On Strategy Improvement Algorithms for Simple Stochastic Games

Rahul Tripathi[1], Elena Valkanova[1], and V. S. Anil Kumar[2]

[1] Department of Computer Science and Engineering, University of South Florida, Tampa, FL 33620, USA. Email: `tripathi@cse.usf.edu`, `evalkanova75@yahoo.com`
[2] Virginia Bioinformatics Institute and Dept. of Computer Science, Virginia Tech. Blacksburg, VT 24061, USA. Email: `akumar@vbi.vt.edu`

Abstract. The study of *simple stochastic games* (SSGs) was initiated by Condon for analyzing the computational power of randomized space-bounded alternating Turing machines. The game is played by two players, MAX and MIN, on a directed multigraph, and when the play terminates at a sink s, MAX wins from MIN a payoff $p(s) \in [0, 1]$. Condon showed that the *SSG value problem*, which given a SSG asks whether the expected payoff won by MAX exceeds 1/2 when both players use their optimal strategies, is in NP $\cap$ coNP. However, the exact complexity of this problem remains open as it is not known whether the problem is in P or is hard for some natural complexity class. In this paper, we study the computational complexity of a strategy improvement algorithm by Hoffman and Karp for this problem. The Hoffman-Karp algorithm converges to optimal strategies of a given SSG, but no nontrivial bounds were previously known on its running time. We show a bound of $O(2^n/n)$ on the convergence time of this algorithm, and a bound of $O(2^{0.78n})$ on a randomized variant. These are the first non-trivial upper bounds on the convergence time of these strategy improvement algorithms.

1 Introduction

Stochastic games, first studied by Shapley in 1953 [Sha53], are two-player games that arise in a number of applications, including complexity theory, game theory, operations research, automated software verification, and reactive systems. Several variations of stochastic games have been studied, and an interesting restriction of this game model is the class of *simple stochastic games* (SSGs). Condon [Con92] initiated the study of SSGs for analyzing the computational power of randomized space-bounded alternating Turing machines. The game is played by two players, MAX and MIN, on a game board, which is a directed multigraph $G = (V, E)$. The vertex set V is partitioned into disjoint subsets V_{MAX}, V_{MIN}, V_{AVE}, and $\mathrm{V}_{\mathrm{SINK}}$. Each vertex $x \in \mathrm{V}_{\mathrm{SINK}}$ has a payoff $p(x) \in [0, 1]$. The play is determined by a token, which is initially placed at a given *start* vertex. At each step, the token is moved along an edge of the graph. If the token is at vertex $x \in V_{\mathrm{MAX}}$, MAX chooses an outgoing edge, while if $x \in V_{\mathrm{MIN}}$,

MIN chooses the outgoing edge. If $x \in V_{\mathrm{AVE}}$, the outgoing edge is chosen uniformly at random, and if $x \in \mathrm{V}_{\mathrm{SINK}}$, then the play stops and MAX wins from MIN the payoff $p(x)$. The goal of MAX is to maximize the payoff won from MIN, while the goal of MIN is to minimize this payoff. Condon studied the computational complexity of the problem SSG-VAL, which is the problem of determining whether the expected payoff won by MAX is greater than 1/2, when both players use their optimal strategies. She showed that SSG-VAL is complete for the class of languages accepted by *logspace randomized alternating Turing machines*, and is in NP $\cap$ coNP. Despite considerable interest, the complexity of this problem is not fully resolved as it is unknown whether the problem is in P or is hard for some natural complexity class. The best known algorithms for SSG-VAL are subexponential-time randomized algorithms of Ludwig [Lud95] and Halman [Hal07]. This puts SSG-VAL among a small list of natural combinatorial problems in NP $\cap$ coNP, which are not yet known to be in P; the sub-exponential upper bound makes this problem rarer still.

Apart from the results in [Lud95,Hal07], very few additional upper bound results are known for SSG-VAL. Condon [Con92] showed that restricted versions of SSG-VAL consisting of only two classes of nodes (out of MAX, MIN and AVE) can be solved in polynomial time. This was extended by Gimbert et al. [GH09] who developed a fixed parameter tractable algorithm in terms of $|V_{\mathrm{AVE}}|$, which runs in time $O(|V_{\mathrm{AVE}}|! \cdot \mathrm{poly}(n))$ where n is the size (in bits) of the input game. There have been numerous other algorithms for this problem based on the general approach of "strategy improvement," which involves switching the choices of nodes that are not locally optimal. One well studied strategy improvement algorithm is the Hoffman-Karp algorithm [HK66] (Section 3) in which choices of *all* locally non-optimal nodes are switched in each iteration until optimal strategies are found. Condon [Con92] showed that the Hoffman-Karp algorithm does converge to the optimum, though its convergence time is still not very well understood. The specific details of such strategy improvement algorithms are important, and as discussed by Condon [Con93], variants of Hoffman-Karp and other natural heuristics do not converge to the optimum.

The focus of this paper is on understanding the convergence time of the Hoffman-Karp algorithm. Let n denote $\min\{|V_{\mathrm{MAX}}|, |V_{\mathrm{MIN}}|\}$. We show that the convergence time of the Hoffman-Karp algorithm is $O(2^n/n)$. This is the first non-trivial upper bound on the convergence time of the Hoffman-Karp algorithm. We also consider a randomized variant of this algorithm, and show that the convergence time of the randomized algorithm is $O(2^{0.78n})$. While these bounds are still exponential, they represent an improved understanding of these strategy improvement algorithms. Our analyses extend those of Mansour and Singh [MS99] for policy iteration algorithms for Markov decision processes.

2 Preliminaries

We now discuss the basic concepts and notations needed for the rest of the paper. We follow the definitions presented in [Som05] for the most part.

