A Survey of Stochastic Games with Limsup and Liminf Objectives*

Krishnendu Chatterjee[1], Laurent Doyen[2**], and Thomas A. Henzinger[3]

[1] Institute of Science and Technology (IST), Austria
[2] Université Libre de Bruxelles (ULB), Belgium
[3] École Polytechnique Fédérale de Lausanne (EPFL), Switzerland

Abstract. A stochastic game is a two-player game played on a graph, where in each state the successor is chosen either by one of the players, or according to a probability distribution. We survey stochastic games with limsup and liminf objectives. A real-valued reward is assigned to each state, and the value of an infinite path is the limsup (resp. liminf) of all rewards along the path. The value of a stochastic game is the maximal expected value of an infinite path that can be achieved by resolving the decisions of the first player. We present the complexity of computing values of stochastic games and their subclasses, and the complexity of optimal strategies in such games.

1 Introduction

A *turn-based stochastic game* is played on a finite graph with three types of states: in player-1 states, the first player chooses a successor state from a given set of outgoing edges; in player-2 states, the second player chooses a successor state from a given set of outgoing edges; and in probabilistic states, the successor state is chosen according to a given probability distribution. The game results in an infinite path through the graph. Every such path is assigned a real value, and the objective of player 1 is to resolve her choices so as to maximize the expected value of the resulting path, while the objective of player 2 is to minimize the expected value. If the function that assigns values to infinite paths is a Borel function (in the Cantor topology on infinite paths), then the game is determined [17]: the maximal expected value achievable by player 1 is equal to the minimal expected value achievable by player 2, and it is called the *value* of the game.

There are several canonical functions for assigning values to infinite paths. If each state is given a reward, then the *max* (resp. *min*) function chooses the

* This research was supported in part by the Swiss National Science Foundation under the Indo-Swiss Joint Research Programme, by the European Network of Excellence on Embedded Systems Design (ArtistDesign), by the European COMBEST project, Quasimodo, Gasics projects, by the PAI program Moves funded by the Belgian Federal Government, and by the CFV (Federated Center in Verification) funded by the F.R.S.-FNRS.

** Postdoctoral researcher of the F.R.S.-FNRS.

maximum (resp. minimum) of the infinitely many rewards along a path; the *limsup* (resp. *liminf*) function chooses the limsup (resp. liminf) of the infinitely many rewards; and the *limit-average* function chooses the long-run average of the rewards. The max and min functions are Borel level-1 functions, whereas limsup and liminf are Borel level-2 functions, and limit-average is a Borel level-3 function. Stochastic games with the limit-average condition (also called *mean-payoff* objective) have been studied extensively in the literature [11, 14, 20, 1, 15]. The study of stochastic games with max and min conditions [2, 4], as well as limsup and liminf conditions [5, 13, 16], is more recent. The max and min functions are natural generalizations of reachability and safety objectives in the non-quantitative setting, while the limsup and liminf functions are natural generalizations of Büchi and coBüchi objectives [18, 19].

In this paper, we survey algorithms and computational complexity results for computing values of turn-based stochastic games and with limsup and liminf objectives. We organize the results according to the different classes of game graphs. We successively consider (*i*) 1-player game graphs, where all states belong to one player, (*ii*) 2-player game graphs, in which there is no probabilistic state, (*iii*) $1^1\!/_2$-player game graphs (or Markov decision processes), in which there is no player-2 state, and (*iv*) $2^1\!/_2$-player game graphs, which is the general case. Along with surveying known results in the field, we also present two algorithmic improvements over the literature for the solution of 1-player and 2-player game graphs with limsup and liminf objectives. We show that 1-player game graphs with $\boldsymbol{n}$ states and $\boldsymbol{m}$ edges can be solved in time $O(\boldsymbol{n}+\boldsymbol{m})$ while the previously known algorithm of [2] runs in time $O(\boldsymbol{n}\log\boldsymbol{n}+\boldsymbol{m})$; for 2-player game graphs, our algorithm runs in time $O(\boldsymbol{mn}\log\boldsymbol{n})$ as compared to the previously known algorithm of [2] that runs in time $O(\boldsymbol{mn}^2)$.

2 Definitions

We consider the class of turn-based stochastic games and some of its subclasses.

Game graphs. A *turn-based probabilistic game graph* ($2^1\!/_2$*-player game graph*) $G = ((S, E), (S_1, S_2, S_P), \delta)$ consists of a finite directed graph (S, E), a partition (S_1, S_2, S_P) of the finite set S of states, and a probabilistic transition function δ: $S_P \to \mathcal{D}(S)$, 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_P are the *probabilistic* states, where the successor state is chosen according to the probabilistic transition function δ. We assume that for $s \in S_P$ 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.

Subclasses of stochastic games. The *turn-based deterministic game graphs* (*2-player game graphs*) are the special case of the $2^1\!/_2$-player game graphs with

$S_P = \emptyset$. The *Markov decision processes* ($1\frac{1}{2}$*-player game graphs*) are the special case of the $2\frac{1}{2}$-player game graphs with $S_1 = \emptyset$ or $S_2 = \emptyset$. We refer to the MDPs with $S_2 = \emptyset$ as *player*-1 MDPs, and to the MDPs with $S_1 = \emptyset$ as *player*-2 MDPs. The *transition systems* (1-*player game graphs*) are the special case of $2\frac{1}{2}$-player game graphs with (a) $S_P = \emptyset$ and (b) either $S_1 = \emptyset$ or $S_2 = \emptyset$. Observe that 1-player game graphs are subclasses of both 2-player game graphs and $1\frac{1}{2}$-player game graphs.

Size of graph. Given a game graph $G = ((S, E), (S_1, S_2, S_P), \delta)$ we use the following notations: (a) we denote by $\boldsymbol{n}$ the number of states, i.e., $\boldsymbol{n} = |S|$; (b) we denote by $\boldsymbol{m}$ the number of edges, i.e., $\boldsymbol{m} = |E|$; (c) we denote by $\varDelta$ the maximum out-degree of the graph, i.e., $\varDelta = \max_{s \in S} |E(s)|$.

