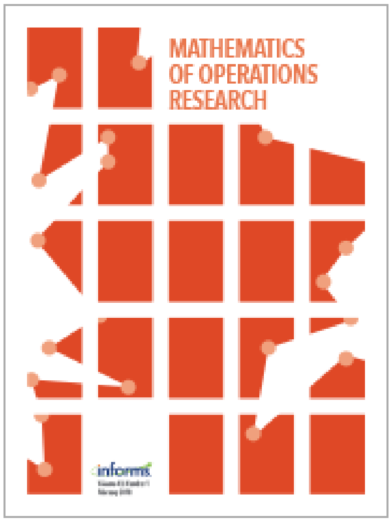

Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

New Algorithms for Solving Zero-Sum Stochastic Games

Miquel Oliu-Barton

To cite this article:
Miquel Oliu-Barton (2020) New Algorithms for Solving Zero-Sum Stochastic Games. Mathematics of Operations Research

Published online in Articles in Advance 28 May 2020

. https://doi.org/10.1287/moor.2020.1055

MATHEMATICS OF OPERATIONS RESEARCH
Articles in Advance, pp. 1–13
ISSN 0364-765X (print), ISSN 1526-5471 (online)
http://pubsonline.informs.org/journal/moor

New Algorithms for Solving Zero-Sum Stochastic Games

Miquel Oliu-Barton[a]

[a] Université Paris-Dauphine, Paris Sciences et Lettres Research University, Centre de Recherche en Mathématiques de la Décision, 75016 Paris, France

Contact: miquel.oliu.barton@normalesup.org, https://orcid.org/0000-0003-1793-6350 (MO-B)

Published Online in Articles in Advance: May 28, 2020

MSC2000 Subject Classification: Primary: 91A15; secondary: 90C47, 90C60

OR/MS Subject Classification: Primary: Games/group decisions: stochastic; secondary: analysis of algorithms: computational complexity

https://doi.org/10.1287/moor.2020.1055

Abstract. Zero-sum stochastic games, henceforth stochastic games, are a classical model in game theory in which two opponents interact and the environment changes in response to the players' behavior. The central solution concepts for these games are the discounted values and the value, which represent what playing the game is worth to the players for different levels of impatience. In the present manuscript, we provide algorithms for computing exact expressions for the discounted values and for the value, which are polynomial in the number of pure stationary strategies of the players. This result considerably improves all the existing algorithms.

Funding: Financial support from Agence Nationale de la Recherche [Grant 15-CE38-0007-01] and the Cowles Foundation at Yale University is gratefully acknowledged.

Keywords: stochastic games • complexity • linear programming • minimax • discounted values • limit value • algorithms

1. Introduction

1.1. Motivation

Zero-sum stochastic games, henceforth stochastic games, were introduced by Shapley [23] in 1953 in order to model the dynamic interaction between two opponents. The theory of stochastic games and its applications have been studied in several scientific disciplines, including economics, operations research, evolutionary biology, and computer science. In addition, mathematical tools that were used and developed in the study of stochastic games are used by mathematicians and computer scientists in other fields. Computer scientists refer to stochastic games as *concurrent games*, a terminology that was introduced to distinguish these games from the so-called turn-based (or simple) stochastic games, where the players alternate moves.

Stochastic games generalize matrix games and Markov decision problems; they are played over a finite set of states, and to each state corresponds a matrix game. Stochastic games are played in stages. At each stage $m \geq 1$, a stage reward g_m is produced, which depends on the current state k_m, commonly observed by the players, and on the current pair of actions (i_m, j_m) chosen by the players. The game is zero-sum, in the sense that player 1 receives g_m, while player 2 receives $-g_m$. A *λ-discounted stochastic game* is one where player 1 maximizes the expectation of the normalized λ-discounted sum $\sum_{m\geq 1} \lambda(1-\lambda)^{m-1} g_m$ for some discount rate $\lambda \in (0,1]$, while player 2 minimizes the same amount. The case where the discount rate is close to zero is of particular importance, as it stands for the case where the players are patient. Alternatively, the interaction between patient players can be modeled by an *undiscounted stochastic game*—that is, one in which player 1 maximizes the expectation of $\liminf_{T\to+\infty} \frac{1}{T}\sum_{m=1}^{T} g_m$, while player 2 minimizes the same amount.

The central solution concept for zero-sum games is its value. When it exists, the value is the maximal amount that each player can obtain in expectation, regardless of her opponent's behavior. The value of the λ-discounted stochastic game is often referred to as its *λ-discounted value*, while the value of the undiscounted stochastic game is referred to as its *value*. In the present manuscript, I propose new algorithms for computing the discounted values and the value of any stochastic game, which are polynomial in the number of pure stationary strategies of the game—that is, strategies that depend only on the current state. More precisely, for a stochastic game with n states and p actions available at each state, we provide explicit bounds that are polynomial in p^n. These results considerably improve all prior algorithms for computing the discounted values and the value of a stochastic game. In particular, they improve the best of them, due to Hansen et al. [11], where the bounds are polynomial in p and $2^{n^{n^2}}$.

1.1.1. Notation. In the sequel, we denote by $K = \{1, \ldots, n\}$ the set of states, for some $n \in \mathbb{N}$. For any state $1 \leq k \leq n$ and any discount rate $\lambda \in (0,1]$, the value of the λ-discounted stochastic game with initial state k is denoted by v_λ^k, and the value of the undiscounted stochastic game with initial state k is denoted by v^k. We also set $v_\lambda := (v_\lambda^1, \ldots, v_\lambda^n) \in \mathbb{R}^n$ and $v := (v^1, \ldots, v^n) \in \mathbb{R}^n$.

1.2. State of the Art

In his seminal paper, Shapley [23] defined stochastic and proved that these games have a λ-discounted value for each $\lambda \in (0, 1]$ and that both players have optimal stationary strategies—that is, strategies that depend only on the current state. Furthermore, a characterization was obtained for the vector of values v_λ, as the unique fixed point of an operator from $\mathbb{R}^n$ to $\mathbb{R}^n$, which is contracting for the L^∞ norm. Blackwell and Ferguson [5] considered a particular stochastic game, the so-called "Big Match," and proved the existence of the value and the equality to $\lim_{\lambda\to 0} v_\lambda = v$. Their result was then extended by Kohlberg [16] to the class of absorbing games—that is, a class of stochastic games in which there is at most one transition between states. For general stochastic games, the convergence of the discounted values v_λ, as λ goes to 0, was proved by Bewley and Kohlberg [3], building on Shapley's characterization of the discounted values and on Tarski–Seidenberg elimination theorem from semialgebraic geometry. The existence of the value v of the undiscounted stochastic game, and the equality $v = \lim_{\lambda\to 0} v_\lambda$, were proved by Mertens and Neyman [18]. An alternative proof of the convergence of v_λ and the existence of v using the theory of finite dimensional linear systems was found by Oliu-Barton [20]. Explicit characterizations for v_λ^k and v^k, for each $1 \le k \le n$, were recently obtained by Attia and Oliu-Barton [1].

Whether the value of a finite stochastic game can be computed in polynomial time is a famous open problem in computer science. This problem is intriguing because the simpler class of *simple stochastic games* is both nondeterministic polynomial (NP) and co-NP by Condon [8], and several famous problems with this property have eventually been shown to be polynomial-time solvable, such as primality testing or linear programming. (A simple stochastic game is one where, for each state, the transition function depends on one player's action only.) The known algorithms fall into two categories: decision procedures for the first order theory of the reals, such as Chatterjee et al. [7], Etessami and Yannakakis [9], and Solan and Vielle [24], and value or strategy iteration methods, such as Henzinger et al. [12], and Rao et al. [21]. All of them are worst-case exponentials in the number of states or in the number of actions. Hansen et al. [10] proved that no value or strategy iteration algorithm can ever achieve a polynomial bound. Recently, Hansen et al. [11] obtained a remarkable improvement using the machinery of real-algebraic geometry in a more indirect manner: They provided an algorithm that, for any fixed number of states, is polynomial in the number of actions. However, the dependence on the number of states is an implicit double-exponential expression, which is problematic in terms of practical computations. In their own words, "the exponent in the polynomial time bound is $O(n)^{n^2}$, i.e., the complexity is doubly exponential in n," from which they claim that "getting a better dependence on n is a very interesting open problem" (Hansen et al. [11, p. 3]).

1.3. Main Results

In the present paper, we propose a new method for computing the discounted values and the value of a stochastic game. Unlike all prior works, we build on the new characterizations that were obtained by Attia and Oliu-Barton [1]. Our algorithms are polynomial in the number of actions, for any *fixed* number of states, but the dependence on the number of states is explicit and simply exponential. Equivalently, our algorithms are polynomial in the number of pure stationary strategies—that is, strategies that depend only on the current state. This improvement opens up the path for actually solving stochastic games in practice. An important ingredient in our work is the following *continuity result*: For any $\varepsilon > 0$, we provide an explicit discount rate $\lambda_\varepsilon \in (0, 1]$ whose bit size is polynomial in the number of pure stationary strategies and in $\log \varepsilon$, so that $|v_\lambda^k - v^k| \le \varepsilon$ for all $\lambda \in (0, \lambda_\varepsilon)$ and $1 \le k \le n$.