A SSG is a two-player game between players MAX and MIN. The game is played on a *game board*, which is a directed multigraph $G = (V, E)$. The vertex set V is partitioned into disjoint subsets V_{MAX}, V_{MIN}, V_{AVE}, and V_{SINK}. All vertices of G, except those of V_{SINK}, have exactly two outgoing edges. The vertices belonging to V_{SINK} have no outgoing edge. The vertex set $V_{\text{POS}} =_{df} V_{\text{MAX}} \cup V_{\text{MIN}} \cup V_{\text{AVE}}$ represents the set of all game positions and the edge set E denotes a possible move in the game. One vertex from V_{POS} is called start vertex. We call $x \in V$ a MAX-*position* if $x \in V_{\text{MAX}}$, a MIN-*position* if $x \in V_{\text{MIN}}$, an AVE-*position* if $x \in V_{\text{AVE}}$, and a *sink* if $x \in \text{V}_{\text{SINK}}$. Each sink x has a rational *payoff* $p(x) \in [0, 1]$. For every $x \in V_{\text{AVE}}$, every edge leaving x is labeled with a probability such that the sum of probabilities over all edges leaving x is one; the probability associated with an edge $(x, y) \in E$, for $x \in V_{\text{AVE}}$, is denoted by $q(x, y)$. The outgoing edges from any vertex $x \in V_{\text{MAX}} \cup V_{\text{MIN}}$ are unlabeled.

Before the start of a play, a token is placed on start vertex x. Each step of the play depends on where the token is located. Suppose that the token is currently at $x' \in V$. If $x' \in V_{\text{MAX}}$, then MAX chooses an outgoing edge. If $x' \in V_{\text{MIN}}$, then MIN chooses an outgoing edge. If $x' \in V_{\text{AVE}}$, then a random outgoing edge is chosen from a distribution $q(x', \cdot)$ over the neighbors of x'. Finally, if $x' \in \text{V}_{\text{SINK}}$, then the play stops and MAX wins from MIN a payoff $p(x') \in [0, 1]$. We refer to a SSG by the name of its game board, i.e., the multigraph G. We denote the game G starting at vertex x by $G(x)$. The above definition generalizes the usual definition of SSGs [Con92,Con93,Lud95,Hal07], which considers two types of sink vertices, 0-sink and 1-sink, but no payoffs. However, as discussed by Condon [Con92], these two definitions are equivalent.

A strategy of a player specifies the choices made by the player during a play of the game. A strategy is *pure* if the choices are made in a deterministic manner, and is *mixed* if the choices are made according to some probability distribution. A pure strategy may depend on time, and, more generally, it may depend on the history of a play. On the other hand, a *stationary* (or *memoryless*) strategy depends only on the current state of a play, and is independent of both time and history. Condon [Con92] showed that both players MAX and MIN have *optimal strategies* (defined in Definition 1), which are pure and stationary. Therefore, in this paper, we consider only pure, stationary strategies of the two players. Formally, a strategy σ of MAX is a function $\sigma : V_{\text{MAX}} \to V$ such that for every $x \in V_{\text{MAX}}$, $(x, \sigma(x)) \in E$. Similarly, a strategy τ of MIN is a function $\tau : V_{\text{MIN}} \to V$ such that for every $x \in V_{\text{MIN}}$, $(x, \tau(x)) \in E$. For every strategy α of a player $P \in \{\text{MAX}, \text{MIN}\}$, we say that a play x_0, x_1, x_2, $\ldots$ of $G(x_0)$ *confirms* to α if, for every $i \geq 0$ and every $x_i \in V_P$, we have $x_{i+1} = \alpha(x_i)$.

For every choice of start vertex x and strategies σ, τ of the players, the expected payoff $v_{\sigma,\tau}(x)$ that MAX wins is defined as $v_{\sigma,\tau}(x) =_{df} \sum_{s \in \text{V}_{\text{SINK}}} q_{\sigma,\tau}(x, s) \cdot p(s)$, where $q_{\sigma,\tau}(x, s)$ denotes the probability that a play of $G(x)$ confirming to σ and τ stops at a sink vertex s. It is implicit in the above definition that $v_{\sigma,\tau}(x) = 0$ if no play of $G(x)$ conforming to σ and τ is finite. A *stopping* SSG is a SSG in which starting at any initial position, every possible play ends at a sink vertex. Stopping SSGs are known to have certain desirable properties (e.g., the existence of a unique optimal value vector). Condon [Con92] showed that there is a polynomial-time transformation that, given any SSG G, constructs a new stopping SSG G' such that G' is as good as G for the purpose of studying the problem SSG-VAL. Since the focus of this paper is to study algorithms for the SSG value problem, we will henceforth assume that a SSG is a stopping SSG.

Given any strategy τ of MIN, a best response (i.e., optimal) strategy $\sigma = \sigma(\tau)$ of MAX w.r.t. τ, if it exists, is one that, for every game position $x \in V_{\mathrm{POS}}$, assures the *maximum* payoff for MAX over all choices of MAX-strategies, i.e., $v_{\sigma,\tau}(x) = \max_{\sigma'} v_{\sigma',\tau}(x)$. Similarly, given any startegy σ of MAX, a best response strategy $\tau = \tau(\sigma)$ of MIN w.r.t. σ, if it exists, is one that, for every game position $x \in V_{\mathrm{POS}}$, assures the *minimum* payoff for MAX over all choices of MIN-strategies, i.e., $v_{\sigma,\tau}(x) = \min_{\tau'} v_{\sigma,\tau'}(x)$. Howard [How60] showed that such strategies always exist, and therefore, $\sigma(\tau)$ and $\tau(\sigma)$ are well defined. Derman's [Der70] LP formulation can be used for constructing these best response strategies for any stopping SSG in polynomial time.

Strategies σ and τ are *optimal strategies* if each is a best response strategy w.r.t. the other. Condon [Con92] showed that every stopping SSG has optimal strategies.

