Math. Program., Ser. A 95: 513–532 (2003)

Digital Object Identifier (DOI) 10.1007/s10107-002-0312-3

T.E.S. Raghavan · Zamir Syed

A policy-improvement type algorithm for solving zero-sum two-person stochastic games of perfect information

Received: January 1998 / Accepted: May 2002
Published online: February 14, 2003 –

Abstract. We give a policy-improvement type algorithm to locate an optimal pure stationary strategy for discounted stochastic games with perfect information. A graph theoretic motivation for our algorithm is presented as well.

Key words. stochastic games – MDP – perfect information – policy iteration

1. Introduction

Discounted stochastic games were first introduced by Shapley [1953]. In a stochastic game Γ, we have a finite set of states $S = \{1, 2, \ldots, s\}$, and for each state $t \in S$ there are two finite sets $A(t) = \{1, 2, \ldots, a(t)\}$ and $B(t) = \{1, 2, \ldots, b(t)\}$ called the action sets for players I and II respectively. For each triple (t, a, b) with $a \in A(t)$ and $b \in B(t)$ there is an immediate reward $r(t, a, b)$ as well as a probability distribution $p(t, a, b)$ on the set S. Given an initial starting state $t_0 \in S$, the game is played as follows. The players simultaneously choose actions $a_0 \in A(t_0)$ and $b_0 \in B(t_0)$ resulting in the payment $r(t_0, a_0, b_0)$ to player I by player II. The system moves to a new state t_1 according to $p(t_0, a_0, b_0)$ and the players again choose actions $a_1 \in A(t_1)$ and $b_1 \in B(t_1)$. Accordingly the payment $r(t_1, a_1, b_1)$ is made to player I by player II and the game moves to a new state t_2 according to $p(t_1, a_1, b_1)$ and so on. The game continues infinitely and the rewards $r(t_n, a_n, b_n)$ are recorded. A general strategy for a player would be a function from the set of all possible histories into the set of probability distributions over the player's action space. A general strategy can therefore be very complicated but nevertheless, given a pair of strategies (π, ρ) for both players, we can, for $0 \leq \beta < 1$ evaluate the expected β-discounted value:

$$\phi_\beta(\pi, \rho)(t_0) = \sum_{n=0}^{\infty} \beta^n r_n(t_0, \pi, \rho) \tag{1.1}$$

where t_0 is the starting state and $r_n(t_0, \pi, \rho)$ is the expected reward (to player I) at the nth stage when the players are using π and ρ. For any two vectors u and v we write $u \leq v$ to mean $u_i \leq v_i$ for all coordinates i. Under this payoff we can define an *optimal*

T.E.S. Raghavan: Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago, e-mail: ter@uic.edu

Partially Funded by NSF Grant DMS 930-1052 and DMS 970-4951

Z. Syed: The Hull Group L.L.C, Chicago, IL 60606, e-mail: zsyed@hdc.com

pair of strategies to be a pair (π^*, ρ^*) such that we have the vector inequality:

$$\phi_\beta(\pi, \rho^*) \leq \phi_\beta(\pi^*, \rho^*) \leq \phi_\beta(\pi^*, \rho) \tag{1.2}$$

for all π and ρ (we also call them *optimal strategies for the two players* and refer to such pairs as *saddle points*). A strategy is said to be *stationary* if it only depends on the current state. Shapley [1953] showed that under the discounted payoff criterion, there always exists an optimal pair in stationary strategies, and further, $\phi_\beta(\Gamma) = \phi_\beta(\pi^*, \rho^*)$ the optimal payoff vector, also called the *value of the stochastic game*, is unique. This allows us to write the value vector as $\phi_\beta(\Gamma) = \phi_\beta(\pi^*, \rho^*)$.