1.4. Organization of the Paper

In Section 2, we introduce the model (Section 2.2), state our main results formally (Section 2.3), and recall some relevant results from the literature (Section 2.4). Section 3 is devoted to proofs of the results that do not involve algorithms—namely, some algebraic properties for v_λ^k and v^k and a bound for the discount rate, which ensures that the two are close to each other. Section 4 is devoted to the algorithms. More precisely, we start by recalling some classical algorithms (Section 4.1). Next, we present two algorithms: one that outputs a crude approximation of v_λ^k and one that outputs an exact expression of v_λ^k (Section 4.2). Similarly, Section 4.3 is devoted to the computation of a crude approximation and of an exact expression, of v^k. Finally, we provide an example in Section 4.4 in order to illustrate our results.

2. Stochastic Games

We now introduce the model of stochastic games and some basic facts. For a more detailed presentation of stochastic games, see, for instance, Sorin [25, chapter 5] and Renault [22].

2.1. Notation

The following notations will be used throughout the paper.

- For each finite set E, we denote its cardinality by $|E|$ and the set of probability distributions over E by $\Delta(E) = \{f : E \to [0,1], \sum_{e\in E} f(e) = 1\}$.
- We denote by n the number of states.
- $I^1, \dots, I^n$ and $J^1, \dots, J^n$ denote $2n$ fixed finite sets of actions.
- We set $I := I^1 \times \cdots \times I^n$ and $J := J^1 \times \cdots \times J^n$.
- We set $d := \min(|I|, |J|)$.
- We set $X := \Delta(I^1) \times \cdots \times \Delta(I^n)$ and $Y := \Delta(J^1) \times \cdots \times \Delta(J^n)$.
- For any $\alpha \in \mathbb{R}$, $\lceil \alpha \rceil$ denotes the unique integer satisfying $\alpha \le \lceil \alpha \rceil < \alpha + 1$.
- For any $p \in \mathbb{N}$ we denote its bit-size by $\mathsf{bit}(p) := \lceil \log_2(p+1) \rceil$.
- For $(p,q) \in \mathbb{N}^2$, we set $\mathsf{bit}(p/q) = \mathsf{bit}(p) + \mathsf{bit}(q)$.
- For any tuplet of nonnegative integers (a,b,c,M), we define

$$\varphi(a,b,c,M) := ab(\mathsf{bit}(a) + \mathsf{bit}(b) + \mathsf{bit}(c) + \mathsf{bit}(M)).$$

- In particular, for any $N \in \mathbb{N}$, one has $\varphi(n,d,N,0) = nd(\mathsf{bit}(n) + \mathsf{bit}(d) + \mathsf{bit}(N))$.

2.2. Model

We now describe the classical model of stochastic games, as in Shapley [23].

A *stochastic game* is described by a tuple (K, I, J, g, q, k), where

- $K = \{1, \dots, n\}$ is a finite set of states.
- For each $1 \le \ell \le n$, I^ℓ and J^ℓ are the sets of available actions for players 1 and 2, respectively, at state ℓ.
- $g : Z \to \mathbb{R}$ is the reward function, where $Z := \{(\ell, i, j) \mid \ell \in K, (i,j) \in I^\ell \times J^\ell\}$.
- $q : Z \to \Delta(K)$ is the transition function.
- $1 \le k \le n$ is the initial state.

The game proceeds in stages as follows. At each stage $m \ge 1$, both players are informed of the current state $k_m \in K$. Then, independently, player 1 chooses an action $i_m \in I^{k_m}$ and player 2 chooses an action $j_m \in J^{k_m}$. The pair (i_m, j_m) is then observed by the players, from which they can infer the stage reward $g_m := g(k_m, i_m, j_m)$. A new state k_{m+1} is then chosen with the probability distribution $q(k_m, i_m, j_m)$, and the game proceeds to stage $m+1$.

- A *λ-discounted stochastic game* is one where player 1 maximizes $\sum_{m\ge1} \lambda(1-\lambda)^{m-1} g_m$ in expectation, while player 2 minimizes the same amount, for some $\lambda \in (0,1]$.
- An *undiscounted stochastic game* is one where player 1 maximizes $\liminf_{T\to+\infty} \frac{1}{T} \sum_{m=1}^{T} g_m$ in expectation, while player 2 minimizes the same amount.

2.2.1. Strategies. A (behavioral) *strategy* is a decision rule from the set of possible observations to the set of probabilities over the set of available actions. For every stage $m \ge 1$, the set of possible observations at stage m is $Z^{m-1} \times K$. A strategy for player 1 is, thus, a sequence of mappings $\sigma = (\sigma_m)_{m\ge1}$ so that $\sigma_m(h_m, k_m) \in \Delta(I^{k_m})$ for all $(h_m, k_m) \in Z^{m-1} \times K$. Similarly, a strategy for player 2 is a sequence of mappings $\tau = (\tau_m)_{m\ge1}$ so that $\tau_m(h_m, k_m) \in \Delta(J^{k_m})$ for all $(h_m, k_m) \in Z^{m-1} \times K$. Both players choose their strategies independently. The sets of strategies are denoted, respectively, by Σ and $\mathcal{T}$. By the Kolmogorov extension theorem, the initial state k, the transition function q, and a pair of strategies (σ, τ) induce a unique probability over the set of plays $Z^{\mathbb{N}}$, endowed with the sigma-algebra generated by the cylinders corresponding to finite histories—that is, the sets $(z_1, \dots, z_p) \times Z^{\mathbb{N}}$ for every $p \in \mathbb{N}$ and $(z_1, \dots, z_p) \in Z^p$. This probability is denoted by $\mathsf{P}^k_{\sigma,\tau}$, and $\mathbb{E}^k_{\sigma,\tau}$ denotes the expectation with respect to $\mathsf{P}^k_{\sigma,\tau}$.

2.2.2. Stationary Strategies. A *stationary strategy* is one that depends on the past observations only through the current state. A stationary strategy of player 1, denoted by $x = (x^1, \dots, x^n)$, is thus an element of X. Similarly, $y = (y^1, \dots, y^n) \in Y$ is a stationary strategy of player 2. The sets I and J are the sets of *pure stationary strategies*. We use the notation $\mathbf{i} = (\mathbf{i}^1, \dots, \mathbf{i}^n) \in I$ and $\mathbf{j} = (\mathbf{j}^1, \dots, \mathbf{j}^n) \in J$.

2.2.3. Discounted and Undiscounted Payoffs. To every pair $(\sigma,\tau)\in\Sigma\times\mathcal{T}$ corresponds a λ-discounted payoff for each discount rate $\lambda\in(0,1]$ and an undiscounted payoff, in the game (K,I,J,g,q,k). They are given by

$$\gamma_\lambda^k(\sigma,\tau) := \mathbb{E}_{\sigma,\tau}^k\left[\sum_{m\ge 1}\lambda(1-\lambda)^{m-1}g_m\right],$$
$$\gamma^k(\sigma,\tau) := \mathbb{E}_{\sigma,\tau}^k\left[\liminf_{T\to+\infty}\frac{1}{T}\sum_{m=1}^{T}g_m\right].$$

2.2.4. The Discounted Values and the Value. For each discount rate $\lambda\in(0,1]$, the λ-discounted stochastic game (K,I,J,g,q,k) has a value, denoted by v_λ^k, whenever

$$v_\lambda^k = \sup_{\sigma\in\Sigma}\inf_{\tau\in\mathcal{T}}\gamma_\lambda^k(\sigma,\tau) = \inf_{\tau\in\mathcal{T}}\sup_{\sigma\in\Sigma}\gamma_\lambda^k(\sigma,\tau).$$

Similarly, the undiscounted stochastic game (K,I,J,g,q,k) has a value, denoted by v^k, whenever

$$v^k = \sup_{\sigma\in\Sigma}\inf_{\tau\in\mathcal{T}}\gamma^k(\sigma,\tau) = \inf_{\tau\in\mathcal{T}}\sup_{\sigma\in\Sigma}\gamma^k(\sigma,\tau).$$

2.2.5. Classical Results. The existence of v_λ^k is due to Shapley [23], while Mertens and Neyman [18] proved the existence of v^k and the equality $v^k=\lim_{\lambda\to 0}v_\lambda^k$.

In the sequel, we refer to v_λ^k and to v^k as the λ-discounted value and the value, respectively, of the stochastic game (K,I,J,g,q,k).

2.3. Main Results

In the sequel, we consider stochastic games that can be described with rational data. For any $N\in\mathbb{N}$, we say that the stochastic game (K,I,J,g,q,k) satisfies (H_N) if $g(\ell,i,j)$ and $q(\ell'\mid\ell,i,j)$ belong to the set $\{0,\frac{1}{N},\frac{2}{N},\dots,1\}$ for all $(\ell,i,j)\in Z$ and $1\le\ell'\le n$. Recall that $n=|K|$ is the number of states.