Definition 1 (optimal strategies). *Let σ and τ be strategies of* MAX *and* MIN*, respectively. Strategies σ and τ are* optimal at $x \in V_{\mathrm{POS}}$ *if, for any strategy σ' of* MAX *and for any strategy τ' of* MIN*, it holds that $v_{\sigma',\tau}(x) \leq v_{\sigma,\tau}(x) \leq v_{\sigma,\tau'}(x)$. Strategies σ and τ are* optimal *if they are optimal at every $x \in V_{\mathrm{POS}}$.*

The expected payoff vector corresponding to a pair of optimal strategies is called an *optimal value vector*. Condon [Con92] showed that every stopping SSG has an optimal value vector, which is *unique*.

Definition 2 (optimal value vector). *For any optimal strategies σ and τ at $x \in V_{\mathrm{POS}}$, the expected payoff $v_{\mathrm{opt}}(x) =_{df} v_{\sigma,\tau}(x)$ is called an* optimal value *of $G(x)$. If, for every $x \in V_{\mathrm{POS}}$, there exist optimal strategies at x, then a vector $v_{\mathrm{opt}} : V_{\mathrm{POS}} \to [0,1]$ of optimal values is called an* optimal value vector *of G.*

We call any mapping $v : V_{\mathrm{POS}} \to [0,1]$ a *value vector*. Sometimes, we extend the domain of such a value vector v to $\mathrm{V_{SINK}}$ and define the mapping of any sink vertex s as the payoff $p(s) \in [0,1]$. We use $\overline{v}$ to denote this extension of v.

The *v-switchable* positions, defined in Definition 3, will be useful in constructing an improved value vector from v (see Lemma 14). Notice that only a position $x \in V_{\mathrm{MAX}} \cup V_{\mathrm{MIN}}$ can possibly be a v-switchable position.

Definition 3 (switchable positions). *Let $v : V_{\mathrm{POS}} \to [0,1]$ be a value vector. For any $x \in V_{\mathrm{MAX}} \cup V_{\mathrm{MIN}}$, we say that x is v-*switchable *if either $x \in V_{\mathrm{MAX}}$ and $v(x) < \max\{\overline{v}(y) \mid (x,y) \in E\}$, or $x \in V_{\mathrm{MIN}}$ and $v(x) > \min\{\overline{v}(y) \mid (x,y) \in E\}$.*

Given a value vector v, we say that a player strategy is *v-greedy* if the strategy makes locally optimal choice w.r.t. v at every position of the player.

Definition 4 (greedy strategies). *Let $v : V_{\mathrm{POS}} \to [0,1]$ be a value vector. A* MAX*-strategy σ is said to be v-*greedy at $x \in V_{\mathrm{MAX}}$ *if $\overline{v}(\sigma(x)) = \max\{\overline{v}(y) \mid$

$(x, y) \in E\}$. *Similarly, a* MIN*-strategy* τ *is said to be* v-greedy at $x \in V_{\text{MIN}}$ *if* $\overline{v}(\tau(x)) = \min\{\overline{v}(y) \mid (x, y) \in E\}$. *For* $P \in \{\text{MAX}, \text{MIN}\}$*, a strategy of* P *is said to be* v-greedy *if it is* v*-greedy at each* $x \in V_P$.

Condon [Con92] introduced an operator F_G corresponding to any SSG G. This operator allows to give an alternate characterization of an optimal value vector of G (see Lemmas 6 and 7).

Definition 5 ([Con92]). *Let* σ *and* τ *be strategies of* MAX *and* MIN*, respectively. We define an operator* $F_G : (V_{\text{POS}} \to [0, 1]) \to (V_{\text{POS}} \to [0, 1])$ *as follows: For every* $v : V_{\text{POS}} \to [0, 1]$*,* $F_G(v) = w$ *such that, for every* $x \in V_{\text{POS}}$*,*

$$w(x) = \begin{cases} \max\{\overline{v}(y) \mid (x, y) \in E\} & \text{if } x \in V_{\text{MAX}}, \\ \min\{\overline{v}(y) \mid (x, y) \in E\} & \text{if } x \in V_{\text{MIN}}, \\ \sum_{(x,y)\in E} q(x, y) \cdot \overline{v}(y) & \text{if } x \in V_{\text{AVE}}. \end{cases}$$

Lemma 6 shows that, for any stopping SSG, there is a unique solution to the local optimality equations given by the operator F_G of Definition 5; this solution is also an optimal value vector of G. (Note that a *fixpoint* of an operator F is an element of the domain of F such that $F(x) = x$.) Lemma 6 implies that there always exist optimal strategies and an optimal value vector of a stopping SSG.

Lemma 6 ([Sha53,Con92]). *Let* G *be a* stopping *SSG. Then there is a* unique fixpoint $v_\star : V_{\text{POS}} \to [0, 1]$ *of the operator* F_G*. Moreover,* $v_\star$ *is an optimal value vector of* G*, and* $v_\star$-greedy strategies $\sigma_\star$ *and* $\tau_\star$ *are optimal strategies of* G*.*

Lemma 7 states that any optimal value vector of a stopping SSG is a fixpoint of the operator F_G. The proof of this lemma is similar to the proofs by Howard [How60] and Condon [Con92].

Lemma 7 (see [Con92]). *Let* G *be a* stopping *SSG and let* v_{opt} *be an optimal value vector of* G*. Then* v_{opt} *is a* fixpoint *of the operator* F_G*.*

Lemma 7 implies that there is a unique optimal value vector of a stopping SSG, since F_G has a unique fixpoint by Lemma 6. Therefore, any pair of optimal strategies of a stopping SSG yields the same optimal value vector of the SSG. Henceforth, we refer to an optimal value vector of a stopping SSG as *the* optimal value vector of the game.

The proof of the following lemma can be found in [Con92,Som05].

