Innovations in Computer Science 2011

Strategy Iteration is Strongly Polynomial for 2-Player Turn-based Stochastic Games with a Constant Discount Factor

Thomas Dueholm Hansen[1*] Peter Bro Miltersen[1*] Uri Zwick[2†]
[1] Department of Computer Science, Aarhus University, Aarhus N 8200, Denmark
[2]School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel
tdh@cs.au.dk bromille@cs.au.dk zwick@tau.ac.il

Abstract: Ye showed recently that the *simplex method* with Dantzig pivoting rule, as well as Howard's *policy iteration* algorithm, solve discounted *Markov decision processes* (MDPs), with a constant discount factor, in *strongly polynomial* time. More precisely, Ye showed that both algorithms terminate after at most $O\big(\frac{mn}{1-\gamma}\log\big(\frac{n}{1-\gamma}\big)\big)$ iterations, where n is the number of states, m is the total number of actions in the MDP, and $0<\gamma<1$ is the discount factor. We improve Ye's analysis in two respects. First, we improve the bound given by Ye and show that Howard's policy iteration algorithm actually terminates after at most $O\big(\frac{m}{1-\gamma}\log\big(\frac{n}{1-\gamma}\big)\big)$ iterations. Second, and more importantly, we show that the same bound applies to the number of iterations performed by the *strategy iteration* (or *strategy improvement*) algorithm, a generalization of Howard's policy iteration algorithm used for solving 2-player turn-based *stochastic games* with discounted zero-sum rewards. This provides the first strongly polynomial algorithm for solving these games, resolving a long standing open problem.

Keywords: Markov decision processes, turn-based stochastic games, strategy iteration, policy iteration, strongly polynomial upper bound.

1 Introduction

Markov Decision Processes (MDPs) are widely used in operations research, machine learning and related disciplines, to model long-term sequential decision making in uncertain, i.e., stochastic, environments. *Stochastic Games* (SGs), a generalization of MDPs to a 2-player setting, are widely used to model long-term sequential decision making in stochastic and adversarial environments. MDPs were first introduced by Bellman [2]. SGs, which form a more general model, were introduced slightly earlier by Shapley [32]. Many variants of MDPs and SGs were studied in the literature. The MDPs and SGs considered in this paper are infinite-horizon *discounted* MDPs/SGs. The SGs we consider are *turn-based* and we thus refer to them as *2-player Turn-Based Stochastic Games* (2TBSG). MDPs may be viewed as degenerate 2TBSGs in which one of the players has no influence on the game. For a thorough treatment of MDPs and their numerous practical applications, see the books of Howard [18], Derman [9], Puterman [29] and Bertsekas [3]. For a similar treatment of SGs, see the books of Filar and Vrieze [13] and Neyman and Sorin [28].

A 2TBSGs is composed of a finite set of *states* and a finite set of *actions*. Each state is controlled by one of the players. In each time unit, the game is in exactly one of the states. Each state has a non-empty set of actions associated with it. The player controlling the state must play one of these actions. Playing an action incurs an immediate *cost*, and results in a probabilistic transition to a new state according to a probability distribution that depends on the action. The process goes on indefinitely. The first player tries to *minimize* the total expected *discounted* cost of the infinite sequence of actions taken, with respect to a fixed *discount factor*. The second player tries to *maximize* this total discounted cost. Discounting captures the fact that a cost incurred at a later stage has a smaller effect than the same cost incurred at an earlier stage. For formal definitions, see Section 2.

* Supported by the Center for Algorithmic Game Theory, funded by the Carlsberg Foundation.

† Supported by grant 1306/08 of the Israeli Science Foundation.

A *policy* or a *strategy* for a player is a possibly probabilistic rule that specifies the action to be taken in each situation, given the full history of play so far. One of the fundamental results in the theory of MDPs and 2TBSGs, is that both players have *positional* optimal strategies. A positional strategy is a strategy that is both *deterministic* and *memoryless*. A *memoryless* strategy is a strategy that depends only on the current state, and not on the full history. MDPs and 2TBSGs are *solved* by finding optimal positional strategies for the players.

MDPs can be solved using linear programming (d'Epenoux [8], Derman [9]). The preferred way of solving MDPs in practice, however, is Howard's [18] *Policy Iteration* algorithm. The policy iteration algorithm maintains and iteratively improves a policy by performing "obvious" improving switches (for details, see Section 5). Howard's algorithm may be viewed as a parallel version of the simplex algorithm in which several pivoting steps are performed simultaneously. The problem of determining the worst case complexity of Howard's algorithm was stated explicitly at least 25 years ago. (It is mentioned, among other places, in Schmitz [31], Littman *et al.* [23] and Mansour and Singh [25].) Meister and Holzbaur [27] established, decades ago, that the number of iterations performed by Howard's algorithm, when the discount factor is fixed, is polynomially bounded in the *bit size* of the input. Their bound, however, is not polynomial in the number of states and actions of the MDP. The first *strongly polynomial* time algorithm for solving MDPs was an interior point algorithm of Ye [34].

Very recently, Ye [35] presented a surprisingly simple proof that Howard's algorithm terminates after at most $O\big(\frac{mn}{1-\gamma}\log\big(\frac{n}{1-\gamma}\big)\big)$ iterations, where n is the number of states, m is the total number of actions, and $0 < \gamma < 1$ is the discount factor. In particular, when the discount factor is constant, the number of iterations is $O(mn \log n)$. Since each iteration only involves solving a system of linear equations, Ye's result established for the first time that Howard's algorithm is a *strongly polynomial* time algorithm, when the discount factor is constant. Ye's proof is based on a careful analysis of an LP formulation of the MDP problem, with LP duality and complementary slackness playing crucial roles.

We significantly improve and extend Ye's [35] analysis. We show that Howard's algorithm actually terminates after at most $O\big(\frac{m}{1-\gamma}\log\big(\frac{n}{1-\gamma}\big)\big)$ iterations, improving Ye's bound by a factor of n. Interestingly, the only added ingredient needed to obtain this significant improvement is a well-known relationship between Howard's policy iteration algorithm and Bellman's [2] *value iteration* algorithm, an algorithm for *approximating* the values of MDPs.

More significantly, and more surprisingly, we are able to obtain the same $O\big(\frac{m}{1-\gamma}\log\big(\frac{n}{1-\gamma}\big)\big)$ bound also for the *Strategy Iteration* (or *Strategy Improvement*) algorithm for the solution of 2TBSGs. This supplies the *first* strongly polynomial algorithm for solving 2TBSGs, with a fixed discount factor, solving a long standing open problem.

The strategy iteration algorithm is a natural generalization of Howard's policy iteration algorithm that can be used to solve 2TBSGs. The strategy iteration algorithm for discounted 2-player games is apparently first described by Rao *et al.* [30]. Hoffman and Karp [17] earlier described a related algorithm for a somewhat different class of SGs.

Prior to our strongly polynomial bound for the strategy iteration algorithm, the best time available on the problem of solving discounted 2TBSGs was a polynomial, but not strongly polynomial, bound of Littman [22], obtained essentially using value iteration. The best time bound expressed solely in terms of the number states and actions was a *subexponential* bound of Ludwig [24]. (See also Björklund and Vorobyov [4, 5] and Halman [16].) Interestingly, these subexponential bounds are obtained using randomized variants of the strategy iteration algorithm that mimic the combinatorial subexponential algorithms of Kalai [20, 21] and Matoušek, Sharir and Welzl [26] for solving *LP-type* problems.

