{val}}^{\overline{\gamma}_i}(Reach(T))$ and $u_{i+1} = \langle\langle 1\rangle\rangle_{\mathsf{val}}^{\overline{\gamma}_{i+1}}(Reach(T))$; and (c) if $u_{i+1} = u_i$, then $u_i = \langle\langle 1\rangle\rangle_{\mathsf{val}}(Reach(T))$, and there exists i such that $u_{i+1} = u_i$.*

The above result follows from the result of [2] specialized for the case of turn-based stochastic games. The strategy improvement algorithm of Condon [6] works only for *halting games*, but Algorithm 2 works if we start with a pure, proper selector for reachability games that are not halting. Hence to use Algorithm 2 to compute values we need to start with a pure, proper selector. We present a procedure to compute a pure, proper selector, and then present termination bounds (i.e., bounds on i such that $u_{i+1} = u_i$). The construction of pure, proper selector is based on the notion of *attractors* defined below.

Attractor strategy. Let $A_0 = W_2 \cup T$, and for $i \geq 0$ we have $A_{i+1} = A_i \cup \{s \in S_1 \cup S_R \mid E(s) \cap A_i \neq \emptyset\} \cup \{s \in S_2 \mid E(s) \subseteq A_i\}$. Since for all $s \in S \setminus W_2$ we have $\langle\langle 1\rangle\rangle_{\mathsf{val}}(\text{Reach}(T)) > 0$, it follows that from all states in $S \setminus W_2$ player 1 can ensure that T is reached with positive probability. It follows that for some $i \geq 0$ we have $A_i = S$. The pure *attractor* selector ξ^* is as follows: for a state $s \in (A_{i+1} \setminus A_i) \cap S_1$ we have $\xi^*(s)(t) = 1$, where $t \in A_i$ (such a t exists by construction). The pure memoryless strategy $\overline{\xi^*}$ ensures that for all $i \geq 0$, from A_{i+1} the game reaches A_i with positive probability. Hence there is no end-component C contained in $S\setminus(W_2 \cup T)$ in the MDP $G_{\overline{\xi^*}}$. It follows that ξ^* is a pure selector that is proper, and the selector ξ^* can be computed in $O(|E|)$ time. This completes Algorithm 2 for turn-based stochastic games. We now present the termination bounds.

Termination bounds. We present termination bounds for binary turn-based stochastic games. A turn-based stochastic game is binary if for all $s \in S_R$ we have $|E(s)| \leq 2$, and for all $s \in S_R$ if $|E(s)| = 2$, then for all $t \in E(s)$ we have $\delta(s)(t) = \frac{1}{2}$, i.e., for all probabilistic states there are at most two successors and the transition function δ is uniform.

LEMMA 5.3. *Let G be a binary Markov chain with $|S|$ states with a reachability objective $Reach(T)$. Then for all $s \in S$ we have $\langle\langle 1\rangle\rangle_{\mathsf{val}}(Reach(T)) = \frac{p}{q}$, with $p, q \in \mathbb{N}$ and $p, q \leq 4^{|S|-1}$.*

LEMMA 5.4. *Let G be a binary turn-based stochastic game with a reachability objective $Reach(T)$. Then for all $s \in S$ we have $\langle\langle 1\rangle\rangle_{\mathsf{val}}(Reach(T)) = \frac{p}{q}$, with $p, q \in \mathbb{N}$ and $p, q \leq 4^{|S_R|-1}$.*

From Lemma 5.4 it follows that at iteration i of Algorithm 2 either the sum of the values either increases by $\frac{1}{4^{|S_R|-1}}$ or else there is a valuation u_i such that $u_{i+1} = u_i$. Since the sum of values of all states can be at most $|S|$, it follows that algorithm terminates in at most $|S| \cdot 4^{|S_R|-1}$ steps. Moreover, since the number of pure memoryless strategies is at most $\prod_{s\in S_1} |E(s)|$, the algorithm terminates in at most $\prod_{s\in S_1} |E(s)|$ steps. It follows from the results of [16] that a turn-based stochastic game graph G can be reduced to a equivalent binary turn-based stochastic game graph G' such that the set of player 1 and player 2 states in G and G' are the same and the number of probabilistic states in G' is

$O(|\delta|)$, where $|\delta|$ is the size of the transition function in G. Thus we obtain the following result.