The main contributions of the present paper concern the computation of the discounted value v_λ^k and the value v^k of a stochastic game satisfying (H_N). These numbers are known to be *algebraic*—that is, there exists polynomials P and Q with integer coefficients and so that $P(v_\lambda^k)=0$ and $Q(v^k)=0$. For an algebraic number $\alpha\in\mathbb{R}$, an *exact expression for* α is a triplet $(P;a,b)$, where P is a polynomial with integer coefficients, (a,b) is a pair of rational numbers, and α is the unique root of P in the interval (a,b). Thus, for instance, $(z^2-2;1,2)$ is an exact expression for $\sqrt{2}$.

The complexity of the algorithms presented in this paper will be measured with the so-called *logarithmic cost model*, which consists of assigning, to every arithmetic operation, a cost that is proportional to the number of bits involved. An algorithm is *polynomial in the variables* $t_1,\dots,t_m$, if its logarithmic cost can be bounded by a polynomial expression of $t_1,\dots,t_m$.

We can now state our results formally.

Theorem 1. *There exists an algorithm that takes as input a stochastic game (K,I,J,g,q,k) satisfying (H_N) for some $N\in\mathbb{N}$ and a discount rate satisfying $\lambda\in\{0,\frac{1}{M},\frac{2}{M},\dots,1\}$ for some $M\in\mathbb{N}$ and outputs an exact expression for its discounted value v_λ^k. The algorithm is polynomial in n, $|I|$, $|J|$, $\log N$, and $\log M$.*

Theorem 2. *There exists an algorithm that takes as input a stochastic game (K,I,J,g,q,k) satisfying (H_N) for some $N\in\mathbb{N}$ and outputs an exact expression for its value v^k. The algorithm is polynomial in n, $|I|$, $|J|$, and $\log N$.*

The algorithms that are mentioned in Theorems 1 and 2 are provided in Sections 4.2 and 4.3, respectively. Though very similar, the second algorithm has an additional ingredient, namely, a new bound on how small the discount rate needs to be so that v_λ^k and v^k are close to each other. This result, which has an interest in its own, can be formalized as follows.

Theorem 3. *For each $r\in\mathbb{N}$, set $\lambda_r := 2^{-4nd(\mathrm{bit}(n)+\mathrm{bit}(d)+\mathrm{bit}(N))-rnd}$. Then, for any stochastic game (K,I,J,g,q,k) satisfying (H_N), one has*

$$\left|v_\lambda^k - v^k\right| \le 2^{-r} \qquad \forall\lambda\in(0,\lambda_r].$$

2.3.1. Comments. The previous results deserve some comments. For simplicity, assume the existence of some $p \in \mathbb{N}$ so that at $|I^\ell| = |J^\ell| = p$ for all $1 \le \ell \le n$.

1. The number of pure stationary strategies for each player is p^n, while the number of actions is np. Consequently, polynomial expressions in $|I|$, $|J|$ or d are *exponential* expressions in n.
2. Theorems 1 and 2 improve the algorithms provided by Hansen et al. [11]. Our main achievement is twofold: One the one hand, we reduce the dependence on n from a double exponential to a simple exponential; on the other, our algorithms are more direct and considerably simpler.
3. Theorem 1 outperforms value iteration algorithms (such as the iteration of Shapley's operator) for small discount rates. Indeed, whereas our algorithm is polynomial in p^n, n and $\log \lambda$, the latter is polynomial in p, n and λ, for small λ.
4. For the discount rate λ_r in Theorem 3, one has $\mathsf{bit}(\lambda_r) = O(nd(r + \mathsf{bit}(n) + \mathsf{bit}(d) + \mathsf{bit}(N))$, which is of order p^n. This result improves Hansen et al. [11, proposition 22], where an expression for the order of the bit size of λ_r is obtained in terms of big O's, namely, of order $p^{O(n^2)}$. Furthermore, the reduction from $p^{O(n^2)}$ to p^n is fairly tight. Indeed, transposing Hansen et al. [10, theorem 8] into the discounted case, it follows that one can construct an example for which a discount rate λ of bit-size $p^{n/2}$ is not enough to ensure that v_λ^k and v^k are close to each other.

2.4. Selected Past Results

We now gather some past results that will be used in our proofs. We start by recalling the auxiliary matrices that were introduced in Attia and Oliu-Barton [1].

2.4.1. The Auxiliary Matrix Games. Consider the play induced by a pair $(\mathbf{i}, \mathbf{j}) \in I \times J$ of pure stationary strategies. Every time that the state $1 \le \ell \le n$ is reached, the players play $(\mathbf{i}^\ell, \mathbf{j}^\ell) \in I^\ell \times J^\ell$, so that the stage reward is $g(\ell, \mathbf{i}^\ell, \mathbf{j}^\ell)$ and the law of the next state is given by $q(\ell, \mathbf{i}^\ell, \mathbf{j}^\ell)$. Hence, the state variable follows a Markov chain with transition matrix $Q(\mathbf{i}, \mathbf{j}) \in \mathbb{R}^{n \times n}$ and the stage rewards can be described by a vector $g(\mathbf{i}, \mathbf{j}) \in \mathbb{R}^n$. For any $\lambda \in (0, 1]$, let $\gamma_\lambda(\mathbf{i}, \mathbf{j}) := (\gamma_\lambda^1(\mathbf{i}, \mathbf{j}), \dots \gamma_\lambda^n(\mathbf{i}, \mathbf{j})) \in \mathbb{R}^n$ be the vector of expected payoffs in the λ-discounted game, as the initial state varies from 1 to n. By stationarity, $Q(\mathbf{i}, \mathbf{j}), g(\mathbf{i}, \mathbf{j})$ and $\gamma_\lambda(\mathbf{i}, \mathbf{j})$ satisfy the recursive relation

$$\gamma_\lambda(\mathbf{i}, \mathbf{j}) = \lambda g(\mathbf{i}, \mathbf{j}) + (1 - \lambda) Q(\mathbf{i}, \mathbf{j}) \gamma_\lambda(\mathbf{i}, \mathbf{j}).$$

The matrix $\mathrm{Id} - (1 - \lambda) Q(\mathbf{i}, \mathbf{j})$ is invertible, so that, by Cramer's rule, one has

$$\gamma_\lambda^k(\mathbf{i}, \mathbf{j}) = \frac{d_\lambda^k(\mathbf{i}, \mathbf{j})}{d_\lambda^0(\mathbf{i}, \mathbf{j})}, \tag{1}$$

where $d_\lambda^0(\mathbf{i}, \mathbf{j}) := \det(\mathrm{Id} - (1 - \lambda) Q(\mathbf{i}, \mathbf{j})) \neq 0$ and where $d_\lambda^k(\mathbf{i}, \mathbf{j})$ is the determinant of the $n \times n$-matrix obtained by replacing the k-th column of $\mathrm{Id} - (1 - \lambda) Q(\mathbf{i}, \mathbf{j})$ with $\lambda g(\mathbf{i}, \mathbf{j})$. The auxiliary matrix $W_\lambda^k(z)$ is obtained by linearizing the quotient in (1) with an auxiliary variable $z \in \mathbb{R}$, as follows.

Definition 1. For any $z \in \mathbb{R}$, the matrix $W_\lambda^k(z) \in \mathbb{R}^{I \times J}$ is defined by setting

$$W_\lambda^k(z)[\mathbf{i}, \mathbf{j}] := d_\lambda^k(\mathbf{i}, \mathbf{j}) - z d_\lambda^0(\mathbf{i}, \mathbf{j}), \quad \forall (\mathbf{i}, \mathbf{j}) \in I \times J.$$

For each real matrix M, we denote its value by $\mathsf{val}\, M$.

2.4.2. Three Useful Results. The following two results are the main object of Attia and Oliu-Barton [1].

Theorem 4. *For any $\lambda \in (0, 1]$, v_λ^k is the unique $z \in \mathbb{R}$ so that $\mathsf{val}\, W_\lambda^k(z) = 0$. Furthermore, the map $z \mapsto \mathsf{val}\, W_\lambda^k(z)$ is strictly decreasing.*

Theorem 5. *$F^k(z) := \lim_{\lambda \to 0} \lambda^{-n}\, \mathsf{val}\, W_\lambda^k(z)$ exists in $\mathbb{R} \cup \{\pm\infty\}$ for all $z \in \mathbb{R}$, and v_λ^k converges, as λ goes to 0, to the unique $w \in \mathbb{R}$ so that $z > w \Rightarrow F^k(z) < 0$ and $z < w \Rightarrow F^k(z) > 0$. Furthermore, the map $z \mapsto F^k(z)$ is strictly decreasing.*

Next, we recall a well-known formula for the value of a matrix, due to Kaplansky [14]. For any matrix M of size $p \times p$, we denote by $S(M)$ the sum of the entries of the adjugate matrix of M, with the convention $S(M) = 1$ if $p = 1$ (i.e., when the adjugate matrix is not defined).