Lemma 8 (see [Con92,Som05]). *Let* G *be a* stopping *SSG and let* v_{opt} *be* the optimal value vector *of* G*. Then the following statements are equivalent:*

1. *strategies* σ *and* τ *are optimal,*
2. $v_{\sigma,\tau}(x) = \max_\sigma \min_\tau v_{\sigma,\tau}(x) = \min_\tau \max_\sigma v_{\sigma,\tau}(x)$ *for every* $x \in V$*,*
3. $v_{\sigma,\tau} = v_{\text{opt}}$*,*
4. *strategies* σ *and* τ *are* $v_{\sigma,\tau}$*-greedy for* MAX *and* MIN*, respectively,*
5. *strategies* σ *and* τ *are* v_{opt}*-greedy for* MAX *and* MIN*, respectively.*

Definition 9 ([Con92]). *The value of a SSG and the SSG value problem* (SSG-VAL) *are defined as follows:*

1. *The* value *of a SSG* G *is* $\max_\sigma \min_\tau v_{\sigma,\tau}(\text{start vertex})$.
2. SSG-VAL $\equiv$ *Given a SSG* G*, is the value of* $G > 1/2$*?*

3 Results

The strategy improvement method is an iterative procedure for constructing optimal strategies within a finite number of iterations in a decision-making scenario (e.g., game). This technique was developed first in the context of Markov decision processes, which are SSGs that have only MAX-positions and AVE-positions, but no MIN-positions. In Section 3.1, we study a strategy improvement algorithm developed by Hoffman and Karp [HK66] (Algorithm 1) for solving the SSG value problem.

The Hoffman-Karp algorithm starts with an initial pair of player strategies. It then iteratively computes new player strategies until a pair of optimal strategies is found. W.l.o.g. assume that $|V_{\text{MAX}}| \leq |V_{\text{MIN}}|$ and let $n = \min\{|V_{\text{MAX}}|, |V_{\text{MIN}}|\}$. In each iteration, the current strategy σ of MAX is changed to σ' by switching the choices of all MAX-positions at which local optimality is not achieved, while the MIN-strategy is always the best response strategy $\tau(\sigma')$ w.r.t. the new MAX-strategy σ'. The algorithm terminates when all MAX-positions (and also MIN-positons) are locally optimal. At this point, the value vector $v_{\sigma,\tau(\sigma)}$ corresponding to the current MAX-strategy σ and MIN-strategy $\tau(\sigma)$ will satisfy the local optimality equations given by the operator F_G (Definition 5), and so it will be a fixpoint of F_G.

Algorithm 1: The Hoffman-Karp Algorithm [HK66]

Input : A stopping SSG G
Output: optimal strategies σ, τ and the optimal value vector v_{opt}

begin
  Let σ and τ be arbitrary strategies of MAX and MIN, respectively
  while $(F_G(v_{\sigma,\tau}) \neq v_{\sigma,\tau})$ **do**
    Let σ' be obtained from σ by switching the choices of all $v_{\sigma,\tau}$-switchable MAX-positions
    Let τ' be an optimal strategy of MIN w.r.t. σ'
    Set $\sigma \leftarrow \sigma'$, $\tau \leftarrow \tau'$
  Output σ, τ and the optimal value vector $v_{\sigma,\tau}$
end

Each iteration of the while loop of Algorithm 1 requires (a) computing $v_{\sigma,\tau}$ given player strategies σ and τ, and (b) computing an optimal MIN-strategy τ' w.r.t. MAX-strategy σ'. Both (a) and (b) are known to take polynomial time. It follows that every iteration of the while loop executes in polynomial time. Thus, it only remains to prove the correctness of the algorithm and bound the number of iterations of the while loop.

The proof of correctness is based on a property of value vectors, which is formally stated in Lemma 14. Using this property, it can be shown that in every iteration, the new value vector $v_{\sigma',\tau'}$ improves upon the initial vector $v_{\sigma,\tau}$ in the following sense: for all positions x, $v_{\sigma,\tau}(x) \leq v_{\sigma',\tau'}(x)$, and for all MAX-switchable positions, this inequality is strict. Thus, it follows that no MAX-strategy can repeat over all the iterations of the algorithm. Since there can be at most 2^n distinct MAX-strategies in a (binary) SSG with n MAX-positions, this algorithm requires at most 2^n iterations of the while loop in the worst case.

It is important to study the convergence time (i.e., the number of iterations) of strategy improvement algorithms for solving SSGs (e.g., the Hoffman-Karp algorithm and its many variants). These algorithms are not complicated from implementation perspective, and so a nontrivial upper bound on their convergence time might have a practical value. Melekopoglou and Condon [MC94] showed that many variations of the Hoffman-Karp algorithm require $\Omega(2^n)$ iterations in the worst case. In these variations, *only one* MAX-switchable position is switched at every iteration as opposed to *all* MAX-switchable positions in the Hoffman-Karp algorithm. In a recent breakthrough, Friedmann [Fri09] presented a super-polynomial lower bound for the discrete strategy improvement algorithm of Vöge and Jurdziński [VJ00] for solving parity games.

In Section 3.1, we prove that the Hoffman-Karp algorithm requires $O(2^n/n)$ iterations in the worst case. This is the first non-trivial upper bound on the worst-case convergence time of this algorithm. In Section 3.2, we propose a randomized variant of the Hoffman-Karp algorithm and prove that with probability almost one the randomized strategy improvement algorithm requires $O(2^{0.78n})$ iterations in the worst case. Our analyses in these sections extend those of Mansour and Singh [MS99] for policy improvement algorithms for Markov decision processes.

We now present some definitions and results, which will be used in the remaining part of this paper.