Plays and strategies. An infinite path, or a *play*, of the game graph G is an infinite sequence $\omega = \langle s_0, s_1, s_2, \ldots \rangle$ of states such that $(s_k, s_{k+1}) \in E$ for all $k \in \mathbb{N}$. We write $\varOmega$ for the set of all plays, and for a state $s \in S$, we write $\varOmega_s \subseteq \varOmega$ for the set of plays that start from the state s. A *strategy* for player 1 is a function σ: $S^* \cdot S_1 \to \mathcal{D}(S)$ that assigns a probability distribution to all finite sequences $w \in S^* \cdot S_1$ of states ending in a player-1 state (the sequence w represents a prefix of a play). Player 1 follows the strategy σ if in each player-1 move, given that the current history of the game is $w \in S^* \cdot S_1$, she chooses the next state according to the probability distribution $\sigma(w)$. A strategy must prescribe only available moves, i.e., for all $w \in S^*$, $s \in S_1$, and $t \in S$, if $\sigma(w \cdot s)(t) > 0$, then $(s, t) \in E$. The strategies for player 2 are defined analogously. We denote by $\varSigma$ and $\varPi$ the set of all strategies for player 1 and player 2, respectively.

Once a starting state $s \in S$ and strategies $\sigma \in \varSigma$ and $\pi \in \varPi$ for the two players are fixed, the outcome of the game is a random walk $\omega_s^{\sigma,\pi}$ for which the probabilities of events are uniquely defined, where an *event* $\mathcal{A} \subseteq \varOmega$ is a measurable set of plays. For a state $s \in S$ and an event $\mathcal{A} \subseteq \varOmega$, we write $\Pr_s^{\sigma,\pi}(\mathcal{A})$ for the probability that a play belongs to $\mathcal{A}$ if the game starts from the state s and the players follow the strategies σ and π, respectively. For a measurable function $f : \varOmega \to \mathrm{I\!R}$ we denote by $\mathbb{E}_s^{\sigma,\pi}[f]$ the *expectation* of the function f under the probability measure $\Pr_s^{\sigma,\pi}(\cdot)$.

Strategies that do not use randomization are called pure. A player-1 strategy σ is *pure* if for all $w \in S^*$ and $s \in S_1$, there is a state $t \in S$ such that $\sigma(w \cdot s)(t) = 1$. A *memoryless* player-1 strategy does not depend on the history of the play but only on the current state; i.e., for all $w, w' \in S^*$ and for all $s \in S_1$ we have $\sigma(w \cdot s) = \sigma(w' \cdot s)$. A memoryless strategy for player 1 can be represented as a function σ: $S_1 \to \mathcal{D}(S)$. A *pure memoryless strategy* is a strategy that is both pure and memoryless. A pure memoryless strategy for player 1 can be represented as a function σ: $S_1 \to S$. We denote by $\varSigma^{PM}$ the set of pure memoryless strategies for player 1. The pure memoryless player-2 strategies $\varPi^{PM}$ are defined analogously.

Quantitative objectives. A *quantitative* objective is specified as a measurable function $f : \varOmega \to \mathrm{I\!R}$. We consider *zero-sum* games, i.e., games that are strictly competitive. In zero-sum games the objectives of the two players are functions

f and $-f$, respectively. We consider quantitative objectives specified as limsup and liminf objectives. These objectives are complete for the second levels of the Borel hierarchy: limsup objectives are Π_2-complete, and liminf objectives are Σ_2-complete. The definition of limsup and liminf objectives is as follows.

– *Limsup objectives.* Let $r : S \to \mathbb{R}$ be a real-valued reward function that assigns to every state s the reward $r(s)$. The limsup objective assigns to every play the maximum reward that appears infinitely often in the play. Formally, for a play $\omega = \langle s_1, s_2, s_3, \ldots \rangle$ we have
$$\text{limsup}(r)(\omega) = \limsup \langle r(s_i) \rangle_{i \geq 0} = \lim_{n \to \infty} \max\{\, r(s_i) \mid i \geq n \,\}.$$
– *Liminf objectives.* Let $r : S \to \mathbb{R}$ be a real-valued reward function that assigns to every state s the reward $r(s)$. The liminf objective assigns to every play the maximum reward v such that the rewards that appear eventually always in the play are at least v. Formally, for a play $\omega = \langle s_1, s_2, s_3, \ldots \rangle$ we have
$$\text{liminf}(r)(\omega) = \liminf \langle r(s_i) \rangle_{i \geq 0} = \lim_{n \to \infty} \min\{\, r(s_i) \mid i \geq n \,\}.$$
The limsup and liminf objectives are complementary in the sense that for all plays ω we have $\text{limsup}(r)(\omega) = -\text{liminf}(-r)(\omega)$. If the reward function r is boolean (that is rewards are only 0 and 1), then (a) the limsup objective correspond to the classical Büchi objective with the set of states with reward 1 as the set of Büchi states; and (b) the liminf objective correspond to the classical coBüchi objective with the set of states with reward 1 as the set of coBüchi states.

Values and optimal strategies. Given a game graph G and a measurable function $f : \Omega \to \mathbb{R}$ we define the *value* functions $\langle\!\langle 1 \rangle\!\rangle_{val}^G$ and $\langle\!\langle 2 \rangle\!\rangle_{val}^G$ for the players 1 and 2, respectively, as the following functions from the state space S to the set $\mathbb{R}$ of reals: for all states $s \in S$, let
$$\langle\!\langle 1 \rangle\!\rangle_{val}^G(f)(s) = \sup_{\sigma \in \Sigma} \inf_{\pi \in \Pi} \mathbb{E}_s^{\sigma,\pi}[f];$$
$$\langle\!\langle 2 \rangle\!\rangle_{val}^G(-f)(s) = \sup_{\pi \in \Pi} \inf_{\sigma \in \Sigma} \mathbb{E}_s^{\sigma,\pi}[-f].$$
In other words, the value $\langle\!\langle 1 \rangle\!\rangle_{val}^G(f)(s)$ gives the maximal expectation with which player 1 can achieve her objective f from state s, and analogously for player 2. The strategies that achieve the values are called optimal: a strategy σ for player 1 is *optimal* from the state s for f if $\langle\!\langle 1 \rangle\!\rangle_{val}^G(f)(s) = \inf_{\pi \in \Pi} \mathbb{E}_s^{\sigma,\pi}[f]$. The optimal strategies for player 2 are defined analogously. We now state the classical determinacy results for $2^1\!/_2$-player games with limsup and liminf objectives.