Theorem 6. *For any matrix M of size $p \times q$, there exists a square submatrix of M, denoted by $\dot{M}$, so that $S(\dot{M}) \neq 0$ and $\mathsf{val}\, M = \frac{\det \dot{M}}{S(\dot{M})}$.*

3. Algebraic Properties of the Values

Throughout this section, (K, I, J, g, q, k) denotes a stochastic game satisfying (H_N) for some $N \in \mathbb{N}$. Recall that a real number α is *algebraic of degree p* if there exists a polynomial P with integer coefficients satisfying $P(\alpha) = 0$, and p is the lowest degree of all such polynomials. The *defining polynomial* of α is the unique polynomial with integer coefficients $P(z) = a_0 + a_1 z + \cdots + a_p z^p$ so that $P(\alpha) = 0$, $a_p > 0$ and $\gcd(a_0, \ldots, a_p) = 1$. In Section 3.1, we combine a technical result from Basu et al. [2] and Theorems 4–6 to establish new bounds for the degree and the coefficients of the defining polynomials of v_λ^k and v^k. These results will be used to analyze the algorithms corresponding to Theorems 1 and 2. In Section 3.2, we establish Theorem 3, a result that reduces the computation of the value of a stochastic game to the computation of its discounted value, for a well-chosen discount rate. This result will be used in the algorithm corresponding to Theorem 2.

3.1. Bounds on the Defining Polynomials of the Values

We start by recalling proposition 8.12 of Basu et al. [2].

Lemma 1. *Let A be an $p \times p$-matrix with polynomial entries in the variables $Y_1, \ldots, Y_\ell$ of degrees bounded by q and integer coefficients of bit size at most ν. Then, $\det A$, considered as a polynomial in $Y_1, \ldots, Y_\ell$, has degrees in $Y_1, \ldots, Y_\ell$ bounded by $pq_1, \ldots, pq_\ell$, and coefficients of bit size at most $p\nu + p\mathsf{bit}(p) + \ell\mathsf{bit}(pq+1)$, where $q = \max(q_1, \ldots, q_\ell)$.*

We can now use Lemma 1 and Theorem 6 to prove the following result.

Lemma 2. *There exists two finite sets, denoted by $\mathcal{P}$ and $\mathcal{Q}$, which contain nonzero polynomials in the variables (λ, z) of degree at most nd in λ and d in z and integer coefficients, so that for each $(\lambda, z) \in (0,1] \times \mathbb{R}$, there exists $P \in \mathcal{P}$ and $Q \in \mathcal{Q}$ so that $\mathsf{val}\, W_\lambda^k(z) = P(\lambda, z)/Q(\lambda, z)$, $Q(\lambda, z) \neq 0$. Moreover, the coefficients of P are of bit size at most $3\varphi(n, d, N, 0)$.*

Proof. Let $(\mathbf{i}, \mathbf{j}) \in I \times J$ be fixed. By construction, $W_\lambda^k(z)[\mathbf{i}, \mathbf{j}] = d_\lambda^k(\mathbf{i}, \mathbf{j}) - z d_\lambda^0(\mathbf{i}, \mathbf{j})$, where $d_\lambda^k(\mathbf{i}, \mathbf{j})$ and $d_\lambda^0(\mathbf{i}, \mathbf{j})$ are the determinants of two $n \times n$ matrices whose entries are polynomial in λ of degree at most one and with coefficients in the set $\{0, \frac{1}{N}, \frac{2}{N}, \ldots, 1\}$. Consequently, $N^n W_\lambda^k(z)$ is a polynomial in λ and z, of degree at most n and 1, respectively, and integer coefficients whose bit size is at most $\nu := n\mathsf{bit}(n) + n\mathsf{bit}(N) + \mathsf{bit}(n+1)$ by Lemma 1. Let $\mathcal{P}$ and $\mathcal{Q}$ be, respectively, the sets of nonzero polynomials obtained as

$$P(\lambda, z) := \det\left(N^n \dot{W}_\lambda^k(z)\right) \quad \text{and} \quad Q(\lambda, z) := S\left(N^n \dot{W}_\lambda^k(z)\right),$$

when $\dot{W}_\lambda^k(z)$ ranges over all possible square submatrices of $W_\lambda^k(z)$. By Theorem 6, there exists a pair $(P, Q) \in \mathcal{P} \times \mathcal{Q}$ so that $Q(\lambda, z) \neq 0$ and

$$\mathsf{val}\, W_\lambda^k(z) = P(\lambda, z)/\left(N^n Q(\lambda, z)\right),$$

where the normalization of the denominator is due to the fact that, for any square matrix M of size $p \in \mathbb{N}$ and any $\alpha \in \mathbb{R}$, one has $\det(\alpha M) = \alpha^p \det(M)$, while $S(\alpha M) = \alpha^{p-1} S(M)$. We now show that P and Q satisfy the desired properties. First, P is nonzero, as $z \mapsto \mathsf{val}\, W_\lambda^k(z)$ is strictly decreasing by Theorem 4. Second, the submatrices of $W_\lambda^k(z)$ are of size at most d so that, by Lemma 1, all the polynomials in $\mathcal{P}$ and $\mathcal{Q}$ are of degree at most nd in λ and d in z, and their coefficients are integers. From Lemma 1, one also obtains a bound for the bit size of the coefficients of P, namely, $d\nu + d\mathsf{bit}(d) + 2\mathsf{bit}(nd+1)$. Replacing ν in the last expression we an expression that be easily bounded by $3\varphi(n, d, N, 0)$, which gives the desired result. □

We are now ready to prove the main result of this section. Again, we assume in the sequel that λ is a multiple of $1/M$ for some $M \in \mathbb{N}$.

Proposition 1. *The defining polynomials of v_λ^k and v^k are of degree at most d and have coefficients of bit size at most $4\varphi(n, d, N, M)$ and $4\varphi(n, d, N, 0)$, respectively.*

Proof. We start by proving the result for the discounted case. Let λ be such that $\lambda M \in \mathbb{N}$. Let $P \in \mathcal{P}$ and $Q \in \mathcal{Q}$ be the two polynomials given in Lemma 2 for $z = v_\lambda^k$. Hence, $Q(\lambda, v_\lambda^k) \neq 0$ and $\mathsf{val}\, W_\lambda^k(v_\lambda^k) = P(\lambda, v_\lambda^k)/Q(\lambda, v_\lambda^k)$. By Theorem 4, $\mathsf{val}\, W_\lambda^k(v_\lambda^k) = 0$, and, consequently, $P(\lambda, v_\lambda^k) = 0$ by choice of P. Now, as $\lambda M \in \mathbb{N}$ and $P(\lambda, z)$ is a nonzero polynomial of degree at most nd in λ and d in z with integer coefficients, the following expression

$$P_\lambda(z) := M^{nd} P(\lambda, z) \qquad \forall z \in \mathbb{R},$$

defines a nonzero polynomial of degree at most d with integer coefficients and satisfying $P_\lambda(v_\lambda^k) = 0$. Consequently, it is a multiple of the defining polynomial of v_λ^k. In particular, v_λ^k has algebraic degree at most d.

To bound the bit size of the coefficients of P_λ, it is enough to note that $\mathsf{bit}(M^{nd}C) \le nd\mathsf{bit}(M) + \mathsf{bit}(C)$ for any $C \in \mathbb{N}$, use the bound $3\varphi(n,d,N,0)$ obtained in Lemma 2 for P, and bound the bit-size of its factors with the Landau-Mignotte [19, theorem 2], which can be stated as follows: For any three polynomials A, B, C with integer coefficients, so that $A = \alpha_0 + \alpha_1 z \cdots + \alpha_a z^a$ and $B = \beta_0 + \beta_1 z + \cdots + \beta_b z^b$, and $A = BC$, one has

$$\max\left(\left|\beta_0\right|, \dots, \left|\beta_b\right|\right) \le \binom{b}{\lceil b/2 \rceil}\left(\alpha_0^2 + \cdots + \alpha_a^2\right)^{1/2}.$$

The Landau-Mignotte bound thus adds an additional term, $(d+1)\mathsf{bit}(d)$, to the previous bound. The result follows, then, from the relation $3\varphi(n,d,N,0) + (d+1)\mathsf{bit}(d) \le 4\varphi(n,d,N,0)$.