What makes our analysis of the strategy iteration algorithm surprising is the fact that Ye's analysis relies heavily on the LP formulation of MDPs. In contrast, no succinct LP formulation is known for 2TBSGs. (Natural attempts fail. See Condon [7].) Our proof is based on finding natural game-theoretic quantities that correspond to the LP-based quantities used by Ye, and by reestablishing, via direct means, (improved versions) of the bounds obtained by Ye using LP duality.

Ye's [35] results and our results, combined with the recent results of Friedmann [14] and Fearnley [12], supply a *complete characterization* of the complexity of the policy/strategy iteration algorithm for MDPs/2TBSGs. The policy/strategy iteration algorithms are *strongly polynomial* for a fixed discount factor, but *exponential* for *non-discounted* problems,

or when the discount factor is part of the input. (In non-discounted problems the discounting criteria is replaced by *limiting average* criteria. In a sense, this is equivalent to letting the discount factor tend to 1. See, e.g., Derman [9].)

The rest of this paper is organized as follows. In Section 2 we define the *2-player turn-based stochastic games* (2TBSG) studied in this paper. In Sections 3, 4 and 5 we summarize known results regarding these games. For completeness, these sections contain concise, but complete, proofs of all results. (The proofs in these three sections are *not* the innovative part of this paper and may be skipped at first reading.) Finally, in Section 6 we obtain our innovative *strongly polynomial* bound on the complexity of the celebrated strategy iteration algorithm, solving a long-standing open problem. We end in Section 7 with some concluding remarks and open problems.

2 2-player turn-based stochastic games

Discounted stochastic games were first studied by Shapley [32]. In his games, the players perform *simultaneous*, or *concurrent*, actions. We consider the subclass of *turn-based* stochastic games.

We briefly review the informal definition of 2-Player Turn-Based Stochastic Games (2TBSGs), before giving a formal definition. A game is composed of states and actions. It starts at some initial state and proceeds, in discrete steps, indefinitely. In each time step one of the players plays an action. (The game is thus a *turn-based* or *perfect information* game.) Each action has a *cost* associated with it. This is the cost paid by player 1 to player 2 when this action is played. (The game is therefore a *zero-sum* game.) Each action also has a *probability distribution* on states associated with it. The next state, after playing a particular action, is chosen randomly according to this probability distribution. (The game is, in general, *stochastic*.) Finally, the game is *discounted*. The first player tries to minimize the expected total discounted cost, while the second player tries to maximize it.

Definition 2.1(**Actions**). *An* action *a over a set of states S is composed of a triplet $(s(a), p(a), c(a))$, where $s(a) \in S$ is the state from which a can be played, $p(a) \in \Delta(S)$ is a probability distribution over states according to which the next state is chosen when a is played, and $c(a) \in \mathbb{R}$ is the* cost *of a.*

Definition 2.2(**2-Player Turn-Based Stochastic Games**). *A* 2-Player Turn-Based (Discounted) Stochastic Game *(2TBSG) is a tuple $G = (S_1, S_2, A, \gamma)$, where S_1 and S_2 are the set of states controlled by players* 1 *and* 2*, respectively, and A is a set of actions. We assume that $S_1 \cap S_2 = \emptyset$ and let $S = S_1 \cup S_2$. For every $i \in S$, we let $A_i = \{a \in A \mid s(a) = i\}$ be the set of actions that can be played from i. We assume that $A_i \neq \emptyset$, for every $i \in S$. We let $A^1 = \cup_{i \in S_1} A_i$ and $A^2 = \cup_{i \in S_2} A_i$ be the sets of all actions that can be played by players* 1 *and* 2*, respectively. Finally, $0 < \gamma < 1$ is a fixed* discount factor*. If the infinite sequence of actions taken by the two players is $a_0, a_1, \ldots$, then the* total discounted cost *of this action sequence is $\sum_{k \geq 0} \gamma^k c(a_k)$.*

If one of the players has only a single action available from each state under her control, the game degenerates into a 1-player game known as a *Markov Decision Process*. (This happens, in particular, when $S_1 = \emptyset$ or $S_2 = \emptyset$.)

We next define the *probability* and *action* matrices of 2TBSGs. These matrices provide a compact representation of 2TBSGs that greatly simplifies their manipulation. Throughout the paper, we use $n = |S|$ and $m = |A|$ to denote the number of states and actions, respectively, in a game.

Definition 2.3(Probability and action matrices). *Let $G = (S_1, S_2, A, \gamma)$ be a 2TBSG. We assume, without loss of generality, that $S = S_1 \cup S_2 = [n]$ and $A = [m]$. We let $P \in \mathbb{R}^{m \times n}$, where $P_{a,i} = p(a)_i$ is the probability of ending up in state i after taking action a, for every $a \in A = [m]$ and $i \in S = [n]$, be the* probability matrix *of the game, and $\mathbf{c} \in \mathbb{R}^m$, where $\mathbf{c}_a = c(a)$ is the cost of action $a \in A = [m]$, be its* cost vector*. We also let $J \in \mathbb{R}^{m \times n}$ be a matrix such that $J_{a,i} = 1$ if and only if $a \in A_i$, and 0 otherwise. Finally, we let $Q = J - \gamma P$ be the* action matrix *of G.*

It is interesting to note that a 2TBSG is fully specified by its action matrix $Q = J - \gamma P$, its cost vector $\mathbf{c}$, and the partition of $S = [n]$ into S_1 and S_2. (Action matrices may be thought of as a stochastic and discounted generalization of the *incidence matrices* of directed graphs.)

Definition 2.4.(Strategies, strategy profiles). *A (positional)* strategy *π_j for player j, is a mapping $\pi_j : S_j \to A$ such that $\pi_j(i) \in A_i$, for every $i \in S_j$. We say that player j uses strategy π_j if whenever the game is in state i, player j chooses action $\pi_j(i)$. A* strategy profile *$\pi = (\pi_1, \pi_2)$ is simply a pair of strategies for*

the two players. We let $\Pi_j = \Pi_j(G)$*, for* $j \in \{1, 2\}$*, be the set of all strategies of player* j*, and let* $\Pi = \Pi(G) = \Pi_1 \times \Pi_2$ *be the set of all strategy profiles in* G*.*

We note that a strategy profile $\pi = (\pi_1, \pi_2)$ may be viewed as a mapping $\pi : S \to A$, i.e., as a strategy in a 1-player version of the game. All strategies considered in this paper are positional. When convenient, we also view a strategy π_j or a strategy profile π as subsets $\pi_j(S), \pi(S) \subseteq A$. A strategy profile $\pi = (\pi_1, \pi_2)$, when viewed as a subset of A, is simply the *union* $\pi_1 \cup \pi_2$. We let $P_\pi \in \mathbb{R}^{n \times n}$ be the matrix obtained by selecting the *rows* of P whose indices belong to π. Note that P_π is a (row) *stochastic matrix*. Its elements are non-negative and the elements in each row sum to 1. Similarly, $\mathbf{c}_\pi \in \mathbb{R}^n$ is the vector containing the costs of the actions that belong to π. We conveniently have $J_\pi = I$ and $Q_\pi = I - \gamma P_\pi$, for every strategy profile π.

Definition 2.5(Value vectors). *For every strategy profile* $\pi = (\pi_1, \pi_2) \in \Pi$*, we let* $\mathbf{v}_\pi = \mathbf{v}_{\pi_1,\pi_2} \in \mathbb{R}^n$ *be a vector such that* $(\mathbf{v}_\pi)_i$*, for every* $i \in S$*, is the* expected *total discounted cost when the game starts at state* i*, player* 1 *uses strategy* π_1*, and player* 2 *uses strategy* π_2*.*

Given two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, we say that $\mathbf{u} \leq \mathbf{v}$ if and only if $\mathbf{u}_i \leq \mathbf{v}_i$, for every $1 \leq i \leq n$. We say that $\mathbf{u} < \mathbf{v}$ if and only if $\mathbf{u} \leq \mathbf{v}$ and $\mathbf{u} \neq \mathbf{v}$.