Theorem 1 (Quantitative determinacy). *Let $G = ((S, E), (S_1, S_2, S_P), \delta)$ be a $2^1\!/_2$-player game graph. For all reward functions $r : S \to \mathbb{R}$ and all states $s \in S$, we have*
$$\langle\!\langle 1 \rangle\!\rangle_{val}^G(\text{limsup}(r))(s) + \langle\!\langle 2 \rangle\!\rangle_{val}^G(\text{liminf}(-r))(s) = 0;$$

$$\langle\langle 1\rangle\rangle^G_{val}(\text{liminf}(r))(s) + \langle\langle 2\rangle\rangle^G_{val}(\text{limsup}(-r))(s) = 0.$$

The above results can be derived from the results in [16] or from the result of Martin [17].

3 Computational and Strategy Complexity

In this section we survey the computational complexity and the structural properties of optimal strategies in various subclasses of stochastic games. We organize our results for various classes of game graphs. The classical algorithmic solutions for stochastic games can be classified as (a) graph-theoretic algorithms or (b) value-iteration algorithms. We briefly discuss the general properties of the value-iteration algorithm and provide specific details of the algorithms for different classes of the game graphs later (in specific subsections).

Value-iteration algorithms and improvement functions. The values of stochastic games and their subclasses with limsup and liminf objectives can be characterized as fixpoint solution of certain nested fixpoint formulas. The characterization provides *symbolic* value-iteration algorithms to compute values by iterating certain *binary improvement functions* parametrized by a predecessor operator Pre that will be instantiated according to the different classes of game graphs. A *valuation* is a function $v\colon S \to \mathbb{R} \cup \{ -\infty, \infty \}$ that maps every state to a real number[1]. We write V for the set of valuations. A binary improvement function $\mathsf{Imp2}$ operates on pairs of valuations and needs to satisfy the following requirements.

Monotone For all valuation pairs $(v_1, u_1), (v_2, u_2)$, if $(v_1, u_1) \leq (v_2, u_2)$, then $\mathsf{Imp2}(v_1, u_1) \leq \mathsf{Imp2}(v_2, u_2)$ (the inequality $\leq$ is pointwise for valuations).

Continuous For every chain $C = \langle (v_0, u_0), (v_1, u_1), (v_2, u_2), \ldots \rangle$ of valuations, the sequence $\mathsf{Imp2}(C) = \langle \mathsf{Imp2}(v_0, u_0), \mathsf{Imp2}(v_1, u_1), \mathsf{Imp2}(v_2, u_2), \ldots \rangle$ is a chain of valuations by monotonicity of $\mathsf{Imp2}$. We require that $\mathsf{Imp2}(\lim C) = \lim \mathsf{Imp2}(C)$.

Directed Either $v \geq \mathsf{Imp2}(v, u) \geq u$ for all valuations v, u with $v \geq u$; or $v \leq \mathsf{Imp2}(v, u) \leq u$ for all real valuations v, u with $v \leq u$.

If the above requirements are satisfied, then we can invoke Kleene's fixpoint theorem for existence of fixpoints with the improvement functions. The binary improvement functions we consider also satisfy the following *locality* property: for all states $s \in S$ and all valuation pairs $(v_1, u_1), (v_2, u_2)$, if $v_1(s') = v_2(s')$ and $u_1(s') = u_2(s')$ for all successors $s' \in E(s)$, then $\mathsf{Imp2}(v_1, u_1)(s) = \mathsf{Imp2}(v_2, u_2)(s)$.

The description of improvement functions. Consider a reward function r, and the corresponding objectives $\text{limsup}(r)$ and $\text{liminf}(r)$. Given a function Pre:

[1] we add $-\infty$ and ∞ to the set of reals in the range of valuations so that the set V of valuations form a complete lattice

$V \to V$, we define the two parametric functions $\mathsf{limsupImp}[\mathsf{Pre}]\colon V \times V \to V$ and $\mathsf{liminfImp}[\mathsf{Pre}]\colon V \times V \to V$ by

$$\begin{aligned}\mathsf{limsupImp}[\mathsf{Pre}](v,u) &= \min\{\max\{r,u,\mathsf{Pre}(u)\},v,\max\{u,\mathsf{Pre}(v)\}\};\\ \mathsf{liminfImp}[\mathsf{Pre}](v,u) &= \max\{\min\{r,u,\mathsf{Pre}(u)\},v,\min\{u,\mathsf{Pre}(v)\}\};\end{aligned}$$

for all valuations $v, u \in V$ (the functions max and min are lifted from real values to valuations in a pointwise fashion). Observe that if $v \geq u$, then $v \geq \mathsf{limsupImp}[\mathsf{Pre}](v,u) \geq u$; and if $v \leq u$, then $v \leq \mathsf{liminfImp}[\mathsf{Pre}](v,u) \leq u$. Thus both $\mathsf{limsupImp}[\mathsf{Pre}]$ and $\mathsf{liminfImp}[\mathsf{Pre}]$ are directed. For different graph models, we will instantiate the parameter Pre differently. We remark that in all the cases that we will consider in this paper, we can simplify the above definitions of the binary improvement functions as follows:

$$\begin{aligned}\mathsf{limsupImp}[\mathsf{Pre}](v,u) &= \min\{\max\{r,u,\mathsf{Pre}(u)\},v,\mathsf{Pre}(v)\};\\ \mathsf{liminfImp}[\mathsf{Pre}](v,u) &= \max\{\min\{r,u,\mathsf{Pre}(u)\},v,\mathsf{Pre}(v)\};\end{aligned}$$

for all valuations $v, u \in V$. To see why the simplification is sound, let $u^{j+1} = \mathsf{limsupImp}[\mathsf{Pre}](v, u^j)$ (according to the original, unsimplified definition) for all $j \geq 0$. For all valuations $v \geq u^0$, if $\mathsf{Pre}(v) \geq u^0$, then for all $j \geq 0$, both $v \geq u^j$ and $\mathsf{Pre}(v) \geq u^j$, and therefore $u^{j+1} = \min\{\max\{p, u^j, \mathsf{Pre}(u^j)\}, v, \mathsf{Pre}(v)\}$. If $u^0(s) = \min_{t \in S} r(t)$ for all $s \in S$, then for all instantiations of Pre (that we will use) for all valuations $v \geq u^0$, we will have $\mathsf{Pre}(v) \geq u^0$, and thus the above simplification is sound. The case $\mathsf{liminfImp}[\mathsf{Pre}]$ and $u^0(s) = \max_{t \in S} r(t)$ for all $s \in S$ is symmetric. In some special cases of boolean reward functions r, the valuations can also be restricted to be functions from states to boolean (such as 2-player game graphs with boolean reward functions). In such cases, we can invoke Tarski-Knaster fixpoint theorem that requires only the monotonicity property. Then the improvement function can be further simplified as follows:

$$\begin{aligned}\mathsf{limsupImp}[\mathsf{Pre}](v,u) &= \min\{\max\{r,\mathsf{Pre}(u)\},\mathsf{Pre}(v)\};\\ \mathsf{liminfImp}[\mathsf{Pre}](v,u) &= \max\{\min\{r,\mathsf{Pre}(u)\},\mathsf{Pre}(v)\};\end{aligned}$$

The above description of the improvement functions does not satisfy the directed property. Also see [4] for a more detailed discussion about the properties of the fixpoint and the requirements of improvement functions.

Fixpoint characterization. Given the two parametric improvement functions, the value function of player 1 for limsup objective can be characterized as a nested fixpoint solution (nesting of a greatest fixpoint and a least fixpoint). In μ-calculus notation, let

$$v^{ls} = (\nu x)(\mu y)\ \mathsf{limsupImp}[\mathsf{Pre}](x,y). \tag{1}$$

Then for suitable instantiation Pre in $\mathsf{limsupImp}[\mathsf{Pre}]$ the valuation v^{ls} gives the value function for a stochastic game with limsup objective. Symmetrically, the value function for liminf objective can also be characterized as a nested fixpoint solution (nesting of a least fixpoint and a greatest fixpoint). In μ-calculus notation, let

$$v^{li} = (\mu x)(\nu y)\ \mathsf{liminfImp}[\mathsf{Pre}](x,y). \tag{2}$$

Then for suitable instantiation Pre in $\mathsf{liminfImp}[\mathsf{Pre}]$ the valuation v^{li} gives the value function for a stochastic game with liminf objective. In all cases that we consider, for the least fixpoint iterations are initialized with the valuation $\min r$ (i.e, the valuation that assigns the value $\min r$ to all states) , and for the greatest fixpoint iterations are initialized with the valuation $\max r$. We will illustrate the value-iteration algorithm and the fixpoint characterization on an example in the case of 2-player game graphs. In the following subsection we present the instantiation of Pre for different classes of game graphs.

3.1 1-player game graphs

In this subsection we present the results for 1-player game graphs with limsup and liminf objectives. For simplicity we consider 1-player game graphs with $S_P = \emptyset$ and $S_2 = \emptyset$ (the results for the case when $S_P = \emptyset$ and $S_1 = \emptyset$ are similar).

Strategy complexity. Pure memoryless optimal strategies exist for 1-player game graphs with limsup and liminf objectives. The result can be obtained as a special case of the result known for 2-player game graphs (see Section 3.2) or $1\frac{1}{2}$-player game graphs (see Section 3.3).

Value-iteration algorithm. We present the value iteration solution for 1-player game graphs. We define the *graph predecessor operator* $\mathsf{maxPre}\colon V \to V$ as the function on valuations defined by

$$\mathsf{maxPre}(v)(s) \;=\; \max\{\, v(s') \mid s' \in E(s) \,\}$$

for all valuations $v \in V$ and all states $s \in S$; that is, the value of $\mathsf{maxPre}(v)$ at a state s is the maximal value of v at the states that are successors of s. If the parameter Pre is instantiated as maxPre, then the nested fixpoint solution of (1) gives the value function for 1-player game graphs with limsup objectives, and the solution of (2) gives the value function for liminf objectives. Each inner improvement fixpoint converges within at most $\boldsymbol{n}$ steps, and the outer improvement fixpoints converges within at most $\boldsymbol{n}$ computations of inner improvement fixpoints. Every improvement step (i.e., each application of the function $\mathsf{limsupImp}[\mathsf{Pre}]$ or $\mathsf{liminfImp}[\mathsf{Pre}]$) can be computed in $O(\boldsymbol{m})$ time. Hence the value-iteration algorithm has the time complexity $O(\boldsymbol{m}\boldsymbol{n}^2)$.