Consider now the undiscounted case. As already argued in the discounted case, for each $\lambda \in (0,1]$, there exists a nonzero polynomial $P \in \mathcal{P}$ of degree at most nd in λ and d in z (the choice of the polynomial depends on λ), with integer coefficients of bit size at most $3\varphi(n,d,N,0)$, and so that $P(\lambda, v_\lambda^k) = 0$. By finiteness of the set $\mathcal{P}$, and because two polynomials cannot intersect infinitely many times in $(0,1]$, one of these polynomials must satisfy $P(\lambda, v_\lambda^k) = 0$ for all λ sufficiently small. For this polynomial, denoted again by $P(\lambda, z)$, let $P_0, \dots, P_{nd}$ be the unique polynomials in z so that

$$P(\lambda, z) = P_0 + \lambda P_1(z) + \cdots + \lambda^{nd} P_{nd}(z) = 0.$$

As $P(\lambda, z)$ is nonzero, there exists $0 \le s \le nd$ and $P_s \neq 0$ so that

$$P(\lambda, z) = \lambda^s P_s(z) + o(\lambda^s).$$

By construction, P_s is a nonzero polynomial of degree at most d and has integer coefficients of bit size at most $3\varphi(n,d,N,0)$. Dividing by λ^s, and letting λ go to 0,

$$0 = \lim_{\lambda \to 0} \frac{P\left(\lambda, v_\lambda^k\right)}{\lambda^s} = P_s\left(v^k\right).$$

Hence, P_s is a multiple of the defining polynomial of v^k. Like in the discounted case, we obtain the desired bound from the Landau-Mignotte bound. □

3.1.1. Comments. This result, which relies on Theorems 4–6, improves the bound provided by Hansen et al. [11]. To see this, consider the case where $|I^\ell| = |J^\ell| = p$ for some $p \in \mathbb{N}$ and all $1 \le \ell \le n$ and $\lambda N \in \mathbb{N}$. In the case of Hansen et al. [11], bounded the algebraic degree of v_λ^k and v^k by $(2p+5)^n$, while Proposition 1 reduces the bound to $d = p^n$. Furthermore, this bound is tight. Our result also reduces the bound on the bit size of the coefficients obtained therein, from $22p^2n^2(2p+5)^n\mathsf{bit}(N)$ to $4np^n(\mathsf{bit}(n) + \mathsf{bit}(p) + \mathsf{bit}(N))$.

3.2. The Distance Between v_λ^k and v^k

In this section, we establish Theorem 3.

3.2.1. Classical Bounds. First of all, recall the following classical bounds from Cauchy [6] and Mahler [17] concerning the roots of polynomial.

Lemma 3. *Let $P(z) = a_0 + a_1 z \cdots + a_p z^\ell$ be a nonzero polynomial with integer coefficients, and let $\|P\|_\infty := \max(|a_0|, \dots, |a_p|)$ and $\|P\|_2 := (\sum_{r=0}^p a_r^2)^{1/2}$. Then,*

i. *If α is a root of P, then $\alpha \ge \frac{1}{2\|P\|_\infty}$.*

ii. *If $\beta \neq \alpha$ is another root of P, then $|\alpha - \beta| \ge p^{-(p+2)/2}\|P\|_2^{1-p}$.*

The following result is a direct consequence of Proposition 1 and Lemma 3(ii).

Lemma 4. *Let P be the defining polynomial of v_λ^k, and let $\varepsilon \le 2^{-8d\varphi(n,d,N,M)}$. Then, P has no root in the interval $(v_\lambda^k - \varepsilon, v_\lambda^k + \varepsilon)$. Similarly, let Q be the defining polynomial of v^k, and let $\varepsilon \le 2^{-8d\varphi(n,d,N,0)}$. Then, Q has no root in the interval $(v^k - \varepsilon, v^k + \varepsilon)$.*

Proof. Let us start by v_λ^k. By definition of the defining polynomial $P(v_\lambda^k) = 0$. By Proposition 1, P is of degree at most d, and its integer coefficients are bounded by $C := 2^{4\varphi(n,d,N,M)}$. Consequently, $\|P\|_2^2 \leq C^2(d+1)$ and, by Lemma 3(ii), any other root z of P satisfies

$$\begin{aligned}\left|z - v_\lambda^k\right| &\geq d^{-(d+2)/2}(d+1)^{(1-d)/2}C^{1-d} \\ &\geq 2^{-8d\varphi(n,d,N,M)}.\end{aligned}$$

This inequality proves the statement for v_λ^k. We omit the proof for v^k, as it goes along the exact same lines: It is enough to replace P, v_λ^k, and $\varphi(n,d,N,M)$ with Q, v^k, and $\varphi(n,d,N,0)$. □

Using Lemma 3(i), Lemma 2, and Theorem 6, we now derive some valuable insight on the asymptotic behavior of the sign of the map $\lambda \mapsto \operatorname{val} W_\lambda^k(z)$ as λ goes to 0, for a well-chosen fixed $z \in \mathbb{R}$. This result will be crucial in the proof of Theorem 3.

Proposition 2. *For any $r \in \mathbb{N}$, set $Z_r := \{0, \frac{1}{2^r}, \dots, \frac{2^r}{2^r}\}$ and $\lambda_r := 2^{-4\varphi(n,d,N,0)-rnd}$. Then, for each $z \in Z_r$,*

$$\begin{cases} \operatorname{val} W_{\lambda_r}^k(z) > 0 \Longrightarrow F^k(z) \in [0,+\infty] \\ \operatorname{val} W_{\lambda_r}^k(z) < 0 \Longrightarrow F^k(z) \in [-\infty,0] \\ \operatorname{val} W_{\lambda_r}^k(z) = 0 \Longrightarrow F^k(z) = 0. \end{cases}$$

Proof. Let $z \in Z_r$ be fixed. Let $\mathcal{P}$ and $\mathcal{Q}$ be the set of polynomials of Lemma 2. Hence, for all $P \in \mathcal{P}$, the polynomial $P(\lambda, z)$ is of degree at most nd in λ and d in z. Furthermore, by the choice of z,

$$P_z(\lambda) := 2^{rnd}P(\lambda, z) \qquad \lambda \in (0,1],$$

defines a polynomial in the variable λ of degree at most nd and with integer coefficients of bit size at most $3\varphi(n,d,N,0) + rnd + 1$. Let $\mathcal{P}(z)$ and $\mathcal{Q}(z)$ be the set of all the polynomials obtained this way, as P and Q range, respectively, in the sets $\mathcal{P}$ and $\mathcal{Q}$. By Theorem 6, for any $\lambda \in (0,1]$, there exists $P_z \in \mathcal{P}(z)$ and $Q_z \in \mathcal{Q}(z)$, the choice of the polynomials depends on λ, so that

$$\operatorname{val} W_\lambda^k(z) = \frac{P_z(\lambda)}{Q_z(\lambda)}.$$

Hence, a necessary condition for the function $\lambda \mapsto \operatorname{val} W_\lambda^k(z)$ to change sign at some $\alpha \in \mathbb{R}$ is that $P_z(\alpha) = 0$ for some polynomial $P_z \in \mathcal{P}(z)$. Applying Lemma 3(i) to the nonzero polynomials in $\mathcal{P}(z)$, it follows that neither of them admits a root in the interval $(0, \lambda_r]$. In other words, the sign of $\lambda \mapsto \operatorname{val} W_\lambda^k(z)$ is constant in the interval $(0, \lambda_r]$. Consider now the three possible cases, $\operatorname{val} W_{\lambda_r}^k(z) > 0$, $\operatorname{val} W_{\lambda_r}^k(z) < 0$, and $\operatorname{val} W_{\lambda_r}^k(z) = 0$. In the first case, $\lambda^{-n} \operatorname{val} W_\lambda^k(z) > 0$ for all $\lambda \in (0, \lambda_r]$ so that

$$F^k(z) := \lim_{\lambda \to 0} \lambda^{-n} \operatorname{val} W_\lambda^k(z) \in [0, +\infty].$$

The second case is similar. For the third, $\operatorname{val} W_{\lambda_r}^k(z) = 0$ implies that $\operatorname{val} W_\lambda^k(z) = 0$ for all $\lambda \in (0, \lambda_r]$, so that one also has $F^k(z) = 0$. □

3.2.2. Proof of Theorem 3. We are now ready to establish Theorem 3, whose statement is as follows: *For each $r \in \mathbb{N}$, let $\lambda_r := 2^{-4\varphi(n,d,N,0)-rnd}$. Then, $|v_\lambda^k - v^k| \leq 2^{-r}$ for all $\lambda \in (0, \lambda_r)$.*

Let $\lambda \in (0, \lambda_r]$ be fixed. First of all, the maps $z \mapsto \operatorname{val} W_\lambda^k(z)$ and $z \mapsto F^k(z)$ are strictly decreasing, by Theorems 4 and 5, respectively. Therefore, either there exists a unique $z \in Z_r = \{0, \frac{1}{2^r}, \dots, \frac{2^r}{2^r}\}$ so that $\operatorname{val} W_\lambda^k(z) = 0$, or there exists $0 \leq m \leq 2^r$ such that $\operatorname{val} W_\lambda^k(m2^{-r}) > 0$ and $\operatorname{val} W_\lambda^k((m+1)2^{-r}) < 0$, and the same is true for F^k. Consider the first case, and let $z \in Z_r$ satisfy $\operatorname{val} W_\lambda^k(z) = 0$. By Theorem 4, this implies $z = v_\lambda^k$, so that, by Proposition 2, one also has $F^k(z) = 0$. But, then, Theorem 5 implies $v^k = z$, so that $v_\lambda^k = v^k$, and the inequality $|v_\lambda^k - v^k| \leq 2^{-r}$ holds. Consider now the second case, and let $1 \leq m \leq 2^r$ be such that $\operatorname{val} W_\lambda^k(m2^{-r}) > 0$ and $\operatorname{val} W_\lambda^k((m+1)2^{-r}) < 0$. On the one hand, Theorem 4 implies

$$m2^{-r} < v_\lambda^k < (m+1)2^{-r}. \tag{2}$$

On the other, Proposition 2 gives $F^k(m2^{-r}) \geq 0$ and $F^k((m+1)2^{-r}) \leq 0$, which, in view of Theorem 5, implies

$$m2^{-r} \leq v^k \leq (m+1)2^{-r}. \tag{3}$$

The combination of (2) and (3) yields the desired inequality $|v^k_\lambda - v^k| \leq 2^{-r}$. □

4. Algorithms

The aim of this section is to describe the algorithms that correspond to Theorems 1 and 2. We start by recalling three classical algorithms in Section 4.1 that are called by the above-mentioned algorithms. Section 4.2 is devoted to the description of two algorithms: The first one outputs arbitrarily close approximations for the discounted values of a stochastic game, while the second one outputs an exact expression for this value. The latter corresponds to the algorithm of Theorem 1. Similarly, Section 4.2 is devoted to the description of three algorithms: The first two output arbitrarily close approximations for the value of a stochastic game, while the third one outputs an exact expression for this value. The latter corresponds to the algorithm of Theorem 2.