Successive approximation methods for finding optimal stationary strategies of discounted stochastic games are well known (Van der Waal [1977]). The question of when, for a specific class of stochastic game, finite step algorithm exists is generally open. In this paper we consider the special class of discounted stochastic games with *perfect information*. In perfect information games, at each state at most one player has more than one action to choose from his action set. If the player who has one action is the same one in all states then it is the classical Markovian Decision Process (MDP). One can solve the discounted MDP via Howard's policy improvement algorithm (Howard [1960]). Our task here is to adapt policy improvement algorithm of discounted MDP to these games. The existence theorem for perfect information stochastic games imposes a strong combinatorial structure on them. This then serves as a motivation for our algorithm, which is an extension of Howard-Blackwell [1962] policy improvement algorithm for the discounted stochastic game.

A second evaluation of the infinite stream of immediate payoffs is the Cesaro average payoff, also called the undiscounted payoff. This is defined by

$$\phi(\pi, \rho)(t_0) = \liminf_{N\to\infty} \frac{1}{N+1} \sum_{n=0}^{N} r_n(t_0, \pi, \rho) \tag{1.3}$$

For the special case of stochastic games with perfect information in undiscounted payoffs we have pure stationary optimal strategies for the two players. (Liggett and Lipman [1969]). Even for the case of MDP with undiscounted payoffs, adapting the policy iteration algorithm is not straightforward. See Derman [1970]). Some strong assumption on induced ergodic structure like all pure stationary policies induce a single ergodic class with no transient states, simplifies the search via policy iteration considerably. For policy iteration algorithm for general undiscounted MDP, see (Derman [1970], Veinott [1966], Miller and Veinott [1969]). A linear programming algorithm by Hordijk and Kallenberg [1979] computes the value for MDP with undiscounted payoff. The problem of extending policy iteration algorithm for undiscounted stochastic games of perfect information is still an open problem. Hoffman and Karp [1966] gave an iterative algorithm to find the value and stationary optimal strategies for all irreducible stochastic games. For a survey on algorithms for stochastic games, see Raghavan and Filar [1991]. Also see Filar and Vrieze [1996].

From the point of view of complexity theory, cyclic games and perfect information stochastic games in *undiscounted and total payoffs* have been of interest to computer scientists. (See Condon [1992], Gurwich, Karzanov and Khachiyan [1988], Ludwig [1995], Melekopoglou and Condon [1994], Zwick and Peterson [1996]).

Condon [1992] studied the so called *simple stochastic games* (SSG). These are special classes of stochastic games called *recursive games of perfect information* (Everett [1957]). In recursive games the immediate payoff is 0 at all non-absorbing states. In simple stochastic games, one further assumes that the immediate payoff is 1 just at one absorbing state called the *I-sink* and the immediate payoff is 0 at exactly one absorbing state called the *II-sink*. Depending on the maximizer, or minimizer or nature who alone has more than one action in that state, they are called *max, min* and *average* states. In average states a coin is tossed to choose one of two available actions. The maximizer prefers the game to get absorbed into I-sink. The minimizer prefers the game to get absorbed into II-sink. Also in these games, the law of motion is exact, in the sense, from say, a max state, depending on the action of the player, the game moves to another state in unit time with probability one.

Condon gives a polynomial time value iteration algorithm when the game has exact law of motion and there are no average states present. The algorithm is quite simple and intuitive. If from any max state t there is an action taking the game to a state with value 1, we define the value at state t as $v(t) = 1$. If both actions at a max state end in a state with value 0, we define $v(t) = 0$. Similar definitions apply at min states. If from an average state τ one reaches the two states t or t' with values $v(t)$, $v(t')$ respectively, we define the value $v(\tau) = \frac{1}{2}v(t) + \frac{1}{2}v(t')$. Since the value is known at the two sinks, the value is determined at all other states by backward induction. She also presents an algorithm for the case when the game terminates with probability one into one of the two sinks. For simple stochastic games a policy iteration type algorithm is given by Ludwig [1995]. He also assumes that for every strategy pair the termination occurs with probability 1 in one of the two sinks. Ludwig's algorithm can be described as follows:

1. Given a simple stochastic game with N states start with an arbitrary pure stationary σ for player I and a random max state s. Consider the substochastic game where player I will follow $\sigma(s)$ when state s is reached.
2. Recursively solve for an optimal strategy σ' of the substochastic game for player I and extend this to a strategy of the original game by setting $\sigma'(s) = \sigma(s)$.
3. Solve for an optimal τ' for the MDP for player II (minimizer) with player I's strategy fixed at σ'.
4. If σ', τ' is optimal, then stop. Otherwise, change the alternative at state s to the second available alternative for player I and set $\sigma(t) = \sigma'(t)$ for $t \neq s$. Go to step 1. again.