Notation 10 *For every* MAX*-strategy* σ*,* S_σ *denotes the set of all* $v_{\sigma,\tau(\sigma)}$-switchable positions *of* G*.*

Definition 11. *For every* MAX*-strategy* σ *and* $S \subseteq V_{\mathrm{MAX}}$*, let* $\mathrm{switch}(\sigma, S) : V_{\mathrm{MAX}} \to V$ *be a* MAX*-strategy obtained from* σ *by switching the choices of the positions of* S *only. That is, for every* $x \in V_{\mathrm{MAX}}$ *such that* (x, y)*,* $(x, z) \in E$ *and* $y = \sigma(x)$*,* $\mathrm{switch}(\sigma, S)(x) = y$ *if* $x \notin S$*, and* $\mathrm{switch}(\sigma, S)(x) = z$ *if* $x \in S$*.*

Definition 12. *Let* u*,* $v \in [0, 1]^N$*, for some* $N \in \mathbb{N}^+$*. We say that*

- $u \preceq v$ *if for each* $x \in [N]$*, it holds that* $u(x) \leq v(x)$*.*
- $u \prec v$ *if* $u \preceq v$ *and there is an* $x \in [N]$ *such that* $u(x) < v(x)$*.*
- u *and* v *are incomparable if there are* $x, y \in [N]$ *such that* $u(x) < v(x)$ *and* $v(y) < u(y)$*.*
- $u \npreceq v$ *if either* $v \prec u$*, or* u *and* v *are incomparable.*
- $u \nprec v$ *if either* $v \preceq u$*, or* u *and* v *are incomparable.*

Fact 13 (see [Juk01]) *Let* $H(x) =_{df} -x \log_2 x - (1-x) \log_2(1-x)$ *for* $0 < x < 1$ *and* $H(0) = H(1) = 0$*. Then,* $\forall s : 0 \leq s \leq t/2$*, we have* $\sum_{k=0}^{s} \binom{t}{k} \leq 2^{t \cdot H(s/t)}$*.*

We establish some properties of value vectors, which are crucial in the analyses presented in Sections 3.1 and 3.2. Because of the space limitations, the proofs of Lemmas 14, 15, and 16 are omitted. They will appear in the full version of the paper.

Lemma 14. *Let σ be a strategy of* MAX *such that S_σ is nonempty and let S be any nonempty subset of S_σ. Let $\sigma' =_{df} \text{switch}(\sigma, S)$ be a* MAX*-strategy. Then, it holds that $v_{\sigma,\tau(\sigma)} \prec v_{\sigma',\tau(\sigma')}$.*

Lemma 15. *Let σ and σ' be* MAX*-strategies such that σ' is obtained from σ by switching a single position $x \in V_{\text{MAX}}$, i.e., $\sigma' = \text{switch}(\sigma, \{x\})$. Then, $v_{\sigma,\tau(\sigma)}$ and $v_{\sigma',\tau(\sigma')}$ are not incomparable, i.e., either $v_{\sigma,\tau(\sigma)} \prec v_{\sigma',\tau(\sigma')}$ or $v_{\sigma',\tau(\sigma')} \preceq v_{\sigma,\tau(\sigma)}$. Moreover, we have $v_{\sigma,\tau(\sigma)} \prec v_{\sigma',\tau(\sigma')}$ if and only if x is $v_{\sigma,\tau(\sigma)}$-switchable.*

3.1 An Improved Analysis of the Hoffman-Karp Algorithm

In this section, we prove that the number of iterations in the worst case required by the Hoffman-Karp strategy improvement algorithm is $O(2^n/n)$. This improves upon the previously known, trivial, worst-case upper bound 2^n for this algorithm.

Lemma 16 shows that the set of all switchable MAX-positions differs from one iteration to another in the Hoffman-Karp strategy improvement algorithm.

Lemma 16. *Let $\sigma_i, \tau(\sigma_i)$ and $\sigma_j, \tau(\sigma_j)$ be player strategies at the start of iterations i and j, respectively, for $i < j$, of the* Hoffman-Karp *algorithm. Then, it holds that $S_{\sigma_i} \nsubseteq S_{\sigma_j}$.*

In the Hoffman-Karp algorithm, the value vectors monotonically increase with the number of iterations by Lemma 14. Thus, Lemma 17 shows that this algorithm rules out, at the end of every iteration, a number of MAX-strategies, which is at least *linear* in the number of switchable MAX-positions.

Lemma 17. *Let σ, $\tau(\sigma)$ and $\sigma' =_{df} \text{switch}(\sigma, S_\sigma)$, $\tau' =_{df} \tau(\sigma')$ be player strategies at the start and at the end, respectively, of an iteration of the* Hoffman-Karp *algorithm. Then, there are at least $|S_\sigma| - 1$ strategies σ_i, $\tau(\sigma_i)$ such that $v_{\sigma,\tau} \prec v_{\sigma_i,\tau(\sigma_i)} \preceq v_{\sigma',\tau'}$.*

Proof. Let the elements of S_σ be denoted by 1, 2, …, $|S_\sigma|$. For every $S \subseteq S_\sigma$, let $\sigma_S =_{df} \text{switch}(\sigma, S)$ be a MAX-strategy. For notational convenience, let τ_S denote $\tau(\sigma_S)$, which is an optimal MIN-strategy w.r.t. σ_S.

Assume w.l.o.g. that $v_{\sigma_{\{1\}},\tau_{\{1\}}}$ is a minimal value vector among the set of value vectors $v_{\sigma_{\{i\}},\tau_{\{i\}}}$ for $1 \le i \le n$. In other words, we assume that for every $2 \le i \le |S_\sigma|$, we have either $v_{\sigma_{\{1\}},\tau_{\{1\}}} \preceq v_{\sigma_{\{i\}},\tau_{\{i\}}}$ or $v_{\sigma_{\{i\}},\tau_{\{i\}}}$ is incomparable to $v_{\sigma_{\{1\}},\tau_{\{1\}}}$. From Lemma 14, we know that $v_{\sigma,\tau} \prec v_{\sigma_{\{1\}},\tau_{\{1\}}}$.