Definition 2.6(Optimal counter strategies) *Let* G *be a 2TBSG and let* $\pi_2 \in \Pi_2(G)$ *be a strategy of player* 2*. A strategy* π_1 *for player* 1 *is said to be an* optimal counter-strategy *against* π_2*, if and only if* $\mathbf{v}_{\pi_1,\pi_2} \leq \mathbf{v}_{\pi'_1,\pi_2}$*, for every* $\pi'_1 \in \Pi_1(G)$*. Similarly, a strategy* π_2 *for player* 2 *is said to be an* optimal counter-strategy *against* π_1*, if and only if* $\mathbf{v}_{\pi_1,\pi_2} \geq \mathbf{v}_{\pi_1,\pi'_2}$*, for every* $\pi'_2 \in \Pi_2(G)$*. For every* $\pi_1 \in \Pi_1(G)$*, we let* $\tau_2(\pi_1)$ *be an optimal counter strategy against* π_1*, if one exists. For every* $\pi_2 \in \Pi_2(G)$*, we let* $\tau_1(\pi_2)$ *be an optimal counter strategy against* π_2*, if one exists.*

It is not immediately clear that optimal counter strategies always exist. (Note, that $\mathbf{v}_{\pi_1,\pi_2} \leq \mathbf{v}_{\pi'_1,\pi_2}$ and $\mathbf{v}_{\pi_1,\pi_2} \geq \mathbf{v}_{\pi_1,\pi'_2}$ are *vector* inequalities. As defined, optimal counter strategies need to be optimal for every initial state.) Furthermore, optimal counter strategies, if they exist, need not be unique. It is well known, however, that optimal counter strategies do always exist, as we shall also show below.

In a two-player zero-sum game, an optimal strategy is by definition one that secures the best possible guarantee on the expected payoff against any opponent. As with finite games, pairs of optimal strategies in a zero-sum stochastic game coincide with the Nash equilibria of the game. This was established by Shapley [32]. For brevity, we take this characterization to be the definition of an optimal strategy.

Definition 2.7(Optimal strategies). *A strategy profile* $\pi = (\pi_1, \pi_2) \in \Pi(G)$ *is said to be* optimal *if and only if* π_1 *is an optimal counter strategy against* π_2*, and* π_2 *is an optimal counter strategy against* π_1*. In such a case we also say that* π_1 *is an optimal strategy for player* 1 *and that* π_2 *is an optimal strategy for player* 2*.*

Shapley [32] also established the following theorem.

Theorem 2.8. *Every 2TBSG has an optimal strategy profile. If* π *and* π' *are two optimal strategy profiles then* $\mathbf{v}_\pi = \mathbf{v}_{\pi'}$*.*

Theorem 2.8 immediately implies the existence of optimal counter strategies against any strategy. It is easy to see that π_1 is an optimal strategy for player 1 if and only if $\mathbf{v}_{\pi_1,\tau_2(\pi_1)} \leq \mathbf{v}_{\pi'_1,\tau_2(\pi'_1)}$, for every $\pi'_1 \in \Pi_1$. An analogous condition clearly holds for player 2. The main result of this paper is a proof that a pair of optimal strategies can be computed in *strongly* polynomial time, when the discount factor is constant.

3 Basic results

For any strategy profile π, the matrix $(I - \gamma P_\pi)$ plays a prominent role in the sequel. (Recall that P_π is the matrix obtained by selecting the rows of P that correspond to actions that belong to π.) We thus start with the following lemma whose trivial proof is omitted.

Lemma 3.1. *For any strategy profile* π*, the matrix* $(I - \gamma P_\pi)$ *is invertible and*

$$(I - \gamma P_\pi)^{-1} = \sum_{k \geq 0} (\gamma P_\pi)^k.$$

All entries of $(I - \gamma P_\pi)^{-1}$ *are non-negative and the entries on the diagonal are strictly positive.*

Lemma 3.2. *For every strategy profile* $\pi \in \Pi$ *and every* $0 < \gamma < 1$*, we have*

$$\mathbf{v}_\pi = (I - \gamma P_\pi)^{-1} \mathbf{c}_\pi.$$

Proof. When the players use the strategy profile π,

the process becomes a Markov chain with rewards with transition matrix P_π. In particular, for every $i, j \in [n]$ and every $k \geq 0$, $(P_\pi^k)_{i,j}$ is the probability that a game that starts at state i is in state j after exactly k steps. The expected total discounted costs, starting from all states are thus

$$\mathbf{v}_\pi \;=\; \left(\sum_{k\geq 0}(\gamma^k P_\pi^k)\right)\mathbf{c}_\pi \;=\; (I-\gamma P_\pi)^{-1}\mathbf{c}_\pi. \qquad \square$$

Definition 3.3(Modified costs). *The* modified cost *vector $\mathbf{c}^\pi \in \mathbb{R}^m$ corresponding to a strategy profile π is defined to be*

$$\mathbf{c}^\pi \;=\; \mathbf{c} - (J-\gamma P)\mathbf{v}_\pi.$$

The *modified cost vector* $\mathbf{c}^\pi$ is obtained from $\mathbf{c}$ via a *potential transformation* that uses $\mathbf{v}_\pi$ as a vector of potentials. (If $h : V \to \mathbb{R}$ is a function assigning potentials to the states, then the modified cost $c_h(a)$ is defined as $c_h(a) = c(a) - h(a) + \gamma \sum_{j\in S} p(a)_j h(j)$.)

It is important to stress the difference between $\mathbf{c}_\pi \in \mathbb{R}^n$, the vector obtained by selecting the entries of $\mathbf{c}$ corresponding to strategy profile π, and the modified cost vector $\mathbf{c}^\pi = \mathbf{c} - (J-\gamma P)\mathbf{v}_\pi \in \mathbb{R}^m$ of Definition 3.3. (This distinction may be confusing at first, but it is extremely useful.)

We let $\mathbf{0}$ be an all zero vector. (The dimension of $\mathbf{0}$ will depend on the context.) Using Lemma 3.2 we immediately get the following basic but important relation.

Lemma 3.4. *For every strategy profile π we have $(\mathbf{c}^\pi)_\pi = \mathbf{0}$.*

Definition 3.5 (Modified value vectors). *For every two strategy profiles π, π', we let $\mathbf{v}^\pi_{\pi'}$ be the value vector of π' corresponding to the modified cost vector $\mathbf{c}^\pi$.*

Lemma 3.6. *For every two strategy profiles π', π we have*

$$\mathbf{v}^\pi_{\pi'} = \mathbf{v}_{\pi'} - \mathbf{v}_\pi.$$

Proof. By Definition 3.3 and Lemma 3.2 we have

$$\begin{aligned}\mathbf{v}^\pi_{\pi'} &= (I-\gamma P_{\pi'})^{-1}(\mathbf{c}^\pi)_{\pi'} \\ &= (I-\gamma P_{\pi'})^{-1}(\mathbf{c}_{\pi'} - (I-\gamma P_{\pi'})\mathbf{v}_\pi) \\ &= \mathbf{v}_{\pi'} - \mathbf{v}_\pi.\end{aligned}$$

□

Recall that $A^1 = \cup_{i\in S_1} A_i$ and $A^2 = \cup_{i\in S_2} A_i$.