Graph-theoretic algorithm. The value function for 1-player game graphs with the limsup and liminf can be obtained in $O(\boldsymbol{m})$ time. The algorithm for limsup objective is as follows. First compute the set of all maximal strongly connected components (this can be done in $O(\boldsymbol{m})$ time). For a bottom maximal strongly connected component C, the value of every state in C is $\max_{s \in C} r(s)$. Then proceed in a bottom up fashion: consider a maximal strongly connected component C' such that for every state $t \in \big(\bigcup_{s \in C'} E(s)\big) \setminus C'$ the value of state t is computed, and let this value be $v(t)$. The value of every state $s \in C'$ is as follows:

1. If either (a) $|C'| \geq 2$, or (b) $|C'| = 1$ and the only state of C' has a self-loop; then for every state $s \in C'$ the value $v(s)$ is given by

$$\max\{\ \max\{\ r(s) \mid s \in C'\ \}, \max\{\ v(t) \mid \exists s \in C' \cdot t \in E(s)\ \}\ \}.$$

2. If $|C'| = 1$ and the only state of C' does not have self-loop, then the value $v(s)$ of the only state s of C' is given by

$$\max\{\ v(t) \mid \exists s \in C' \cdot t \in E(s)\ \}.$$

Thus value of every state can be computed in $O(\boldsymbol{m})$ time. The algorithm for liminf objectives is similar. We know of no implementation of the nested value improvement scheme that matches this complexity. We summarize the results in the following theorem.

Theorem 2 (Complexity of 1-player game graphs). *For all 1-player game graphs with limsup and liminf objectives, the following assertions hold.*

1. *Pure memoryless optimal strategies exist.*
2. *The value function can be computed in $O(\boldsymbol{n}^2\boldsymbol{m})$ time by the value-iteration algorithm.*
3. *The value function can be computed in $O(\boldsymbol{m})$ time by the graph-theoretic algorithm.*

Remark 1. The graph-theoretic algorithm we present runs in $O(\boldsymbol{m})$ time, as compared to the previously known algorithm of [2] that runs in $O(\boldsymbol{m}+\boldsymbol{n}\cdot\log\boldsymbol{n})$ time. The algorithm of [2] first sorted states with respect to the rewards and then applied algorithms for Büchi (or coBüchi) objectives, whereas our algorithm does not need the sorting step of the previous algorithm.

3.2 2-player game graphs

We now present the results for 2-player game graphs with limsup and liminf objectives.

Strategy complexity. Pure memoryless optimal strategies exist for 2-player game graphs with limsup and liminf objectives. The result has several different proofs. In [13] Gimbert and Zielonka present sufficient conditions on measurable functions (that specify quantitative objectives) that ensures existence of pure memoryless optimal strategies in 2-player game graphs. It was also shown in [13] that limsup and liminf objectives satisfy the required conditions, and hence existence of pure memoryless optimal strategies in 2-player game graphs with limsup and liminf objectives follows.

Value-iteration algorithm. The value-iteration solution for 2-player game graphs uses the *game graph predecessor operator* $\mathsf{maxminPre}\colon V \to V$ defined by

$$\mathsf{maxminPre}(v)(s) \ = \ \begin{cases} \max\{\ v(s') \mid s' \in E(s)\ \} & \text{if } s \in S_1; \\ \min\{\ v(s') \mid s' \in E(s)\ \} & \text{if } s \in S_2; \end{cases}$$

for all valuations $v \in V$ and all states $s \in S$. In other words, the value of $\mathsf{maxminPre}(v)$ at a player-1 state s is the maximal value of v at the successors of s, and at a player-2 state s it is the minimal value of v at the successors of s. If the parameter Pre is instantiated as $\mathsf{maxminPre}$, then the nested fixpoint solution of (1) gives the value function for 2-player game graphs with limsup objectives, and the solution of (2) gives the value function for liminf objectives. Each inner improvement fixpoint converges within at most $\boldsymbol{n}$ steps, and the outer improvement fixpoints converges within at most $\boldsymbol{n}$ computations of inner improvement fixpoints. Every improvement step (i.e., each application of the function $\mathsf{limsupImp}[\mathsf{maxminPre}]$ or $\mathsf{liminfImp}[\mathsf{maxminPre}]$) can be computed in $O(\boldsymbol{m})$ time. Hence the value-iteration algorithm has the time complexity $O(\boldsymbol{m}\boldsymbol{n}^2)$.

Example 1 (2-player game with limsup objective). Consider the deterministic game shown in Fig. 1, where the reward function r is indicated by state labels. We consider the objective limsup(r) for player 1 (the $\Box$ player). We specify valuations as value vectors; the outer initial valuation is $v^0 = \langle 15, 15, 15, 15, 15\rangle$, and the inner initial valuation is $u^0 = \langle 5, 5, 5, 5, 5\rangle$. We compute the first inner improvement fixpoint: $u_0^0 = \langle 5, 5, 5, 5, 5\rangle$, and since

$$u_0^{j+1} \;=\; \min\{\, \max\{\, r, u_0^j, \mathsf{maxminPre}(u_0^j)\,\}, v^0, \mathsf{maxminPre}(v^0)\,\}$$

for all $j \geq 0$, where $v^0 = \mathsf{maxminPre}(v^0) = \langle 15, 15, 15, 15, 15\rangle$, we obtain $u_0^1 = \langle 5, 5, 15, 10, 5\rangle$. Note that u_0^1 coincides with the reward function r. Next we obtain $u_0^2 = \max\{\, r, u_0^1, \mathsf{maxminPre}(u_0^1)\,\} = \langle 10, 5, 15, 10, 10\rangle$. Finally $u_0^3 = u_0^4 = \langle 10, 10, 15, 10, 10\rangle$, which is the first inner improvement fixpoint v^1. Intuitively, $v^i(s)$ is the largest reward that player 1 can ensure to visit at least i times from s. The second inner improvement chain starts with $u_1^0 = \langle 5, 5, 5, 5, 5\rangle$ using

$$u_1^{j+1} \;=\; \min\{\, \max\{\, r, u_1^j, \mathsf{maxminPre}(u_1^j)\,\}, v^1, \mathsf{maxminPre}(v^1)\,\},$$