Claim 1 *For every $2 \le i \le |S_\sigma|$, it holds that $v_{\sigma_{\{1\}},\tau_{\{1\}}} \preceq v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}}$.*

For now assume that Claim 1 is true. (We prove this claim at the end of the proof of Lemma 17.) Pick a minimal value vector, say $v_{\sigma_{\{1,2\}},\tau_{\{1,2\}}}$, among the set of value vectors $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}}$ for $2 \leq i \leq n$. This gives the sequence

$$v_{\sigma,\tau} \prec v_{\sigma_{\{1\}},\tau_{\{1\}}} \preceq v_{\sigma_{\{1,2\}},\tau_{\{1,2\}}}.$$

Repeating the arguments of Claim 1 with $v_{\sigma_{\{1,2\}},\tau_{\{1,2\}}}$ in place of $v_{\sigma_{\{1\}},\tau_{\{1\}}}$ gives the following statement: for every $3 \leq i \leq |S_\sigma|$, it holds that $v_{\sigma_{\{1,2\}},\tau_{\{1,2\}}} \preceq v_{\sigma_{\{1,2,i\}},\tau_{\{1,2,i\}}}$. Next, we pick a minimal vector, say $v_{\sigma_{\{1,2,3\}},\tau_{\{1,2,3\}}}$, among the set of value vectors $v_{\sigma_{\{1,2,i\}},\tau_{\{1,2,i\}}}$ for $3 \leq i \leq |S_\sigma|$. This gives the sequence

$$v_{\sigma,\tau} \prec v_{\sigma_{\{1\}},\tau_{\{1\}}} \preceq v_{\sigma_{\{1,2\}},\tau_{\{1,2\}}} \preceq v_{\sigma_{\{1,2,3\}},\tau_{\{1,2,3\}}}.$$

By proceeding in the above manner, we can obtain a monotonically increasing sequence $v_{\sigma,\tau} \prec v_{\sigma_{\{1\}},\tau_{\{1\}}} \preceq v_{\sigma_{\{1,2\}},\tau_{\{1,2\}}} \preceq v_{\sigma_{\{1,2,3\}},\tau_{\{1,2,3\}}} \preceq \cdots \preceq v_{\sigma_{S_\sigma},\tau_{S_\sigma}} = v_{\sigma',\tau'}$. This completes the proof of the lemma. □ (Lemma 17)

We now give a proof of Claim 1.

Proof of Claim 1. Assume to the contrary that, for some $2 \leq i \leq |S_\sigma|$, we have $v_{\sigma_{\{1\}},\tau_{\{1\}}} \not\preceq v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}}$. Then, it must be the case that either $v_{\sigma_{\{1\}},\tau_{\{1\}}}$ and $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}}$ are incomparable or $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}} \prec v_{\sigma_{\{1\}},\tau_{\{1\}}}$. Since $\sigma_{\{1,i\}} = \text{switch}(\sigma_{\{1\}}, \{i\})$, Lemma 15 implies that $v_{\sigma_{\{1\}},\tau_{\{1\}}}$ and $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}}$ are not incomparable. Therefore, we must have $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}} \prec v_{\sigma_{\{1\}},\tau_{\{1\}}}$.

We next show that $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}} \prec v_{\sigma_{\{i\}},\tau_{\{i\}}}$. Notice that $\sigma_{\{1,i\}} = \text{switch}(\sigma_{\{i\}}, \{1\})$. By Lemma 15, we must have either $v_{\sigma_{\{i\}},\tau_{\{i\}}} \preceq v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}}$ or $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}} \prec v_{\sigma_{\{i\}},\tau_{\{i\}}}$. If the former holds, then by transitivity of $\preceq$, we get that $v_{\sigma_{\{i\}},\tau_{\{i\}}} \preceq v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}} \prec v_{\sigma_{\{1\}},\tau_{\{1\}}}$. However, the relation $v_{\sigma_{\{i\}},\tau_{\{i\}}} \prec v_{\sigma_{\{1\}},\tau_{\{1\}}}$ contradicts the minimality of $v_{\sigma_{\{1\}},\tau_{\{1\}}}$. Therefore, we must have $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}} \prec v_{\sigma_{\{i\}},\tau_{\{i\}}}$.

Thus, we have shown that $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}} \prec v_{\sigma_{\{1\}},\tau_{\{1\}}}$ and $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}} \prec v_{\sigma_{\{i\}},\tau_{\{i\}}}$. We claim that both $1, i \in S_{\sigma_{\{1,i\}}}$. To see this, note that $\sigma_{\{1\}} = \text{switch}(\sigma_{\{1,i\}}, \{i\})$ and $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}} \prec v_{\sigma_{\{1\}},\tau_{\{1\}}}$. So, by Lemma 15, we must have $i \in S_{\sigma_{\{1,i\}}}$. In the same way, we must have $1 \in S_{\sigma_{\{1,i\}}}$.

We now know that $\sigma = \text{switch}(\sigma_{\{1,i\}}, \{1,i\})$, where $\{1,i\} \subseteq S_{\sigma_{\{1,i\}}}$. Lemma 14 implies that we must have $v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}} \prec v_{\sigma,\tau}$. However, we also have $\{1,i\} \subseteq S_\sigma$ and $\sigma_{\{1,i\}} = \text{switch}(\sigma, \{1,i\})$. So, Lemma 14 also implies that $v_{\sigma,\tau} \prec v_{\sigma_{\{1,i\}},\tau_{\{1,i\}}}$. This gives a contradiction. □ (Claim 1)

We are now ready to prove the main result of Section 3.1.

Theorem 18. *The* Hoffman-Karp *algorithm requires at most* $O(2^n/n)$ *iterations in the worst case, where* $n = \min\{|V_{\text{MAX}}|, |V_{\text{MIN}}|\}$.

Proof. W.l.o.g. ssume that $n = |V_{\text{MAX}}| \leq |V_{\text{MIN}}|$. We partition the analysis of the number of iterations into two cases: (1) iterations in which $|S_\sigma| \leq n/3$ and (2) iterations in which $|S_\sigma| > n/3$.

By Lemma 16, the set of all switchable MAX-positions cannot repeat throughout the iterations of the Hoffman-Karp algorithm. Therefore, in case (1), the number of iterations in which $|S_\sigma| \le n/3$ is bounded by $\sum_{k=0}^{n/3} \binom{n}{k}$, which is at most $2^{n \cdot H(1/3)}$ by Fact 13. In case (2), since $|S_\sigma| > n/3$ in each such iteration, by Lemma 17 the Hoffman-Karp algorithm rules out at least $n/3$ strategies σ_i such that $v_{\sigma,\tau} \prec v_{\sigma_i,\tau(\sigma_i)} \preceq v_{\sigma',\tau'}$. (Note that σ, τ refers to the current strategy pairs and σ', τ' refers to the strategy pairs in the next iteration.) Therefore, the number of iterations in which $|S_\sigma| > n/3$ is bounded by $2^n/(n/3) = 3 \cdot 2^n/n$.