Even though no polynomial type algorithm exists for these games, interestingly the above algorithm is subexponential in terms of expected number of operations. If S is the set of max vertices, N is the set of min vertices and n is the total number of all vertices, Ludwig, shows that the expected number of operations required by the algorithm is $2^{O(\sqrt{\min(|S|,|N|)})} \times p(n)$. (Here $p(n)$ is a polynomial in n).

A description of our algorithm:

Unlike in simple stochastic games, the game can move to any state with specific transition probability that could depend on the action chosen by the player. The immediate

payoffs are quite general which would depend on the current state and the action chosen by the player who has more than one action available in that state. There may be no sinks. Our algorithm uses a certain lexicographic search in the policy improvement process. At each iteration one player's strategy is fixed and the other player's strategy changes just at one state. Two pure stationary strategies that differ just at one state are called *adjacent-strategies*. It is only because of this adjacent strategy property that we can compare the two vectors $\phi_\beta(f, g)$ with $\phi_\beta(f', g)$ and conclude that one weakly dominates the other in all coordinates.

The algorithm has only a peripheral similarity to Ludwig's algorithm. The key properties of lexicographic improvements are based on effectively going back and forth between the strategies of both players in various iterations. Even for simple stochastic games, this algorithm can be used for the recursive search of optimal strategies of subgames. We can renumber the states so that the first t_1 states are for player I and the next t_2 states are for player II and the last set of $s - t_1 - t_2$ states are for nature.

1. Starting with any arbitrary pair of pure stationary strategies f, g for the two players, find among the states $1, 2, \ldots, t_1$, some state a where for the first time, a change of action different from $f(a)$ just at state a results in an adjacent strategy f' with $\phi_\beta(f', g) \geq \phi_\beta(f, g)$ with strict inequality in some coordinate. If there are no such improvements possible go to step 3.
2. Set $f = f'$ and go to step 1.
3. Look for the first state b in $t_1+1, \ldots, t_1+t_2$ where an action different from $g(b)$ gives player II an adjacent strategy g' such that the discounted payoff $\phi_\beta(f, g') \leq \phi_\beta(f, g)$ with strict inequality at some state. If no such improvement for player II is possible go to step 5.
4. Set $g = g'$ and go to step 1.
5. The pair (f, g) is optimal for the two players.

The key property is that though the payoffs may be increasing some times and decreasing at other times, old pure stationary pairs are never revisited and hence, there is no cycling. Since there are finitely many pure stationary pairs, the algorithm terminates in finite steps locating an optimal pair.

The algorithm will be presented followed by some graph theoretic motivation. We exhibit a run of our algorithm on an example with 5 states and close with a conjecture concerning the afore mentioned graph structure of this class of games.

2. Perfect information games

For player I a *pure stationary strategy* is simply a function $f : S \to \cup_{i=1}^{s} A(i)$ with $f(t) \in A(t)$ for all t, that is, in state t player I always chooses the action $f(t)$. Similarly a function $g : S \to \cup_{i=1}^{s} B(i)$ with $g(t) \in B(t)$ for all t is a pure stationary strategy for player II. Shapley [1953] showed that under the discounted payoff criterion, perfect information stochastic games admit equilibria in pure stationary strategies, for both players. For a pair of pure stationary strategies (f, g) we write $\phi_\beta(f, g)$ to be the vector of expected discounted payoffs, resulting from f and g. Thus, $\phi_\beta(f, g)(s)$, the s-th coordinate of the vector $\phi_\beta(f, g)$ is the β- discounted expected payoff to player I from player II when the game starts at state s. We also write $Q(f, g)$ for the transition

matrix of the system, indexed by the state space, resulting from f and g. The ith row of $Q(f, g)$ will be written as $Q_i(f, g)$. For perfect information games, since the immediate transition probability at each state is determined by at most one player, we can always write either $Q_t(f, g) = Q_t(f)$ or $Q_t(f, g) = Q_t(g)$ depending on who controls state $t \in S$. In case it is a state of nature, the transitions are given a priori in such states t and we could even suppress the dependence of Q_t on f or g. Likewise we write $r(f, g)$ to be the vector indexed by the state space whose tth component is $r(t, f(t), g(t))$. Just like $Q(f, g)$, $r(f, g)$ can be broken into parts which depend on only one of f or g.