where $v^1 = \langle 10, 10, 15, 10, 10\rangle$ and $\mathsf{maxminPre}(v^1) = \langle 10, 10, 10, 10, 10\rangle$. Since $\max\{\, r, u_1^0, \mathsf{maxminPre}(u_1^0)\,\} = \langle 5, 5, 15, 10, 5\rangle$, we obtain $u_1^2 = \langle 5, 5, 10, 10, 5\rangle$ and $u_1^3 = \langle 10, 5, 10, 10, 10\rangle$ Then $u_1^3 = u_1^4 = \langle 10, 10, 10, 10, 10\rangle$, which is the second inner improvement fixpoint v^2. This is also the desired outer improvement fixpoint; that is, $v^{ls} = v^2 = v^3 = \langle 10, 10, 10, 10, 10\rangle$. The player-1 strategy that chooses at state s_0 the successor s_3 ensures that against all strategies of player 2, the reward 10 will be visited infinitely often. Dually, the player-2 strategy that chooses at s_1 the successor s_0 ensures that against all strategies of player 1, the reward 15 will be visited at most once. Hence $\langle 10, 10, 10, 10, 10\rangle$ is the 2-player game valuation of the player-1 objective limsup(r): from any start state, player 1 can ensure that reward 10 will be visited infinitely often, but she cannot ensure reward 15. ∎

Graph-theoretic algorithm. The value function for 2-player game graphs with the limsup and liminf objectives can be computed in $O(\frac{\boldsymbol{m}\boldsymbol{n}\log(\Delta)\log(\boldsymbol{k})}{\log(\boldsymbol{n})})$ time, where $\boldsymbol{k}$ is the number of different rewards of the reward function in the game graph. The algorithm for limsup objectives is as follows: we first sort the

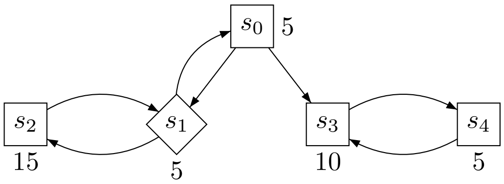

Fig. 1. Deterministic game with limsup objective.

rewards in ascending order, and let the reward values in ascending order be $r_1 < r_2 < \cdots < r_{\boldsymbol{k}}$. To check if the value at a state s is at at least r_i, for $1 \leq i \leq \boldsymbol{k}$, we consider all states with rewards at least r_i as Büchi states, and then check if player 1 can satisfy the Büchi objective from s. A game with a Büchi objective can be solved in $O(\frac{\boldsymbol{mn} \log(\Delta)}{\log(\boldsymbol{n})})$ time by graph-theoretic algorithms [6]. By a binary search over the sorted set of rewards we can compute the value in $O(\frac{\boldsymbol{mn} \log(\Delta) \log(\boldsymbol{k})}{\log(\boldsymbol{n})})$ time. The algorithm for liminf objectives is similar, and it uses solution of games with coBüchi objectives instead of Büchi objectives. We know of no implementation of the nested value improvement scheme that matches this complexity. We summarize the results in the following theorem.

Theorem 3 (Complexity of 2-player game graphs). *For all 2-player game graphs with limsup and liminf objectives, the following assertions hold.*

1. *Pure memoryless optimal strategies exist.*
2. *The value function can be computed in $O(\boldsymbol{n}^2\boldsymbol{m})$ time by the value-iteration algorithm.*
3. *The value function can be computed in $O(\frac{\boldsymbol{mn} \log(\Delta) \log(\boldsymbol{k})}{\log(\boldsymbol{n})})$ time by the graph-theoretic algorithm.*

Remark 2. Observe that for the worst case complexity for graph-theoretic algorithmic solution we have $\Delta = O(\boldsymbol{n})$ and $\boldsymbol{k} = O(n)$, and then the graph-theoretic algorithm runs in time $O(\boldsymbol{mn} \log(\boldsymbol{n}))$. The worst-case complexity of the previously known algorithm (of [2]) is $O(\boldsymbol{mn}^2)$.

3.3 1½-player game graphs

We now present the results for 1½-player game graphs with limsup and liminf objectives.

Strategy complexity. Pure memoryless optimal strategies exist for 1½-player game graphs with limsup and liminf objectives. This fact can be proved by straightforward extension of the results and proof techniques for MDPs with Büchi and coBüchi objectives. The existence of pure memoryless optimal strategies in MDPs with Büchi and coBüchi objectives has been shown in [8, 10].

Value-iteration algorithm. To present the value iteration solution for $1\frac{1}{2}$-player game graphs, we need the *probabilistic graph predecessor operator* $\mathsf{maxPre}^P \colon V \to V$ defined by

$$\mathsf{maxPre}^P(v)(s) \;=\; \begin{cases} \max\{\, v(s') \mid s' \in E(s) \,\} & \text{if } \; s \in S_1; \\ \sum_{s' \in E(s)} v(s') \cdot \delta(s)(s') & \text{if } \; s \in S_P; \end{cases}$$

for all valuations $v \in V$ and all states $s \in S$. In other words, the value of $\mathsf{maxPre}^P(v)$ at a player-1 state s is the maximal value of v at the successors of s, and the value of $\mathsf{maxPre}^P(v)$ at a probabilistic state s is the average value of v at the successors of s. If the parameter Pre is instantiated as maxPre^P, then the nested fixpoint solution of (1) gives the value function for $1\frac{1}{2}$-player game graphs with limsup objectives, and the solution of (2) gives the value function for liminf objectives. Unlike the case of 1-player and 2-player game graphs, the inner and outer iterations do not necessarily converge in finitely many iterations, but converge only in the limit. We now present the result on the *boundedness properties* of values for rational rewards and transition probabilities that allows to compute the exact values by value-iteration algorithms.