Lemma 3.7 (Optimality condition). *A strategy profile π is optimal iff $(\mathbf{c}^\pi)_{A^1} \geq \mathbf{0}$ and $(\mathbf{c}^\pi)_{A^2} \leq \mathbf{0}$.*

Proof. Suppose that $(\mathbf{c}^\pi)_{A^1} \geq \mathbf{0}$ and $(\mathbf{c}^\pi)_{A^2} \leq \mathbf{0}$. Let $\pi = (\pi_1, \pi_2)$. We prove that π_1 is an optimal counter strategy against π_2. By Lemma 3.4 we have $(\mathbf{c}^\pi)_{\pi_1} = \mathbf{0}$, $(\mathbf{c}^\pi)_{\pi_2} = \mathbf{0}$ and hence $\mathbf{v}^\pi_{\pi_1,\pi_2} = \mathbf{0}$. For every $\pi'_1 \in \Pi_1$, we have $(\mathbf{c}^\pi)_{\pi'_1} \geq \mathbf{0}$, as $\pi'_1 \subseteq A^1$, and hence $(\mathbf{c}^\pi)_{\pi'_1,\pi_2} \geq \mathbf{0}$. Thus clearly $\mathbf{v}^\pi_{\pi'_1,\pi_2} \geq \mathbf{0} = \mathbf{v}^\pi_{\pi_1,\pi_2}$, and π_1 is indeed an optimal counter strategy against π_2. The proof that π_2 is an optimal counter strategy against π_1 is analogous.

Suppose now that there is an action $a \in A_{i_0}$, where $i_0 \in S_1$, such that $(\mathbf{c}^\pi)_a < 0$. (The case in which $i_0 \in S_2$ and $(\mathbf{c}^\pi)_a > 0$ is analogous.) Again, let $\pi = (\pi_1, \pi_2)$. Let $\pi'_1 \in \Pi_1$ be a policy such that $\pi'_1(i) = \pi_1(i)$, if $i \neq i_0$, and $\pi'_1(i_0) = a$. We then have $(\mathbf{c}^\pi)_{\pi'_1} < \mathbf{0}$ and $(\mathbf{c}^\pi)_{\pi_2} = \mathbf{0}$. Thus $\mathbf{v}^\pi_{\pi'_1,\pi_2} < \mathbf{0}$. (The strict inequality follows from Lemma 3.1. All entries of $(I-\gamma P_{\pi'_1,\pi_2})^{-1}$ are non-negative, and the entries on the diagonal are strictly positive.) Thus π_1 is *not* an optimal counter strategy against π_2. □

In the second part of the proof above, π'_1 is obtained from π_1 by a *profitable switch*. Profitable switches are closely related to the pivoting steps performed by the simplex algorithm. They also lie at the core of the strategy iteration algorithm whose analysis is the main focus of this paper.

Definition 3.8(Flux vectors). *For every strategy profile π, let $\mathbf{x}_\pi \in \mathbb{R}^{1\times n}$ be a row vector such that $(\mathbf{x}_\pi)_i$, for every $i \in S$, is the sum of the discounted costs, over all states, when the cost of action $\pi(i)$ is 1, while the cost of all other actions is 0, and when the players use strategy profile π.*

We let $\mathbf{e} = (1, 1, \ldots, 1)^T \in \mathbb{R}^n$ be an all one vector. Using Lemma 3.2, we easily get

Lemma 3.9. *For every strategy profile π, we have*

$$\mathbf{x}_\pi \;=\; \mathbf{e}^T(I-\gamma P_\pi)^{-1}.$$

It is in fact possible to view Lemma 3.9 as the definition of $\mathbf{x}_\pi$. The meaning of the flux vectors given in Definition 3.8 is not used in the sequel. (The flux vectors are intimately related to the *dual* linear program formulation of MDPs.)

Lemma 3.10. *For every strategy profile π, we have*

$$\mathbf{x}_\pi \mathbf{e} = \frac{n}{1-\gamma}.$$

Proof. By Lemma 3.9, Lemma 3.1, and the fact that $\mathbf{e}^T(P_\pi)^k\mathbf{e} = n$, for every $k \geq 0$, we have:

$$\mathbf{x}_\pi\mathbf{e} = \mathbf{e}^T(I-\gamma P_\pi)^{-1}\mathbf{e} = \sum_{k\geq 0}\mathbf{e}^T(\gamma P_\pi)^k\mathbf{e}$$
$$= n\sum_{k\geq 0}\gamma^k = \frac{n}{1-\gamma}.$$

Lemma 3.11. *For every strategy profile π, we have*

$$\mathbf{e}^T\mathbf{v}_\pi = \mathbf{x}_\pi\mathbf{c}_\pi.$$

Proof. By Lemma 3.2 and then Lemma 3.9, we get $\mathbf{e}^T\mathbf{v}_\pi = \mathbf{e}^T(I-\gamma P_\pi)^{-1}\mathbf{c}_\pi = \mathbf{x}_\pi\mathbf{c}_\pi$. □

Lemma 3.12. *For every strategy profile π, we have*

$$\mathbf{e}^T(\mathbf{v}_{\pi'} - \mathbf{v}_\pi) = \mathbf{x}_{\pi'}(\mathbf{c}^\pi)_{\pi'}.$$

Proof. By Lemma 3.6 and then Lemma 3.11, we have $\mathbf{e}^T(\mathbf{v}_{\pi'} - \mathbf{v}_\pi) = \mathbf{e}^T\mathbf{v}^\pi_{\pi'} = \mathbf{x}_{\pi'}(\mathbf{c}^\pi)_{\pi'}$. □

4 Value iteration

If $\mathbf{x} \in \mathbb{R}^m$ and $B \subseteq [m]$, we let $\min_B \mathbf{x} = \min_{j\in B}\mathbf{x}_j$, and similarly $\max_B \mathbf{x} = \max_{j\in B}\mathbf{x}_j$. We also let $\operatorname{argmin}_B\mathbf{x} = \operatorname{argmin}_{j\in B}\mathbf{x}_j$ and $\operatorname{argmax}_B\mathbf{x} = \operatorname{argmax}_{j\in B}\mathbf{x}_j$.

Definition 4.1(Value iteration operator). *The* value iteration *operator $\mathcal{T} : \mathbb{R}^n \to \mathbb{R}^n$ is defined as follows:*

$$(\mathcal{T}\mathbf{v})_i = \begin{cases} \min_{A_i}\mathbf{c} + \gamma P\mathbf{v}\,, & \text{if } i \in S_1, \\ \max_{A_i}\mathbf{c} + \gamma P\mathbf{v}\,, & \text{if } i \in S_2. \end{cases}$$

The operator $\mathcal{T}$ is a contraction with Lipschitz constant γ.

Lemma 4.2. *For every $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$ we have $\|\mathcal{T}\mathbf{u} - \mathcal{T}\mathbf{v}\|_\infty \leq \gamma\,\|\mathbf{u}-\mathbf{v}\|_\infty$.*

Proof. Assume that $i \in S_1$ and that $(\mathcal{T}\mathbf{u})_i \geq (\mathcal{T}\mathbf{v})_i$. (The other cases are analogous.) Let $a = \operatorname{argmin}_{A_i} c + \gamma P\mathbf{u}$ and $b = \operatorname{argmin}_{A_i} c + \gamma P\mathbf{v}$. Then,

$$\begin{aligned}(\mathcal{T}\mathbf{u} - \mathcal{T}\mathbf{v})_i &= (\mathbf{c}_a + \gamma P_a\mathbf{u}) - (\mathbf{c}_b + \gamma P_b\mathbf{v}) \\ &\leq (\mathbf{c}_b + \gamma P_b\mathbf{u}) - (\mathbf{c}_b + \gamma P_b\mathbf{v}) \\ &= \gamma P_b(\mathbf{u}-\mathbf{v}) \\ &\leq \gamma\,\|\mathbf{u}-\mathbf{v}\|_\infty.\end{aligned}$$

The last inequality follows from the fact that the elements in P_b are non-negative and sum-up to 1. □

Banach fixed point theorem now implies that the value iteration operator $\mathcal{T}$ has a unique fixed point.