4.1. Auxiliary Results

Recall that the complexity of the algorithms is measured with the logarithmic cost model. The logarithmic cost of an algorithm can be bounded by (1) a bound of the number of arithmetic operations that it requires, and (2) a bound of the bit size of the numbers that are involved in them. In particular, if these two bounds are polynomial expressions in some variables $t_1, \ldots, t_m$, so is the logarithmic cost of the algorithm.

We now recall three well-known algorithms. The first one, because of Kannan et al. [13], allows us to compute the defining polynomial of an algebraic number efficiently. It will be referred as the *KLL algorithm*. The second, because of Karmarkar [15], allows us to solve linear programs efficiently and will be referred to as the *Karmarkar algorithm*. The third one, which allows us to compute the determinant of a square matrix efficiently, is taken from Basu et al. [2], where it is referred to as the *Dodgson-Jordan-Bareiss algorithm*.

Theorem 7. *Let $\alpha \in \mathbb{R}$ be an algebraic number of degree $p \in \mathbb{N}$ and defining polynomial $P(z) = a_0 + a_1 z + \cdots + a_p z^p$, $z \in \mathbb{R}$. The KLL algorithm outputs the defining polynomial of $\alpha \in \mathbb{R}$ when given as inputs $(D, C, \bar{\alpha})$ satisfying*

- *$B \in \mathbb{N}$ is a bound on the algebraic degree of α, that is, versus $p \leq D$.*
- *$2^C \in \mathbb{N}$ bounds the integer coefficients $|a_0|, \ldots, |a_p|$.*
- *$\bar{\alpha} \in \mathbb{Q}$ so that $|\alpha - \bar{\alpha}| \leq 2^{-s}\frac{1}{12D}$, where*

$$s = s(b, C) := \lceil D^2/2 + (3D+4)\log_2(D+1) + 2DC \rceil.$$

This algorithm requires $O(pD^4(D+C))$ arithmetic operations on integers of bit-size $O(D^2(D+C))$.

Theorem 8. *Let M be a $p \times q$ matrix with rational entries that can be encoded in C bits. The Karmarkar algorithm inputs M and outputs its value $\mathrm{val}\, M$ and requires $O(p^{3.5}C)$ arithmetic operations on integers of bit size at most $O(C)$.*

Theorem 9. *Let M be a $p \times p$ matrix with integer entries of bit-size C. The Dodgson-Jordan-Bareiss algorithm inputs M and outputs $\det M$ and requires $O(p^3)$ arithmetic operations on integers of bit-size $O(p\,\mathrm{bit}(p)C)$.*

For the three above-mentioned algorithms, the following assertions hold.

- The *KLL algorithm* is polynomial in D and C, the bounds for the degree and the bit size of the coefficients, respectively, of the defining polynomial of α.
- The *Karmarkar algorithm* and the *Dogson-Jordan-Bareiss algorithm* are polynomial in the size of the matrix and in the bit size of their entries.

4.2. Computing the Discounted Values

We start by describing a bisection algorithm, directly derived from Theorem 4, that outputs arbitrarily close approximations of the λ-discounted value v^k_λ of a stochastic game (K, I, J, g, q, k). As we will show later on (see 4.3.3), this algorithm can also be used to obtain arbitrarily close approximations of v^k, thanks to Theorem 3.

4.2.1. Algorithm 1 Approx

The algorithm goes as follows.

Input: A stochastic game (K, I, J, g, q, k) satisfying (H_N) for some $N \in \mathbb{N}$, a discount rate λ satisfying $\lambda M \in \mathbb{N}$ for some $M \in \mathbb{N}$, and a precision level $r \in \mathbb{N}$.

Output: A rational number $u \in \{0, \frac{1}{2^r}, \dots, \frac{2^r-1}{2^r}\}$ so that $v_\lambda^k \in [u, u + \frac{1}{2^r}]$.

Computation cost: Polynomial in n, $|I|$, $|J|$, $\log N$, $\log M$ and r.

1. Set $\underline{w} := 0$, $\overline{w} := 1$
2. WHILE $\overline{w} - \underline{w} > 2^{-r}$ DO
   2.1 $z := \frac{\underline{w}+\overline{w}}{2}$
   2.2 Compute $W_\lambda^k(z)$
   2.3 Compute $v := \operatorname{val} W_\lambda^k(z)$
   2.4 IF $v \geq 0$, THEN $\underline{w} := z$
   2.5 IF $v \leq 0$, THEN $\overline{w} := z$
3. RETURN $u := \underline{w}$.

4.2.2. Computation Cost of Algorithm 1 Approx. By Theorem 4, each iteration of step 2 reduces the interval $[\underline{w}, \overline{w}]$ by a factor of 1/2, while satisfying $\underline{w} \leq v_\lambda^k \leq \overline{w}$. Consequently, the algorithm terminates after at most r steps, and the output u satisfies $v_\lambda^k \in [u, u + \frac{1}{2^r}]$. As steps 2.1, 2.4, and 2.5 require one operation each, the computation cost of the algorithm depends essentially on the computation cost of steps 2.2 and 2.3, which is the object of the following lemma.

Lemma 5. *Let $L = O(n\mathsf{bit}(n)\mathsf{bit}(N)\mathsf{bit}(M))$. For all $r \in \mathbb{N}$ and $z \in \{0, \frac{1}{2^r}, \dots, \frac{2^r-1}{2^r}\}$,*

i. *The computation of $W_\lambda^k(z)$ with the Dogson-Jordan-Bareiss algorithm requires $O(n^3|I||J|)$ arithmetic operations, with numbers of bit-size $O(L+r)$.*

ii. *The computation of $\mathsf{val}\, W_\lambda^k(z)$ with the Karmarkar algorithm requires $O(d^{3.5}(L+r))$ arithmetic operations, with numbers of bit-size $O(L+r)$.*

Proof. Let $r \in \mathbb{N}$ and $z \in \{0, \frac{1}{2^r}, \dots, \frac{2^r-1}{2^r}\}$ be fixed.

i. Thanks to the assumptions (H_N) and $\lambda M \in \mathbb{N}$, and by the definition of $W_\lambda^k(z)$, for each $(\mathbf{i}, \mathbf{j}) \in I \times J$, $W_\lambda^k(z)[\mathbf{i}, \mathbf{j}] = d_\lambda^0(\mathbf{i}, \mathbf{j}) - z d_\lambda^0(\mathbf{i}, \mathbf{j})$, where $d_\lambda^0(\mathbf{i}, \mathbf{j})$ and $d_\lambda^k(\mathbf{i}, \mathbf{j})$ are determinants of some $n \times n$ matrices whose entries are multiples of $\frac{1}{NM}$. Multiplying each entry by NM, so that all entries are integers, it follows then from Theorem 9 that $d_\lambda^0(\mathbf{i}, \mathbf{j})$ and $d_\lambda^0(\mathbf{i}, \mathbf{j})$ can be computed in $O(n^3)$ arithmetic operations on integers of bit-size $O(n\mathsf{bit}(n)\mathsf{bit}(NM)) = O(L)$. The entries of $W_\lambda^k(z)$ are then of bit size at most $O(L+r)$ because, by the choice of z, $\mathsf{bit}(z) = O(r)$. Finally, the total number of operations is simply $O(n^3|I||J|)$ because the matrix $W_\lambda^k(z)$ is of size $|I| \times |J|$.

ii. As already noted in the proof of (*i*), the entries $W_\lambda^k(z)$ are of bit size at most $O(L+r)$. The result follows then directly from Theorem 8. □

The next result is a direct consequence from Lemma 5.

Theorem 10. *Algorithm 1 Approx computes a 2^{-r}-approximation of v_λ^k for any $r \in \mathbb{N}$, and its computation cost is polynomial in n, $|I|$, $|J|$, $\log N$, $\log M$, and r.*

4.2.3. Algorithm 1 Exact

Next, we combine Algorithm 1 Approx and the KLL algorithm in order to obtain an exact expression for v_λ^k.

Input: A stochastic game (K, I, J, g, q, k) satisfying (H_N) for some $N \in \mathbb{N}$, and a discount rate $\lambda \in (0, 1]$ so that $\lambda M \in \mathbb{N}$ for some $M \in \mathbb{N}$.