Precision of values. We assume that all transition probabilities and rewards are given as rational numbers, and for simplicity (but without loss of generality) we assume that all rewards are positive. From the existence of pure memoryless optimal strategies, and the results of [9, 20] it follows that all values in $1\frac{1}{2}$-player game graphs with limsup and liminf objectives are again rationals and that the denominators can be bounded. Let $\delta_u = \max\{\, d \mid \delta(s)(s') = \frac{n}{d} \text{ for } s \in S_P \text{ and } s' \in E(s) \,\}$ be the largest denominator of all transition probabilities. Let $r_u = \mathrm{lcm}\{\, d \mid r(s) = \frac{n}{d} \text{ for } s \in S \,\}$ be the least common multiple of all reward denominators. Let $r_{\max} = \max\{\, n \mid r(s) = \frac{n}{d} \text{ for } s \in S \,\}$ be the largest numerator of all rewards. Then, for all states $s \in S$, both $\langle\!\langle 1 \rangle\!\rangle_{val}^G(\mathrm{limsup}(r))(s)$ and $\langle\!\langle 1 \rangle\!\rangle_{val}^G(\mathrm{liminf}(r))(s)$ have the form $\frac{n}{d}$ for positive integers n and d with $n, d \leq \boldsymbol{\gamma}$, where

$$\boldsymbol{\gamma} \;=\; \delta_u^{4\boldsymbol{m}} \cdot r_u \cdot r_{\max}.$$

This *boundedness* property of values for limsup and liminf objectives in $1\frac{1}{2}$-player game graphs is the key for proving computability of the two improvement fixpoints. The inner fixpoint can be computed as follows: the improvement function can be iterated for $2 \cdot \boldsymbol{\gamma}^2$ iterations, and the valuation obtained is rounded to the nearest multiple of $\frac{1}{\gamma}$ to obtain the inner fixpoint (the argument is similar to the value-iteration algorithms of [9, 20]). Similarly, the valuation of the outer fixpoint can be obtained by rounding after $2 \cdot \boldsymbol{\gamma}^2$ iterations of the outer fixpoint computation. Hence the value-iteration algorithm has the time complexity $O(\boldsymbol{\gamma}^4)$.

Graph-theoretic algorithm and linear program. The value function for $1\frac{1}{2}$-player game graphs with the limsup and liminf objective can be computed in polynomial time. Let $\boldsymbol{k}$ be the number of different reward values. The key steps of the algorithm for limsup objective is as follows: (a) first the rewards are sorted in ascending order; (b) then *qualitative analysis* (computing the set of states with value 1) of sub-graphs of the given $1\frac{1}{2}$-player game graph with Büchi objectives is performed, and there are $\boldsymbol{k}$ calls to the qualitative analysis algorithm (see [5] for details) for Büchi objectives which can be performed in polynomial time using algorithms of [7]; (c) after the above analysis the value function can be obtained by solving a linear program. The algorithm for liminf objective is similar and it uses qualitative analysis for coBüchi objectives (see [5] for details). We know of no implementation of the nested value improvement scheme that runs in polynomial time. We summarize the results in the following theorem.

Theorem 4 (Complexity of $1\frac{1}{2}$-player game graphs). *For all $1\frac{1}{2}$-player game graphs with limsup and liminf objectives, the following assertions hold.*

1. *Pure memoryless optimal strategies exist.*
2. *The value function can be computed in $O(\gamma^4)$ time by the value-iteration algorithm.*
3. *The value function can be computed in polynomial time by the graph-theoretic algorithm and linear programming.*

3.4 $2\frac{1}{2}$-player game graphs

Finally, in this section we present the results for $2\frac{1}{2}$-player game graphs with limsup and liminf objectives.

Strategy complexity. Pure memoryless optimal strategies exist for $2\frac{1}{2}$-player game graphs with limsup and liminf objectives. The results (Theorem 3.19 of [12]) showed that if for a quantitative objective f and its complement $-f$ pure memoryless optimal strategies exist in $1\frac{1}{2}$-player game graphs, then pure memoryless optimal strategies also exist in $2\frac{1}{2}$-player games. Since pure memoryless optimal strategies exist for both limsup and liminf objectives in $1\frac{1}{2}$-player game graphs (Theorem 4), the existence of pure memoryless optimal strategies follows for $2\frac{1}{2}$-player games with limsup and liminf objectives.

Value-iteration algorithm. To present the value-iteration solution for $2\frac{1}{2}$-player game graphs, we need the *probabilistic game graph predecessor operator* $\mathsf{maxminPre}^P \colon V \to V$ defined by

$$\mathsf{maxminPre}^P(v)(s) \;=\; \begin{cases} \max\{\, v(s') \mid s' \in E(s) \,\} & \text{if } \; s \in S_1; \\ \min\{\, v(s') \mid s' \in E(s) \,\} & \text{if } \; s \in S_2; \\ \sum_{s' \in E(s)} v(s') \cdot \delta(s)(s') & \text{if } \; s \in S_P; \end{cases}$$

for all valuations $v \in V$ and all states $s \in S$. The predecessor operator $\mathsf{maxminPre}^P$ is a generalization of game graph predecessor operator $\mathsf{maxminPre}$ and the probabilistic graph predecessor operator maxPre^P. If the parameter Pre is instantiated as $\mathsf{maxminPre}^P$, then the nested fixpoint solution of (1) gives the value functions for $2\frac{1}{2}$-player game graphs with limsup objectives, and the solution of (2) gives the value function for liminf objectives. The boundedness properties of the values of $1^1\!/_2$-player game graphs also holds for $2^1\!/_2$-player game graphs, and bounds on the number of iterations to compute the fixpoints for $1^1\!/_2$-player game graphs also generalize to $2^1\!/_2$-player game graphs. Hence if all the rewards and transition probabilities are rational, then the value function for $2^1\!/_2$-player game graphs with limsup and liminf objectives can be computed in $O(\gamma^4)$ time using value-iteration algorithm.

Optimal algorithm. The problem to decide, given a state s and a rational number q, whether the value function at s is at least q for $2^1\!/_2$-player game graphs with limsup and liminf objectives lies in NP $\cap$ coNP [5]. The result follows from existence of pure memoryless strategies, and the polynomial time algorithms to compute values in $1^1\!/_2$-player game graphs with limsup and liminf objectives. No polynomial-time algorithms are known for computing values for limsup and liminf objectives in $2^1\!/_2$-player game graphs. In particular, the qualitative analysis and the linear-programming approach for $1^1\!/_2$-player game graphs do not generalize to $2^1\!/_2$-player game graphs. We summarize the results in the following theorem.