Corollary 4.3. *There is a unique vector $\mathbf{v}^* \in \mathbb{R}^n$ such that $\mathcal{T}\mathbf{v}^* = \mathbf{v}^*$.*

We next define the *strategy extraction* operators that play an important role in this section, and the central role in the next section.

Definition 4.4(Strategy extraction operators). *The* strategy extraction *operators $\mathcal{P}_1 : \mathbb{R}^n \to \Pi_1$ and $\mathcal{P}_2 : \mathbb{R}^n \to \Pi_2$ and $\mathcal{P} : \mathbb{R}^n \to \Pi$ are defined as follows:*

$$(\mathcal{P}_1\mathbf{v})(i) = \operatorname{argmin}_{A_i}\mathbf{c} + \gamma P\mathbf{v} \quad , \quad i \in S_1,$$
$$(\mathcal{P}_2\mathbf{v})(i) = \operatorname{argmax}_{A_i}\mathbf{c} + \gamma P\mathbf{v} \quad , \quad i \in S_2.$$

and

$$\mathcal{P}\mathbf{v} = (\mathcal{P}_1\mathbf{v}, \mathcal{P}_2\mathbf{v}).$$

The following relation between the value iteration and strategy extraction operator is immediate.

Lemma 4.5. *For every $\mathbf{v} \in \mathbb{R}^n$ we have $\mathcal{T}\mathbf{v} = (\mathbf{c} + \gamma P\mathbf{v})_\pi$, where $\pi = \mathcal{P}\mathbf{v}$.*

The following simple lemma provides an interesting relation between the strategy extraction operator and modified cost vectors.

Lemma 4.6. *For every strategy profile π we have*

$$(\mathcal{P}_1\mathbf{v}_\pi)(i) = \operatorname{argmin}_{A_i}\mathbf{c}^\pi \quad , \quad i \in S_1,$$
$$(\mathcal{P}_2\mathbf{v}_\pi)(i) = \operatorname{argmax}_{A_i}\mathbf{c}^\pi \quad , \quad i \in S_2.$$

Proof. Let $\mathbf{v} = \mathbf{v}_\pi$. If $a \in A_i$ then,

$$(\mathbf{c}^\pi)_a = \mathbf{c}_a - (\mathbf{v}_i - \gamma P_a\mathbf{v}) = (\mathbf{c} + \gamma P\mathbf{v})_a - \mathbf{v}_i.$$

Thus, if $a, a' \in A_i$, then $(\mathbf{c} + \gamma P\mathbf{v})_a \leq (\mathbf{c} + \gamma P\mathbf{v})_{a'}$ if and only if $(\mathbf{c}^\pi)_a \leq (\mathbf{c}^\pi)_{a'}$. □

The following lemma supplies a simple proof of Theorem 2.8. (This is, in fact, the original proof given by Shapley [32].)

Lemma 4.7. *Let $\mathbf{v}^* \in \mathbb{R}^n$ be the unique fixed point of $\mathcal{T}$ and let $\pi = \mathcal{P}\mathbf{v}^*$. Then, π is an optimal strategy profile.*

Proof. By Lemma 4.5, we get that $\mathbf{v}^* = \mathcal{T}\mathbf{v}^* = \mathbf{c}_\pi + \gamma P_\pi\mathbf{v}^*$. By Lemma 3.2 we get $\mathbf{v}_\pi = \mathbf{v}^*$. We next show that π satisfies the optimality condition of Lemma 3.7, and hence is an optimal strategy profile. Suppose that $i \in S_1$ and that $a \in A_i$. By Lemma 4.6, we have $\pi(i) = (\mathcal{P}_1\mathbf{v}^*)(i) = \operatorname{argmin}_{A_i}\mathbf{c}^\pi$. As $(\mathbf{c}^\pi)_{\pi(i)} = 0$, we get that $(c^\pi)_a \geq 0$. Similarly, if $i \in S_2$ and $a \in A_i$, we get that $(c^\pi)_a \leq 0$. □

The *value iteration* algorithm, given at the top of Figure 1, repeatedly applies the value iteration operator $\mathcal{T}$ to an initial vector $\mathbf{u}^0 \in \mathbb{R}^n$, generating a sequence of vectors $(\mathbf{u}^k)_{k=0}^N$, where $\mathbf{u}^{k+1} = \mathcal{T}\mathbf{u}^k$, until the difference between two successive vectors is small enough, i.e., $\|\mathbf{u}^{k-1} - \mathbf{u}^k\|_\infty < \epsilon$.

Function VALUE-ITERATION($\mathbf{u}^0, \epsilon$)

$k \leftarrow 0$;
repeat
  $\mathbf{u}^{k+1} \leftarrow \mathcal{T}\mathbf{u}^k$;
  $k \leftarrow k + 1$
until $\|\mathbf{u}^{k-1} - \mathbf{u}^k\|_\infty < \epsilon$;
return $\mathbf{u}^k$

Function STRATEGY-ITERATION(σ^0)

$k \leftarrow 0$;
repeat
  $\tau^k = \tau_2(\sigma^k)$;
  $\mathbf{v}^k \leftarrow \mathbf{v}_{\sigma^k,\tau^k}$;
  $\sigma^{k+1} \leftarrow \mathcal{P}_1\mathbf{v}^k$ (if possible $\sigma^{k+1} \leftarrow \sigma^k$) ;
  $k \leftarrow k + 1$
until $\sigma^{k-1} = \sigma^k$;
return σ^k

Figure 1: The VALUE-ITERATION and STRATEGY-ITERATION algorithms.

Lemma 4.8. *Let $(\mathbf{u}^k)_{k=0}^N$ be the sequence of value vectors generated by a call* VALUE-ITERATION($\mathbf{u}^0, \epsilon$)*, for some $\epsilon > 0$. Let $\mathbf{v}^*$ be the optimal value vector. Then, for every $0 \le k \le N$ we have*

$$\|\mathbf{u}^k - \mathbf{v}^*\|_\infty \le \gamma^k \|\mathbf{u}^0 - \mathbf{v}^*\|_\infty.$$

Proof. By Lemma 4.2 and the fact that $\mathcal{T}\mathbf{v}^* = \mathbf{v}^*$, we have

$$\|\mathbf{u}^k - \mathbf{v}^*\|_\infty = \|\mathcal{T}\mathbf{u}^{k-1} - \mathcal{T}\mathbf{v}^*\|_\infty \le \gamma \|\mathbf{u}^{k-1} - \mathbf{v}^*\|_\infty.$$

The claim follows easily by induction. □

It follows immediately from Lemma 4.8, that for any $\mathbf{u} \in \mathbb{R}^n$, the infinite sequence of vectors generated by the call VALUE-ITERATION($\mathbf{u}^0, 0$) converges to the optimal value vector $\mathbf{v}^*$. Also, for every $\epsilon > 0$, the call VALUE-ITERATION($\mathbf{u}^0, \epsilon$) eventually terminates.

5 Strategy iteration

The *strategy iteration* algorithm is given at the bottom of Figure 1. It was first described for the MDP case by Howard [18] and is called *policy iteration* or *Howard's algorithm* in that context. It was described for 2-player stochastic games by Rao *et al.* [30]. (Their algorithm actually works on more general imperfect information games for which it is a non-terminating approximation algorithm.)