It follows that the Hoffman-Karp algorithm requires at most $2^{n \cdot H(1/3)} + 3 \cdot 2^n/n \le 4 \cdot 2^n/n$ iterations in the worst case. □ (Theorem 18)

3.2 A Randomized Variant of the Hoffman-Karp Algorithm

We propose a randomized strategy improvement algorithm (Algorithm 2) for the problem SSG-VAL (Definition 9). This algorithm can be seen as a variation of the Hoffman-Karp algorithm in that, instead of deterministically choosing all switchable MAX-positions, the randomized algorithm chooses a uniformly random subset of the switchable MAX-positions in each iteration. Similar to the results in Section 3.1, our results in this section extend those of Mansour and Singh [MS99] for policy iteration algorithms for Markov decison processes. We mention that Condon [Con93] also presented a randomized variant of the Hoffman-Karp algorithm, which is different from ours. The expected number of iterations of Condon's algorithm is at most $2^{n-f(n)} + 2^{o(n)}$, for any function $f(n) = o(n)$, where $n = \min\{|V_{\mathrm{MAX}}|, |V_{\mathrm{MIN}}|\}$.

Algorithm 2: A Randomized Variant of the Hoffman-Karp Algorithm

Input : A stopping SSG G
Output: Optimal strategies σ, τ and the optimal value vector v_{opt}

begin
  Let σ and τ be arbitrary strategies of MAX and MIN, respectively
  while $(F_G(v_{\sigma,\tau}) \neq v_{\sigma,\tau})$ **do**
    $S \leftarrow$ a uniformly random subset of $v_{\sigma,\tau}$-switchable MAX-positions
    Let $\sigma' \leftarrow \mathrm{switch}(\sigma, S)$
    Let τ' be an optimal strategy of MIN w.r.t. σ'
    Set $\sigma \leftarrow \sigma'$, $\tau \leftarrow \tau'$
  Output σ, τ and the optimal value vector $v_{\sigma,\tau}$
end

Lemma 19 shows that if a MAX-strategy σ' does not yield an improved value vector $v_{\sigma',\tau(\sigma')}$ compared to the value vector $v_{\sigma_{i+1},\tau_{i+1}}$ at the start of iteration $i+1$, then σ' cannot appear after iteration $i+1$.

Lemma 19. *In each iteration i of Algorithm 2, let $\sigma_i, \tau(\sigma_i)$ be player strategies at the start of this iteration. Let $S \subseteq S_{\sigma_i}$ be a subset of $v_{\sigma_i,\tau(\sigma_i)}$-switchable* MAX-*positions and let $\sigma' =_{df}$* switch(σ_i, S) *be a* MAX-*strategy. If $v_{\sigma_{i+1},\tau(\sigma_{i+1})} \not\prec v_{\sigma',\tau(\sigma')}$, then for any $i+2 \le j$, $\sigma' \neq \sigma_j$.*

Proof. By Lemma 14, for each iteration j, it holds that $v_{\sigma_{j-1},\tau(\sigma_{j-1})} \prec v_{\sigma_j,\tau_j}$. Assume to the contrary that, for some $i+2 \le j$, $\sigma' = \sigma_j$. Then, by transitivity, we have $v_{\sigma_{i+1},\tau(\sigma_{i+1})} \preceq v_{\sigma_{j-1},\tau_{j-1}} \prec v_{\sigma_j,\tau(\sigma_j)} = v_{\sigma',\tau(\sigma')}$. This contradicts the assumption that $v_{\sigma_{i+1},\tau(\sigma_{i+1})} \not\prec v_{\sigma',\tau(\sigma')}$. ☐ (Lemma 19)

Lemma 20 shows that by switching the choices at a uniformly random subset of switchable MAX-positions, we can rule out (at the end of every iteration) a number of MAX-strategies, which on average is at least *exponential* in the number of switchable MAX-positions.

Lemma 20. *In Algorithm 2, let σ_i, $\tau(\sigma_i)$ be player strategies at the start of an iteration in which S_{σ_i} is nonempty, and let σ_{i+1}, $\tau(\sigma_{i+1})$ be the player strategies at the end of this iteration. Then, the expected number of* MAX-*strategies σ' such that $v_{\sigma_{i+1},\tau(\sigma_{i+1})} \not\prec v_{\sigma',\tau(\sigma')}$ is at least $2^{|S_{\sigma_i}|-1}$.*

Proof. Consider an iteration in which σ_i, $\tau(\sigma_i)$ are player strategies such that S_{σ_i} is nonempty. Let U denote the set of all MAX-strategies obtained from σ_i by switching some subset of S_{σ_i}, i.e., $U = \{\sigma \mid (\exists S \subseteq S_{\sigma_i})[\sigma = \mathrm{switch}(\sigma_i, S)]\}$. Clearly, $|U| = 2^{|S_{\sigma_i}|}$. For each strategy $\sigma \in U$, we associate sets U_σ^+ and U_σ^- that are defined as follows:

$$U_\sigma^+ =_{df} \{\sigma' \in U \mid v_{\sigma,\tau(\sigma)} \prec v_{\sigma',\tau(\sigma')}\} \text{ and } U_\sigma^- =_{df} \{\sigma' \in U \mid v_{\sigma',\tau(\sigma')} \prec v_{\sigma,\tau(\sigma)}\}.$$

Notice that, for any pair σ, $\sigma' \in U$, we have $\sigma' \in U_\sigma^+$ if and only if $\sigma \in U_{\sigma'}^-$. From this equivalence, it follows that $\sum_{\sigma\in U} |U_\sigma^+| = \sum_{\sigma\in U} |U_\sigma^-| \le \frac{|U|^2}{2}$. Thus, for a strategy σ_{i+1} chosen uniformly at random from U in iteration i, the expected number of MAX-strategies $\sigma' \in U$ such that $v_{\sigma_{i+1},\tau(\sigma_{i+1})} \not\prec v_{\sigma',\tau(\sigma')}$ is $= |U| - \frac{1}{|U|} \cdot \sum_{\sigma\in U} |U_\sigma^+| \ge \frac{|U|}{2} = 2^{|S_{\sigma_i}|-1}$. ☐ (Lemma 20)

We are now ready to prove the main result of Section 3.2.