Output: An exact expression for v_λ^k.

Computation cost: Polynomial in n, $|I|$, $|J|$, $\log N$ and $\log M$.

1. Initialization phase:
   1.1 Set $C := 4\varphi(n, d, N, M)$
   1.2 Set $s := \lceil d^2/2 + (3d+4)\log_2(d+1) + 2dC \rceil$
   1.3 Set $r := s\lceil \log_2 12d \rceil$
2. Run Algorithm 1 Approx with inputs (K, I, J, g, q, k), the discount rate λ, and precision level r. Denote its output by u.
3. Run the KLL algorithm with inputs d, C and u, and output P.
4. RETURN $(P; u, u + 2^{-r})$.

4.2.4. Proof of Theorem 1. To prove this result, it is enough to show that Algorithm 1 Exact computes an exact expression for v_λ^k and that its computation cost is polynomial in n, $|I|$, $|J|$, $\log N$, and $\log M$. To do so, recall that the algebraic degree of v_λ^k and the bit size of the coefficients of its defining polynomial are bounded by d and C, respectively, by Proposition 1. Second, by Theorem 10, Algorithm 1 Exact, step 2 returns u so that $\mathsf{bit}(u) \le 2r$ and $|u - v_\lambda^k| \le 2^{-r}$, and the computation cost is polynomial in n, $|I|$, $|J|$, $\log N$, $\log M$, and r. Third, by Theorem 7, the definition of C, r and s in Algorithm 1 Exact, step 1 ensure that Algorithm 1 Exact, step 3 provides the defining polynomial P of v_λ^k and that the computation cost is polynomial in d and C. As C and r are (bounded by) polynomial expressions in n, d, and $\log N$, the entire algorithm is thus polynomial in n, $|I|$, $|J|$, $\log N$, and $\log M$. It remains to show that P has no other root than v_λ^k in the interval $(u, u+2^{-r})$ so that $(P; u, u+2^{-r})$ is an exact expression for v_λ^k. To see this, note that, by definition, one has $r \ge 8d\varphi(n,d,N,M)$. By Lemma 4, this implies that P has no other root in the interval $(v_\lambda^k - 2^{-r}, v_\lambda^k + 2^{-r})$, and the result follows because this interval contains $(u, u+2^{-r})$ thanks to $|u - v_\lambda^k| \le 2^{-r}$. □

4.3. Computing the Value

Like for the discounted case, we start by proposing a bisection algorithm, directly derived from Theorem 5, which outputs arbitrarily close approximations of the value v^k of a stochastic game (K,I,J,g,q,k). Note, however, that the natural algorithm would consist in computing the sign of $F^k(z) := \lim_{\lambda\to 0} \lambda^{-n} \mathsf{val}\, W_\lambda^k(z)$ at each iteration, but this computation seems very costly. Luckily, there is a way out to this issue. Indeed, by Proposition 2, this computation is equivalent to that of the sign of $\mathsf{val}\, W_\lambda^k(z)$, for a well-chosen λ, and this can be done efficiently because it is a linear program (provided that the bit size of λ is polynomial). The following bisection algorithm is built upon this observation.

4.3.1. Algorithm 2 Approx

The algorithm goes as follows.

Input: A stochastic game (K,I,J,g,q,k) satisfying (H_N), and a precision level $r \in \mathbb{N}$.
Output: A rational number $u \in \{0, \frac{1}{2^r}, \dots, \frac{2^r-1}{2^r}\}$ so that $v^k \in [u, u+\frac{1}{2^r}]$.
Computation cost: Polynomial in n, $|I|$, $|J|$, $\log N$, and r.

1.1. Set $\lambda_r := 2^{-4\varphi(n,d,N,0)-rnd}$
1.2. Set $\underline{w} := 0$, $\overline{w} := 1$
2. WHILE $\overline{w} - \underline{w} > 2^{-r}$ DO
  2.1 $z := \frac{\underline{w}+\overline{w}}{2}$
  2.2 Compute $W_{\lambda_r}^k(z)$
  2.3 Compute $v := \mathsf{val}\, W_{\lambda_r}^k(z)$
  2.4 IF $v \ge 0$, THEN $\underline{w} := z$
  2.5 IF $v \le 0$ THEN $\overline{w} := z$
3. RETURN $u := \underline{w}$.

4.3.2. Computation Cost of Algorithm 2 Approx. The next result is a direct consequence of Proposition 2, Lemma 5, and the definition of λ_r.

Theorem 11. *Algorithm 2 Approx computes a 2^{-r}-approximation of v^k for any $r \in \mathbb{N}$, and its computation cost is polynomial in n, $|I|$, $|J|$, $\log N$, and r.*

Proof. By Proposition 2, the sign of $W_{\lambda_r}^k(z)$ coincides with the sign of $F^k(z) = \lim_{\lambda\to 0} \lambda^{-n} \mathsf{val}\, W_\lambda^k(z)$ at every z that is called by the algorithm. It follows, then, from Theorem 5 that Algorithm 2 Approx provides a 2^{-r}-approximation of v^k. By Lemma 5, its computation cost is polynomial in n, d, $\mathsf{bit}(N)$, and $\mathsf{bit}(\lambda_r)$. The result follows, then, from the fact that the bit size of λ_r is polynomial in n, d, $\log N$, and r. □

4.3.3. Algorithm 2 Approx Bis

Alternatively, one can use Theorem 3 to obtain arbitrary close approximations of v^k directly from Algorithm 1 Approx, as follows.

Input: A stochastic game (K,I,J,g,q,k) satisfying (H_N), and a precision level $r \in \mathbb{N}$.
Output: A rational number $u \in \{0, \frac{1}{2^r}, \dots, \frac{2^r-1}{2^r}\}$, so that $v^k \in [u, u+\frac{1}{2^r}]$.
Computation cost: Polynomial in n, $|I|$, $|J|$, $\log N$, and r.

1. Set $\lambda_{r+1} := 2^{-4\varphi(n,d,N,0)-(r+1)nd}$.
2. Run Algorithm 1 Approx with inputs the stochastic game (K,I,J,g,q,k), the discount rate λ_{r+1}, and the precision level $r+1$. Let u denote its output.
3. RETURN u.

4.3.4. Computation Cost of Algorithm 2 Approx Bis.

Theorem 12. *Algorithm 2 Approx Bis computes a 2^{-r}-approximation of v^k for any $r \in \mathbb{N}$, and its computation cost is polynomial in n, $|I|$, $|J|$, $\log N$, and r.*

Proof. By Theorem 10, Algorithm 2 Approx Bis, step 2 outputs u so that $|u - v^k_{\lambda_{r+1}}| \leq \frac{1}{2^{r+1}}$, and the cost is polynomial in $n, |I|, |J|, \log N, r$, and $\mathsf{bit}(\lambda_r)$. The latter being polynomial $n, |I|, |J|, \log N$, and r, the cost is thus polynomial in these variables, too. Finally, by the choice of λ_{r+1}, Theorem 3 implies $|v^k - v^k_{\lambda_{r+1}}| \leq \frac{1}{2^{r+1}}$. The result follows, because

$$\left|u - v^k\right| \leq \left|u - v^k_{\lambda_{r+1}}\right| + \left|v^k - v^k_{\lambda_{r+1}}\right| \leq \frac{1}{2^r}. \quad \square$$

4.3.5. Algorithm 2 Exact

Like in the discounted case, one can now combine Algorithm 2 Approx (or Algorithm 2 Approx Bis) with the KLL algorithm to obtain an algorithm that outputs an exact expression for v^k. The algorithm goes as follows.

Input: A finite stochastic game (K, I, J, g, q, k) satisfying (H_N) for some $N \in \mathbb{N}$.
Output: An exact expression for v^k.
Computation cost: Polynomial in n, $|I|$, $|J|$, and $\log N$.

1. Initialization phase:
   1.1 Set $C := 4\varphi(n, d, N, 0)$.
   1.2 Set $s := \lceil d^2/2 + (3d + 4)\log_2(d + 1) + 2dC \rceil$
   1.3 Set $r := s\lceil \log_2 12d \rceil$
2. Run Algorithm 2 Approx Bis with inputs the stochastic game (K, I, J, g, q, k) and the precision level $r + 1$. Denote its output by u.
3. Run the KKL algorithm with inputs d, C and u. Denote its output by Q.
4. RETURN $(Q; u, u + 2^{-r})$.