The strategy iteration algorithm receives an initial strategy σ^0 of player 1, and generates a sequence $\pi^k = (\sigma^k, \tau^k)$ of strategy profiles of the two players, ending with an optimal strategy profile. Each iteration of the algorithm receives a strategy σ^k and produces an *improved* strategy σ^{k+1} as follows. The algorithm first computes an optimal counter-strategy $\tau^k = \tau_2(\sigma^k)$ for player 2 against σ^k. (We assume here that this can be done in strongly polynomial time. One way of doing it is to apply the strategy iteration algorithm on a restricted game in which σ^k is the only strategy available to player 1). Next, it *evaluates* the strategy profile $\pi^k = (\sigma^k, \tau^k)$, by solving a system of linear equations, and obtains its value vector $\mathbf{v}^k = \mathbf{v}_{\pi^k}$. It then lets $\sigma^{k+1} = \mathcal{P}_1\mathbf{v}_{\pi^k}$. Ties are broken, if possible, in favor of actions that are in σ^k. (This is important, as termination is not guaranteed without this provision.) The algorithm terminates when two consecutive strategies σ^k and σ^{k+1} are identical.

The step $\sigma^{k+1} = \mathcal{P}_1\mathbf{v}_{\pi^k}$ is the main step of the strategy iteration algorithm. As we shall (implicitly) see below, σ^{k+1} is obtained from σ^k by performing a collection of improving switches.

To prove the correctness of the STRATEGY-ITERATION algorithm we use the following lemma. (Note that π^1 in the lemma is obtained from π^0 using one iteration of the STRATEGY-ITERATION algorithm.)

Lemma 5.1. *Let $\sigma^0 \in \Pi_1$, $\pi^0 = (\sigma^0, \tau_2(\sigma^0))$ and $\sigma^1 = \mathcal{P}_1\mathbf{v}_{\pi^0}$, $\pi^1 = (\sigma^1, \tau_2(\sigma^1))$. Then $\mathbf{v}_{\pi^0} \ge \mathbf{v}_{\pi^1}$.*

Proof. We show that $\mathbf{v}^{\pi^0}_{\pi^0} = \mathbf{0} \ge \mathbf{v}^{\pi^0}_{\pi^1}$, which by Lemma 3.6 implies that $\mathbf{v}_{\pi^0} \ge \mathbf{v}_{\pi^1}$. To show that $\mathbf{v}^{\pi^0}_{\pi^1} \le \mathbf{0}$, we show that $(\mathbf{c}^{\pi^0})_{\pi^1} \le \mathbf{0}$. The fact that $(\mathbf{c}^{\pi^0})_{\sigma^1} \le \mathbf{0}$ follows from the fact that for every $i \in S_1$ we have $\sigma^1(i) = \operatorname{argmin}_{A_i} \mathbf{c}^{\pi^0}$ and hence $(\mathbf{c}^{\pi^0})_{\sigma^1(i)} \le (\mathbf{c}^{\pi^0})_{\sigma^0(i)} = 0$. The fact that $(\mathbf{c}^{\pi^0})_{\tau^1} \le \mathbf{0}$ follows from fact that τ^0 is an optimal counter strategy against σ^0, so in fact $(\mathbf{c}^{\pi^0})_{A^2} \le \mathbf{0}$. □

Lemma 5.2. *For every initial strategy σ^0,*

STRATEGY-ITERATION(σ^0) *terminates after a finite number of iterations. If* $(\mathbf{v}^k)_{k=0}^N$ *is the sequence of value vectors generated by the call, then,* $\mathbf{v}^{k-1} > \mathbf{v}^k \geq \mathbf{v}^*$*, for every* $1 \leq k < N$*. Furthermore,* $\mathbf{v}^{N-1} = \mathbf{v}^N = \mathbf{v}^*$ *and* $\pi^{N-1} = \pi^N$ *is an optimal strategy profile.*

Proof. The claim $\mathbf{v}^{k-1} \geq \mathbf{v}^k$, for every $1 \leq k \leq N$ follows easily from Lemma 5.1 by induction. Next, we note that if $\mathbf{v}^{k-1} = \mathbf{v}^k$, for some k, then by the reasoning used in the proof of Lemma 5.1, we must have $(\mathbf{c}^{\pi^{k-1}})_{A^1} \geq \mathbf{0}$ and $(\mathbf{c}^{\pi^{k-1}})_{A^2} \leq \mathbf{0}$. By the optimality condition, we get that π^{k-1} is an optimal strategy profile. By the tie breaking mechanism used, we also get that $\pi^k = \pi^{k-1}$. Finally, the fact that $\mathbf{v}^{k-1} > \mathbf{v}^k$, for every $1 \leq k < N$, implies that strategy profiles encountered cannot repeat themselves. As there is only a finite number of such profiles, the sequence of strategy profiles generated must be finite. □

We next relate the sequences of value vectors obtained by running STRATEGY-ITERATION(σ^0) and VALUE-ITERATION($\mathbf{v}_{\pi^0}, \epsilon$), where $\pi^0 = (\sigma^0, \tau_2(\sigma^0))$. The following lemmas for the case of MDPs are well-known and appear, e.g., in Meister and Holzbaur [27]. The proofs for the 2-player case are essentially identical. (They may be folklore.)

Lemma 5.3. *Let* $\sigma^0 \in \Pi_1$*,* $\pi^0 = (\sigma^0, \tau_2(\sigma^0))$*, and* $\sigma^1 = \mathcal{P}_1 \mathbf{v}_{\pi^0}$*,* $\pi^1 = (\sigma^1, \tau_2(\sigma^1))$*. Then* $\mathcal{T}\mathbf{v}_{\pi^0} \geq \mathbf{v}_{\pi^1}$*.*

Proof. Let $i \in S_1$. As $\sigma^1(i) = \operatorname{argmin}_{A_i} \mathbf{c} + \gamma P \mathbf{v}_{\pi^0}$, $\mathbf{v}_{\pi^0} \geq \mathbf{v}_{\pi^1}$, and $\mathbf{c}_{\pi^1} + \gamma P_{\pi^1} \mathbf{v}_{\pi^1} = \mathbf{v}_{\pi^1}$, we have

$$\begin{aligned}(\mathcal{T}\mathbf{v}_{\pi^0})_i &= \min_{A_i} \mathbf{c} + \gamma P \mathbf{v}_{\pi^0} = (\mathbf{c} + \gamma P \mathbf{v}_{\pi^0})_{\sigma^1(i)} \\ &\geq (\mathbf{c} + \gamma P \mathbf{v}_{\pi^1})_{\sigma^1(i)} = (\mathbf{v}_{\pi^1})_i.\end{aligned}$$

Similarly, if $i \in S_2$, then

$$\begin{aligned}(\mathcal{T}\mathbf{v}_{\pi^0})_i &= \max_{A_i} \mathbf{c} + \gamma P \mathbf{v}_{\pi^0} \geq (\mathbf{c} + \gamma P \mathbf{v}_{\pi^0})_{\tau^1(i)} \\ &\geq (\mathbf{c} + \gamma P \mathbf{v}_{\pi^1})_{\tau^1(i)} = (\mathbf{v}_{\pi^1})_i.\end{aligned}$$ □

Using Lemma 5.3, we immediately get:

Lemma 5.4. *Let* $(\mathbf{v}^k)_{k=0}^N$ *be the value vectors generated by* STRATEGY-ITERATION(σ^0)*, and let* $(\mathbf{u}^k)_{k=0}^\infty$ *be the value vectors generated by* VALUE-ITERATION($\mathbf{v}_{\pi^0}, 0$)*, where* $\pi^0 = (\sigma^0, \tau_2(\sigma^0))$*. Then,* $\mathbf{v}^k \leq \mathbf{u}^k$*, for every* $0 \leq k \leq N$*.*

Proof. We prove the lemma by induction. We have $\mathbf{v}^0 = \mathbf{u}^0$. Suppose now that $\mathbf{v}^k \leq \mathbf{u}^k$. Then, by Lemma 5.3 and the monotonicity of the value iteration operator, we have:

$$\mathbf{v}^{k+1} \leq \mathcal{T}\mathbf{v}^k \leq \mathcal{T}\mathbf{u}^k = \mathbf{u}^{k+1}.$$ □

Combining Lemmas 4.8 and 5.4, we get

Lemma 5.5. *Let* $(\mathbf{v}^k)_{k=0}^N$ *be the sequence of value vectors generated by* STRATEGY-ITERATION(σ^0)*, for some* $\sigma^0 \in \Pi_1$*. Let* $\mathbf{v}^*$ *be the optimal value vector. Then, for every* $0 \leq k \leq N$ *we have*

$$\|\mathbf{v}^k - \mathbf{v}^*\|_\infty \leq \gamma^k \|\mathbf{v}^0 - \mathbf{v}^*\|_\infty.$$

6 Strongly polynomial bound

In this section, the main section of the paper, we present our strongly polynomial bound on the number of iterations performed by the strategy iteration algorithm. We begin with some technical lemmas.

Lemma 6.1. *Let* π', π *be two strategy profiles such that* $\mathbf{v}_{\pi'} \geq \mathbf{v}_\pi$ *and let* $a = \pi'(i)$ *where* $i \in S$*. Then,*

$$(\mathbf{v}_{\pi'} - \mathbf{v}_\pi)_i \geq (\mathbf{c}^\pi)_a.$$

Proof. By Lemma 3.2 and Definition 3.3 we have:

$$\begin{aligned}(\mathbf{v}_{\pi'})_i - (\mathbf{v}_\pi)_i &= (\mathbf{c} + \gamma P \mathbf{v}_{\pi'})_a - (\mathbf{v}_\pi)_i \geq \\ (\mathbf{c} + \gamma P \mathbf{v}_\pi)_a - (\mathbf{v}_\pi)_i &= (\mathbf{c}^\pi)_a.\end{aligned}$$ □

Lemma 6.2. *Let* π'', π *be two strategy profiles such that* $\mathbf{v}_{\pi''} \geq \mathbf{v}_\pi$ *and let* $a = \operatorname{argmax}_{\pi''} \mathbf{c}^\pi$*. Then,*

$$\|\mathbf{v}_{\pi''} - \mathbf{v}_\pi\|_1 \leq \frac{n}{1-\gamma} (\mathbf{c}^\pi)_a.$$

Proof. As $\mathbf{v}_{\pi''} \geq \mathbf{v}_\pi$, we get using Lemma 3.12 and then Lemma 3.10 that

$$\begin{aligned}||\mathbf{v}_{\pi''} - \mathbf{v}_\pi||_1 &= \mathbf{e}^T(\mathbf{v}_{\pi''} - \mathbf{v}_\pi) = \mathbf{x}_{\pi''}(\mathbf{c}^\pi)_{\pi''} \\ &\leq \mathbf{x}_{\pi''} \mathbf{e} \, (\mathbf{c}^\pi)_a = \frac{n}{1-\gamma} (\mathbf{c}^\pi)_a.\end{aligned}$$ □

Lemma 6.3. *Let* π'', π', π *be three strategy profiles such that* $\mathbf{v}_{\pi''} \geq \mathbf{v}_{\pi'} \geq \mathbf{v}_\pi$*. Let* $a = argmax_{\pi''} \mathbf{c}^\pi$ *and suppose that* $a \in \pi'$*. Then,*

$$||\mathbf{v}_{\pi'} - \mathbf{v}_\pi||_1 \geq \frac{1-\gamma}{n} \|\mathbf{v}_{\pi''} - \mathbf{v}_\pi\|_1.$$

Proof. Let $i \in S$ be the state for which $\pi''(i) = \pi'(i) = a$. By Lemma 6.1 and Lemma 6.2 we get

$$\begin{aligned}\|\mathbf{v}_{\pi'} - \mathbf{v}_\pi\|_1 &\geq (\mathbf{v}_{\pi'} - \mathbf{v}_\pi)_i \geq (\mathbf{c}^\pi)_a \\ &\geq \frac{1-\gamma}{n} \|\mathbf{v}_{\pi''} - \mathbf{v}_\pi\|_1.\end{aligned}$$ □

Lemma 6.4. *Let $(\sigma^k)_{k=0}^N$ be the sequence of player 1 strategies generated by the* Strategy-Iteration *algorithm, starting from some initial strategy σ^0. Let $L = \log_{1/\gamma} \frac{n^2}{1-\gamma}$. Then, every strategy σ^k contains an action that does not appear in any strategy σ^ℓ, where $k + L < \ell \leq N$.*

Proof. Let $(\pi^k)_{k=0}^N$, where $\pi^k = (\sigma^k, \tau^k)$, be the sequence of strategy profiles generated by the strategy iteration algorithm. By the correctness of the strategy iteration algorithm, $\pi^* = \pi^N$ is composed of optimal strategies for the two players. Let $a = argmax_{\pi^k} \mathbf{c}^{\pi^*}$. By Lemma 3.7, we have $(\mathbf{c}^{\pi^*})_a \geq 0$ for every $a \in A^1$, and $(\mathbf{c}^{\pi^*})_a \leq 0$ for every $a \in A^2$. We may assume, therefore, that $a \in A^1$, i.e., that a is an action controlled by player 1. Suppose, for the sake of contradiction, that $a \in \pi^\ell$, for some $k + L < \ell \leq N$. Using Lemma 6.3, with $\pi'' = \pi^k$, $\pi' = \pi^\ell$ and $\pi = \pi^*$, we get that

$$\|\mathbf{v}_{\pi^\ell} - \mathbf{v}_{\pi^*}\|_1 \;\geq\; \frac{1-\gamma}{n} \|\mathbf{v}_{\pi^k} - \mathbf{v}_{\pi^*}\|_1.$$

On the other hand, using Lemma 5.5, we get that

$$\|\mathbf{v}_{\pi^\ell} - \mathbf{v}_{\pi^*}\|_\infty \;\leq\; \gamma^{\ell-k} \|\mathbf{v}_{\pi^k} - \mathbf{v}_{\pi^*}\|_\infty.$$

Thus,

$$\begin{aligned} \|\mathbf{v}_{\pi^\ell} - \mathbf{v}_{\pi^*}\|_1 &\leq n \|\mathbf{v}_{\pi^\ell} - \mathbf{v}_{\pi^*}\|_\infty \\ &\leq n\gamma^{\ell-k} \|\mathbf{v}_{\pi^k} - \mathbf{v}_{\pi^*}\|_\infty \\ &\leq n\gamma^{\ell-k} \|\mathbf{v}_{\pi^k} - \mathbf{v}_{\pi^*}\|_1. \end{aligned}$$

It follows that $n\gamma^{\ell-k} \geq \frac{1-\gamma}{n}$ and hence

$$\gamma^L \;>\; \gamma^{\ell-k} \;\geq\; \frac{1-\gamma}{n^2},$$

a contradiction. □

Theorem 6.5. *The* Strategy-Iteration *algorithm, starting from any initial strategy, terminates with an optimal strategy after at most $(m+1)(1 + \log_{1/\gamma} \frac{n^2}{1-\gamma}) = O(\frac{m}{1-\gamma} \log \frac{n}{1-\gamma})$ iterations.*