Theorem 21. *With probability at least $1-2^{-2^{\Omega(n)}}$, Algorithm 2 requires at most $O(2^{0.78n})$ iterations in the worst case, where $n = \min\{|V_{\mathrm{MAX}}|, |V_{\mathrm{MIN}}|\}$.*

Proof. Let $c \in (0, 1/2)$ that we will fix later in the proof. As in the proof of Theorem 18, the number of iterations i in which $\sigma_i, \tau(\sigma_i)$ are player strategies and $|S_{\sigma_i}| \le cn$ is bounded by $\sum_{k=0}^{cn} \binom{n}{k}$, which is at most $2^{n \cdot H(c)}$ by Fact 13.

We next bound the number of iterations i in which $\sigma_i, \tau(\sigma_i)$ are player strategies and $|S_{\sigma_i}| > cn$. Let $\sigma_{i+1}, \tau(\sigma_{i+1})$ be player strategies at the end of iteration i. By Lemma 20, the expected number of MAX-strategies σ' such that $v_{\sigma_{i+1},\tau(\sigma_{i+1})} \not\prec v_{\sigma',\tau'}$ is at least $2^{|S_{\sigma_i}|-1}$. By Lemma 19, none of these strategy pairs $\sigma', \tau(\sigma')$ is chosen in any future iteration. Therefore, the expected number of MAX-strategies σ' that Algorithm 2 rules out in each such iteration is at least $2^{|S_{\sigma_i}|-1} \ge 2^{cn}$. It follows from Markov's inequality that, with probability at least $1/2$, Algorithm 2 rules out at least 2^{cn-1} MAX-strategies in each such iteration.

We say that an iteration in which $|S_{\sigma_i}| > cn$ is *good* if Algorithm 2 rules out at least 2^{cn-1} MAX-strategies at the end of it. We know from above that the probability that an iteration in which $|S_{\sigma_i}| > cn$ is good is at least 1/2. By Chernoff bounds, for any $t > 0$, at least 1/4 of the t iterations in which $|S_{\sigma_i}| > cn$ will be good with probability at least $1 - e^{-t/16}$. Thus, the total number of iterations is at most $2^{n \cdot H(c)} + 2^{n \cdot (1-c)+3}$ with a high probability. Choosing $c \in (0, 1/2)$ such that $H(c) = 1 - c$ gives $c \approx 0.227$. For $c = 0.227$, this number of iterations is bounded by $O(2^{0.78n})$. Also, when the number of iterations (and so t) is at most $2^{0.78n}$, then the probability of success is $\geq 1 - 2^{-2^{\Omega(n)}}$. $\square$ (Theorem 21)

Acknowledgements: V.S. Anil Kumar was supported partially by the following grants: DTRA CNIMS Grant HDTRA1-07-C-0113, NSF Nets Grant CNS-0626964, NSF HSD Grant SES-0729441, NSF NETS CNS-0831633, and NSF CNS-0845700.

References

[Con92] A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.

[Con93] A. Condon. On algorithms for simple stochastic games. In J. Cai, editor, *Advances in Computational Complexity Theory*, volume 13, pages 51–73. DIMACS series in DM&TCS, American Mathematical Society, 1993.

[Der70] C. Derman. *Finite State Markovian Decision Processes*, volume 67 of *Mathematics in Science and Engineering*. Academic Press, New York, 1970.

[Fri09] O. Friedmann. An exponential lower bound for the parity game strategy improvement algorithm as we know it. In *Proceedings of the 24th IEEE Symposium on LICS*. IEEE Computer Society, 2009. To appear.

[GH09] H. Gimbert and F. Horn. Solving simple stochastic games with few random vertices. In *Foundations of Software Science and Comput. Structures*, 2009.

[Hal07] N. Halman. Simple stochastic games, parity games, mean payoff games and DPGs are all LP-type problems. *Algorithmica*, 49(1):37–50, 2007.

[HK66] A. Hoffman and R. Karp. On nonterminating stochastic games. *Management Science*, 12:359–370, 1966.

[How60] R. Howard. *Dynamic Programming and Markov Processes*. M.I.T. Press, Cambridge, MA, 1960.

[Juk01] S. Jukna. *Extremal Combinatorics*. Springer, 2001.

[Lud95] W. Ludwig. A subexponential randomized algorithm for the simple stochastic game problem. *Information and Computation*, 117(1):151–155, 1995.

[MC94] M. Melekopoglou and A. Condon. On the complexity of the policy improvement algorithm for Markov decision processes. *ORSA Journal of Computing*, 6(2):188–192, 1994.

[MS99] Y. Mansour and S. Singh. On the complexity of Policy Iteration. In *Proceedings of the 15th Conference on UAI*, pages 401–408, July 1999.

[Sha53] L. Shapley. Stochastic games. In *Proceedings of National Academy of Sciences (U.S.A.)*, volume 39, pages 1095–1100, 1953.

[Som05] R. Somla. New algorithms for solving simple stochastic games. *Electronic Notes in Theoretical Computer Science*, 119(1):51–65, 2005.

[VJ00] J. Vöge and M. Jurdziński. A discrete strategy improvement algorithm for solving parity games. In *Proceedings of the 12th International Conference on CAV*, pages 202–215. Springer Verlag *LNCS #1855*, 2000.