4.3.6. Proof of Theorem 2. We are now ready to prove Theorem 2. More precisely, we show that Algorithm 2 Exact computes an exact expression for v^k, and that (its computation cost) is polynomial in n, $|I|$, $|J|$, and $\log N$. The proof is similar to that of Theorem 1. First, recall that the algebraic degree of v^k and the bit size of the coefficients of its defining polynomial are bounded by d and C, respectively, by Proposition 1. Second, by Theorem 12, Algorithm 2 Exact, step 2 returns u so that $\mathsf{bit}(u) \leq 2r$ and $|u - v^k| \leq 2^{-r}$, and the computation cost is polynomial in n, $|I|$, $|J|$, $\log N$, and r. Third, by Theorem 7, the definition of C, r, and s in Algorithm 1 Exact, step 1 ensure that Algorithm 2 Exact, step 3 provides the defining polynomial Q of v^k, and that the computation cost is polynomial in d and C. As C and r are (bounded by) polynomial expressions in n, d and $\log N$, the entire algorithm is thus polynomial in n, $|I|$, $|J|$, and $\log N$. It remains to show that Q has no other root than v^k in the interval $(u, u + 2^{-r})$ so that $(Q; u, u + 2^{-r})$ is an exact expression for v^k. To see this, note that, by definition, one has $r \geq 8d\varphi(n, d, N, 0)$. By Lemma 4, this implies that Q has no other root in the interval $(v^k - 2^{-r}, v^k + 2^{-r})$, and the result follows because this interval contains $(u, u + 2^{-r})$ thanks to $|u - v^k| \leq 2^{-r}$. $\square$

4.4. An Example

We now provide an example to illustrate our results. The stochastic game goes back to Kohlberg [4]. It is played over four states, two of which are absorbing—that is, once these states are reached, the state can no longer change. The game can be described as follows, where each box represents a state, arrows indicate deterministic transitions from the current state to the state that is pointed, and numbers indicate stage rewards:

$$\begin{array}{|c|c|}\hline 1 \circlearrowleft & 0 \rightarrow \\ \hline 0 \rightarrow & 0 \downarrow \\ \hline \end{array} \qquad \begin{array}{|c|c|}\hline -1 \circlearrowleft & 0 \leftarrow \\ \hline 0 \leftarrow & 0 \downarrow \\ \hline \end{array},$$

$$\begin{array}{|c|}\hline 1 \circlearrowleft \\ \hline \end{array} \qquad \begin{array}{|c|}\hline -1 \circlearrowleft \\ \hline \end{array}.$$

In this example, one has $d = |I| = |J| = 4$, so that $\mathsf{bit}(d) = 3$, $N = \mathsf{bit}(N) = 1$, and $n = 4$. However, it is easy to see that the bounds obtained in Section 3 do not depend on n, but, rather, on $\bar{n}$, the number of nonabsorbing states. We focus on the initial state $k = 1$.

4.4.1. Computation of a 0.05-Approximation of v^1. First of all, $2^{-5} < 0.05 < 2^{-4}$, so we need our algorithm to compute a 2^{-r}-approximation with $r = 5$. By Proposition 1, the defining polynomial of v^1 is of degree at most $\bar{n}d = 8$, and the bit size of its coefficients are bounded by $C := 4\varphi(\bar{n}, d, N, 0) = 192$. By Algorithm 2 Approx Bis, all we need to do is set $\lambda_{r+1} := 2^{-C-(r+1)\bar{n}d} = 2^{-240}$, and then run Algorithm 1 Approx with input λ_{r+1} and precision $r + 1 = 6$. The latter will compute the value of the following $|I| \times |J|$ matrix

$$W^1_\lambda(z) = \lambda^2 \begin{pmatrix} \lambda^2(1-z) & \lambda(1-z) & -\lambda(1-\lambda) - \lambda z & -(2\lambda - \lambda^2)z \\ \lambda(1-z) & \lambda(1-z) & -(2\lambda-\lambda^2)z & -(1-\lambda)^2 - z \\ -\lambda(1-\lambda) - \lambda z & -(2\lambda - \lambda^2)z & \lambda(1-z) - \lambda z & 1 - \lambda - z \\ -(2\lambda - \lambda^2)z & -(1-\lambda)^2 - z & 1 - \lambda - z & 1 - \lambda - z \end{pmatrix},$$

for $\lambda = 2^{-240}$ and well-chosen z in the set $\{m2^{-6}, 0 \le m \le 2^6\}$. The output u will then satisfy $|u - v^1_\lambda| \le 2^{-6}$, and Theorem 3 gives then $|v^1_\lambda - v^1| \le 2^{-6}$, so that $|u - v^1| \le 2^{-5}$.

4.4.2. Computation of an Exact Expression of v^1. By Theorem 2, this can be done with Algorithm 2 Exact, which starts by setting $C := 4\varphi(\bar{n}, d, N, 0) = 192$, $s := \lceil d^2/2 + (3d+4)\log_2(d+1) + 2dC \rceil = 1582$ and $r := s\lceil \log_2 12d \rceil = 8836$, before calling Algorithm 2 Approx Bis with precision level $r + 1$. The latter sets then $\lambda_{r+1} = 2^{-C-(r+1)\bar{n}d} = 2^{-70880}$ and runs Algorithm 1 Approx with input $\lambda = 2^{-70880}$ and precision $r + 1$. The output is denoted u, and satisfies $|u - v^1| \le 2^{-8836}$. Finally, one needs to run the KKL algorithm with inputs d, C and u to obtain the defining polynomial Q of v^1, which together with u and $u + 2^{-8836}$ give the desired exact expression for v^1.

Acknowledgments
The author thanks Krishnendu Chatterjee for his useful comments and time and Kristoffer Hansen for his insight and advice. The author also thanks anonymous referees of the journal for comments, which have greatly contributed in the presentation and organization of the results.

References
[1] Attia L, Oliu-Barton M (2019) A formula for the value of a stochastic game. *Proc. Natl. Acad. Sci. USA* 116(52):26435–26443.
[2] Basu S, Pollack R, Roy M-F (2007) *Algorithms in Real Algebraic Geometry*, Algorithms and Computation in Mathematics, vol. 10 (Springer Science & Business Media, New York).
[3] Bewley T, Kohlberg E (1976) The asymptotic theory of stochastic games. *Math. Oper. Res.* 1(3):197–208.
[4] Bewley T, Kohlberg E (1978) On stochastic games with stationary optimal strategies. *Math. Oper. Res.* 3(2):104–125.
[5] Blackwell D, Ferguson TS (1968) The big match. *Ann. Math. Statist.* 39(1):159–163.
[6] Cauchy A (1832) Calcul des indices des fonctions. *J. l'École Polytechnique* 15(25):176–229.
[7] Chatterjee K, Majumdar R, Henzinger TA (2008) Stochastic limit-average games are in EXPTIME. *Internat. J. Game Theory* 37:219–234.
[8] Condon A (1992) The complexity of stochastic games. *Inform. Comput.* 96(2):203–224.
[9] Etessami K, Yannakakis M (2006) Recursive concurrent stochastic games. Bugliesi M, Preneel B, Sassone V, Wegener I, eds. *Internat. Colloq. Automata Languages Programming. Part II*, Lecture Notes in Computer Science, vol. 4052 (Springer, Berlin), 324–335.
[10] Hansen KA, Ibsen-Jensen R, Miltersen PB (2011) The complexity of solving reachability games using value and strategy iteration. Kulikov AS, Vereshchagin NK, eds. *Internat. Comput. Sci. Sympos. Russia* (Springer, Berlin), 77–90.
[11] Hansen KA, Koucký M, Lauritzen N, Miltersen PB, Tsigaridas EP (2011) Exact algorithms for solving stochastic games. *Proc. 43rd Annual ACM Sympos. Theory Comput.* (ACM, New York), 205–214.
[12] Henzinger TA, de Alfaro L, Chatterjee K (2006) Strategy improvement for concurrent reachability games. *Third Internat. Conf. Quant. Evaluation Systems (QEST)* (IEEE, New York), 291–300.
[13] Kannan R, Lenstra AK, Lovász L (1988) Polynomial factorization and nonrandomness of bits of algebraic and some transcendental numbers. *Math. Comput.* 50(18):235–250.
[14] Kaplansky I (1945) A contribution to von Neumann's theory of games. *Ann. Math.* 46(3):474–479.
[15] Karmarkar N (1984) A new polynomial-time algorithm for linear programming. *Proc. 16th Annual ACM Sympos. Theory Comput.* (ACM, New York), 302–311.
[16] Kohlberg E (1974) Repeated games with absorbing states. *Ann. Statist.* 2(4):724–738.
[17] Mahler K (1964) An inequality for the discriminant of a polynomial. *Michigan Math. J.* 11(3):257–262.
[18] Mertens J-F, Neyman A (1981) Stochastic games. *Internat. J. Game Theory* 10:53–66.
[19] Mignotte M (1974) An inequality about factors of polynomials. *Math. Comput.* 28(12):1153–1157.
[20] Oliu-Barton M (2014) The asymptotic value in stochastic games. *Math. Oper. Res.* 39(3):712–721.
[21] Rao S, Chandrasekaran R, Nair KPK (1973) Algorithms for discounted stochastic games. *J. Optimization Theory Appl.* 11:627–637.
[22] Renault J (2019) A tutorial on zero-sum stochastic games. Preprint, submitted May 16, https://arxiv.org/abs/1905.06577.
[23] Shapley LS (1953) Stochastic games. *Proc. Natl. Acad. Sci. USA* 39(10):1095–1100.
[24] Solan E, Vieille N (2010) Computing uniformly optimal strategies in two-player stochastic games. *Econom. Theory* 42:237–253.
[25] Sorin S (2002) *A First Course on Zero-Sum Repeated Games*, Mathématiques et Applications, vol. 37 (Springer Science & Business Media, New York).