Proof. Let $\bar{L} = \lfloor 1 + \log_{1/\gamma} \frac{n^2}{1-\gamma} \rfloor$. Consider strategies $\sigma^0, \sigma^{\bar{L}}, \sigma^{2\bar{L}}, \ldots$. By Lemma 6.4, every strategy in this subsequence contains a new action that would never be used again. As there are only m actions, the total number of strategies in the sequence is at most $(m+1)\bar{L} = (m+1)(1 + \log_{1/\gamma} \frac{n^2}{1-\gamma})$. Finally, note that $\log_{1/\gamma} x = \frac{\log x}{\log 1/\gamma} \leq \frac{x}{1-\gamma}$. □

7 Concluding remarks

We have shown that the strategy iteration algorithm is *strongly polynomial* for 2TBSGs with a *fixed* discount factor. Friedmann [14], on the other hand, has recently shown that the strategy iteration algorithm is *exponential* for non-discounted 2TBSG, or when the discount factor is part of the input.

The existence of polynomial time algorithms for 2TBSGs when the discount factor is part of the input, or for the non-discounted case, remains an intriguing and a challenging open problem, with many possible consequences for complexity theory and automatic verification. As shown by Andersson and Miltersen [1], this is equivalent to finding a polynomial time algorithm for Condon's [6] *Simple Stochastic Games* (SSGs). Such an algorithm will immediately provide polynomial time algorithms for *Mean Payoff Games* (MPGs) (see [10], [15], [36]) and *Parity Games* (PGs) (see, e.g., [11], [33], [19]).

We believe that our results give some hope of obtaining a polynomial time algorithm for this problem. In an earlier work, Ye [34] gave a polynomial time algorithm for the analogous MDP problem. His algorithm uses interior point methods and its analysis relies again on the LP formulation of the MDP problem. Given the "deLPfication" of Ye's [35] analysis of the policy iteration algorithm carried out here, one could speculate that looking at interior point methods for the two-player case, with Ye's [34] algorithm for MDPs as a starting point, would be a fertile approach.

References

[1] D. Andersson and P. Miltersen. The complexity of solving stochastic games on graphs. In *Proc. of 20th ISAAC,* pages 112-121, 2009.

[2] R. Bellman. *Dynamic programming.* Princeton University Press, 1957.

[3] D. Bertsekas. *Dynamic programming and optimal control.* Athena Scientific, 2nd edition, 2001.

[4] H. Björklund and S. Vorobyov. Combinatorial structure and randomized subexponential algorithms for innite games. *Theoretical Computer Science*, 349(3):347-360, 2005.

[5] H. Björklund and S. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. *Discrete Applied Mathematics*, 155(2):210- 229, 2007.

[6] A. Condon.The complexity of stochastic games. *Informa-tion and Computation,* 96:203-224, 1992.

[7] A. Condon. On algorithms for simple stochastic games. *DI-MACS Series in Discrete Mathematics and Theoretical Com-puter Science*, 13:51-71, 1993.

[8] F.d'Epenoux. A probabilistic production and inventory problem. *Management Science*, 10(1):98-108, 1963.

[9] C. Derman. *Finite state Markov decision processes.* Aca-demic Press, 1972.

[10] A. Ehrenfeucht and J. Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8:109-113, 1979.

[11] E. Emerson and C. Jutla. Tree automata, μ-calculus and de-terminacy. In *Proceedings of the 32nd FOCS*, pages 368- 377. IEEE Computer Society Press, 1991.

[12] J. Fearnley. Exponential lower bounds for policy iteration. In *Proc. of 37th ICALP*, 2010. Preliminary version available at http://arxiv.org/abs/1003.3418v1.

[13] J. Filar and K. Vrieze. *Competitive Markov decision pro-cesses.* Springer-Verlag New York, Inc., New York, NY, USA, 1996.

[14] O. Friedmann. An exponential lower bound for the parity game strategy improvement algorithm as we know it. In *Proc. of 24th LICS*, pages 145-156, 2009.

[15] V. Gurvich, A. Karzanov, and L. Khachiyan. Cyclic games and an algorithm to and minimax cycle means in directed graphs. *USSR Computational Mathematics and Mathematical Physics*, 28:85-91, 1988.

[16] N. Halman. Simple stochastic games, parity games, mean payoff games and discounted payoff games are all LP-type problems. *Algorithmica*, 49(1):37-50, 2007.

[17] A.Hoffman and R. Karp. On nonterminating stochastic games. *Management Science*, 12:359-370, 1966.

[18] R. Howard. *Dynamic programming and Markov processes.* MIT Press, 1960.

[19] M. Jurdzínski, M. Paterson, and U. Zwick. A deterministic subexponential algorithm for solving parity games. *SIAM Journal on Computing*, 38(4):1519-1532, 2008.

[20] G. Kalai. A subexponential randomized simplex algorithm (extended abstract). In *Proc. of 24th STOC,* pages 475-482, 1992.

[21] G. Kalai. Linear programming, the simplex algorithm and simple polytopes. *Mathematical Programming*, 79:217-233, 1997.

[22] M. Littman. *Algorithms for sequential decision making.* PhD thesis, Brown University, Department of Computer Science, 1996.

[23] M. Littman, T. Dean, and L. Kaelbling. On the complexity of solving markov decision problems. In *Proc. of the 11th UAI*, pages 394-402, 1995.

[24] W. Ludwig. A subexponential randomized algorithm for the simple stochastic game problem. *Information and Computation*, 117(1):151-155, 1995.

[25] Y. Mansour and S. Singh. On the complexity of policy iteration. In *Proc. of the 15th UAI*, pages 401-408, 1999.

[26] J. Matou$\check{}$ $\check{}$sek, M. Sharir, and E. Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4-5):498- 516, 1996.

[27] U.Meister and U. Holzbaur. A polynomial time bound for Howard's policy improvement algorithm. *OR Spektrum,* 8:37-40, 1986.

[28] A. Neyman and S. Sorin, editors. Stochastic Games and Applications, volume 570 of *NATO Science Series C: Mathematical and Physical Sciences.* Springer, 2003.

[29] M. Puterman. *Markov decision processes.* Wiley, 1994.

[30] S. Rao, R. Chandrasekaran, and K. Nair. Algorithms for discounted games. *Journal of Optimization Theory and Appli-cations*, pages 627-637, 1973.

[31] N. Schmitz. How good is Howard's policy improvement algorithm? *Mathematical Methods of Operations Research*, 29:315-316, 1985.

[32] L. Shapley. Stochastic games. *Proc. Nat. Acad. Sci. U.S.A.*, 39:1095-1100, 1953.

[33] J. Vöge and M. Jurdzínski. A discrete strategy improvement algorithm for solving parity games (Extended abstract). In *International Conference on Computer-Aided Verification, CAV 2000,* volume 1855 of *LNCS*, pages 202-215. Springer, 2000.

[34] Y.Ye. A new complexity result on solving the Markov decision problem. *Mathematics of Operations Research*, 30(3):733-749, 2005.

[35] Y. Ye. The simplex method is strongly polynomial for the Markov decision problem with a fixed discount rate. Available at http: //www. stanford. edu/ ~ yyye/ simplexmdp1. pdf, 2010.

[36] U. Zwick and M. Paterson. The complexity of mean pay-off games on graphs. *Theoretical Computer Science*, 158(1- 2):343-359, 1996.