Theorem 5 (Complexity of $2^1\!/_2$-player game graphs). *For all $2^1\!/_2$-player game graphs with limsup and liminf objectives, the following assertions hold.*

1. *Pure memoryless optimal strategies exist.*
2. *The value function can be computed in $O(\gamma^4)$ time by value-iteration algorithm.*
3. *Given a state s and a rational number q, whether the value function at s is at least q can be decided in NP $\cap$ coNP.*

4 Conclusion

In this survey, we considered stochastic games and their subclasses with limsup and liminf objectives. In Table 1, we summarize the results for the different classes of game graphs. We presented a comprehensive study of the known results in terms of the complexity of strategies, and the two classical algorithmic solutions, namely, value-iteration algorithms and graph-theoretic algorithms. For 1-player and 2-player games, we also improved the previously known graph-theoretic algorithms and their complexity.

Finally, note that the 1-player game graphs with limsup and liminf objective can be viewed as weighted automata with limsup and liminf functions, and computing the value of such games can then be viewed as computing the greatest value of a word in such weighted automata, which amounts to solving the so-called quantitative emptiness problem [3].

References

1. H. Bjorklund, S. Sandberg, and S. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. In *MFCS'04*, pages 673–685, 2004.

$\boldsymbol{n}$ states $\boldsymbol{m}$ edges	Objective limsup(r)	Objective liminf(r)
1-player graphs	Predecessor operator $\mathsf{limsupImp}[\mathsf{maxPre}](v,u)$ Value-iteration complexity $O(\boldsymbol{n}^2\boldsymbol{m})$ Best known complexity $O(\boldsymbol{m})$	Predecessor operator $\mathsf{liminfImp}[\mathsf{maxPre}](v,u)$ Value-iteration complexity $O(\boldsymbol{n}^2\boldsymbol{m})$ Best known complexity $O(\boldsymbol{m})$
2-player games	Predecessor operator $\mathsf{limsupImp}[\mathsf{maxminPre}](v,u)$ Value-iteration complexity $O(\boldsymbol{n}^2\boldsymbol{m})$ Best known complexity $O(\boldsymbol{nm}\log\boldsymbol{n})$	Predecessor operator $\mathsf{liminfImp}[\mathsf{maxminPre}](v,u)$ Value-iteration complexity $O(\boldsymbol{n}^2\boldsymbol{m})$ Best known complexity $O(\boldsymbol{nm}\log\boldsymbol{n})$
1½-player graphs	Predecessor operator $\mathsf{limsupImp}[\mathsf{maxPre}^P](v,u)$ Value-iteration complexity $O(\boldsymbol{\gamma}^4)$ Best known complexity is polynomial time	Predecessor operator $\mathsf{liminfImp}[\mathsf{maxPre}^P](v,u)$ Value-iteration complexity $O(\boldsymbol{\gamma}^4)$ Best known complexity is polynomial time
2½-player games	Predecessor operator $\mathsf{limsupImp}[\mathsf{maxminPre}^P](v,u)$ Value-iteration complexity $O(\boldsymbol{\gamma}^4)$ Best known complexity is NP $\cap$ coNP	Predecessor operator $\mathsf{liminfImp}[\mathsf{maxminPre}^P](v,u)$ Value-iteration complexity $O(\boldsymbol{\gamma}^4)$ Best known complexity NP $\cap$ coNP

Table 1. Nested value improvement for limsup and liminf objectives. Recall that $\boldsymbol{\gamma}$ is such that $16^{\boldsymbol{n}} \in O(\boldsymbol{\gamma})$.

2. A. Chakrabarti, L. de Alfaro, T. A. Henzinger, and M. Stoelinga. Resource interfaces. In *EMSOFT*, LNCS 2855, pages 117–133. Springer, 2003.
3. K. Chatterjee, L. Doyen, and T. A. Henzinger. Quantitative languages. In *CSL*, LNCS 5213, pages 385–400. Springer, 2008.
4. K. Chatterjee and T. A. Henzinger. Value iteration. In *25 Years of Model Checking*, LNCS 5000, pages 107–138. Springer, 2008.
5. K. Chatterjee and T. A. Henzinger. Probabilistic systems with limsup and liminf objectives. In *ILC*. 2009.
6. K. Chatterjee, T. A. Henzinger, and N. Piterman. Algorithms for Büchi games. In *GDV*. 2006.
7. K. Chatterjee, M. Jurdziński, and T.A. Henzinger. Simple stochastic parity games. In *CSL'03*, volume 2803 of *LNCS*, pages 100–113. Springer, 2003.
8. K. Chatterjee, M. Jurdziński, and T.A. Henzinger. Quantitative stochastic parity games. In *SODA'04*, pages 121–130. SIAM, 2004.
9. 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.

10. L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, 1997.
11. J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer-Verlag, 1997.
12. H. Gimbert. *Jeux positionnels*. PhD thesis, Université Paris 7, 2006.
13. H. Gimbert and W. Zielonka. Games where you can play optimally without any memory. In *CONCUR'05*, pages 428–442. Springer, 2005.
14. R.M. Karp. A characterization of the minimum cycle mean in a digraph. *Discrete Mathematics*, 23:309–311, 1978.
15. T. A. Liggett and S. A. Lippman. Stochastic games with perfect information and time average payoff. *Siam Review*, 11:604–607, 1969.
16. A. Maitra and W. Sudderth, editors. *Discrete Gambling and Stochastic Games*. Springer, 1996.
17. D.A. Martin. The determinacy of Blackwell games. *The Journal of Symbolic Logic*, 63(4):1565–1581, 1998.
18. R. McNaughton. Infinite games played on finite graphs. *Annals of Pure and Applied Logic*, 65:149–184, 1993.
19. W. Thomas. Languages, automata, and logic. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, Beyond Words, chapter 7, pages 389–455. Springer, 1997.
20. U. Zwick and M. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.