THEOREM 5.2. *Let $G = \langle (S,E), (S_1, S_2, S_R), \delta \rangle$ be a turn-based stochastic game with a reachability objective Reach(T). Algorithm 2 computes the values in G in time $O\big(\min\{\prod_{s \in S_1} |E(s)|, 2^{O(|\delta|)}\} \cdot poly(|G|)\big)$; where poly is polynomial function.*

The results of [13] presented an algorithm for turn-based stochastic games that works in time $O(|S_R|! \cdot poly(|G|))$. The algorithm of [13] works only for turn-based stochastic games, for general turn-based stochastic games the complexity of the algorithm of [13] is better. However, for turn-based stochastic games where the transition function at all states can expressed in constant bits we have $|\delta| = O(|S_R|)$. In these cases the reachability strategy improvement algorithm (that works for both concurrent and turn-based stochastic games) works in time $2^{O(|S_R|)} \cdot poly(|G|)$ as compared to the time $2^{O(|S_R| \cdot \log(|S_R|)} \cdot poly(|G|)$ of the algorithm of [13].

Acknowledgements. This research was supported in part by the NSF grants CCR-0132780, CNS-0720884, and CCR-0225610, and by the Swiss National Science Foundation. We thank the anonymous referees for useful comments that helped us improve the proofs and the paper.

References

[1] D.P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995. Volumes I and II.

[2] K. Chatterjee, L. de Alfaro, and T.A. Henzinger. Strategy improvement in concurrent reachability games. In *QEST'06: Quantitative Evaluation of Systems*. IEEE Computer Society Press, 2006.

[3] K. Chatterjee, L. de Alfaro, and T.A. Henzinger. Termination criteria for solving concurrent safety and reachability games. *CoRR*, abs/0809.4017, 2008.

[4] K. Chatterjee, M. Jurdziński, and T.A. Henzinger. Simple stochastic parity games. In *CSL'03: Computer Science Logic*, volume 2803 of *LNCS*, pages 100–113. Springer, 2003.

[5] A. Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992.

[6] A. Condon. On algorithms for simple stochastic games. In *Advances in Computational Complexity Theory*, volume 13 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 51–73. American Mathematical Society, 1993.

[7] L. de Alfaro and T.A. Henzinger. Concurrent omega-regular games. In *LICS'00: Symposium on Logic in Computer Science*, pages 141–154. IEEE Computer Society Press, 2000.

[8] L. de Alfaro, T.A. Henzinger, and O. Kupferman. Concurrent reachability games. *Theoretical Computer Science*, 386(3):188–217, 2007.

[9] L. de Alfaro and R. Majumdar. Quantitative solution of omega-regular games. *Journal of Computer and System Sciences*, 68:374–397, 2004.

[10] C. Derman. *Finite State Markovian Decision Processes*. Academic Press, 1970.

[11] K. Etessami and M. Yannakakis. Recursive concurrent stochastic games. In *ICALP 06: Automata, Languages, and Programming*, volume 4052 of *LNCS*, pages 324–335. Springer, 2006.

[12] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.

[13] H. Gimbert and F. Horn. Simple stochastic games with few random vertices are easy to solve. In *FoSSaCS'08: Foundations of Software Science and Computational Structures*, volume 4962 of *LNCS*, pages 5–19, 2008.

[14] P.R. Kumar and T.H. Shiau. Existence of value and randomized strategies in zero-sum discrete-time stochastic dynamic games. *SIAM J. Control and Optimization*, 19(5):617–634, 1981.

[15] D.A. Martin. The determinacy of Blackwell games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998.

[16] U. Zwick and M.S. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.