For the discounted MDP there is the *policy-improvement* algorithm of Howard [1960], which can be used to determine optimal policies. Starting at an arbitrary policy f_0 and moving along suitable adjacent policies, the algorithm produces a sequence of improvements $f_1, f_2, \ldots, f_k$ until an optimal policy is reached. For the sequence of policies, the corresponding values ϕ_β are strictly monotonic. Thus with only finitely many pure stationary policies, the algorithm terminates in finite steps. Extending the policy-improvement algorithm of MDP's to stochastic games was initially attempted by Pollatschek and Avi-Itzhak [1969]. Only with a stringent condition on the transitions and the discount factor (see O.J. Vrieze [1983], Van der Waal [1977]) they proved that their algorithm terminates for these games . For a general survey on algorithms for stochastic games see Raghavan and Filar [1991]. Kallenberg [1983] is another source for policy-improvement in MDPs.

3. Algorithm

We rearrange the states so that player I has more than one action and player II has exactly one action in states $1, \ldots, t_1$ and player I has exactly one action and player II has more than one action in states $t_1 + 1, \ldots, t_1 + t_2$. The rest of the states can as well be dubbed as states of nature. When a strategy of one player is fixed, we are in a discounted MDP and for them it is enough to find the best pure stationary strategy among all pure stationary strategies. Thus, because of Shapley's existence theorem of optimal pure stationary strategies for our games, we only need to find a pure stationary pair (f^*, g^*) such that

$$\phi_\beta(f, g^*) \leq \phi_\beta(f^*, g^*) \leq \phi_\beta(f^*, g) \tag{3.4}$$

for all pure stationary strategies f and g. For a pair of pure stationary strategies (f, g) we write $[(f, g) = (f(1), g(1)), \ldots, (f(s), g(s)]$ where $(f(t), g(t))$ is the pair of actions chosen in state t under (f, g). For any state t at least one of $f(t)$ or $g(t)$ is 1. (The player is essentially a dummy for that state). An *adjacent improvement of type I* is a new pair of pure stationary strategies (h, g) where:

1. h and f differ in exactly one state, namely
   there exists $\bar{t}$, $1 \leq \bar{t} \leq t_1$, with $h(\bar{t}) \neq f(\bar{t})$ and $h(\tau) = f(\tau)$ for $\tau \neq \bar{t}$.
2. $\phi_\beta(h, g) \geq \phi_\beta(f, g)$ and $\phi_\beta(h, g)(t) > \phi_\beta(f, g)(t)$ for some $1 \leq t \leq s$.

The purpose of the second condition is clearly that player I is better off playing h than f against player II's g. The first condition is an adjacency condition required in our algorithm. It states that h differs from f in exactly one state. Of course we have the corresponding definition for *adjacent improvement of type II*, namely it is a pair (f, h) where:

$\exists t', t_1 + 1 \leq t' \leq t_1 + t_2$, with $g(t') \neq h(t')$ and $g(\tau) = h(\tau)$ for $\tau \neq t'$ such that $\phi_\beta(f, h) \leq \phi_\beta(f, g)$ and $\phi_\beta(f, h)(t) < \phi_\beta(f, g)(t)$ for some $1 \leq t \leq s$

Notice that in both cases we require a *strict* improvement in ϕ_β value in some state. A pair of pure stationary strategies (f', g') will be called an *improvement* of (f, g) if it is an adjacent improvement of either type I or type II. Note that in such a case we would have either $f' = f$ or $g' = g$ depending on the type of improvement.

Algorithm:

1. Choose arbitrarily a pair of pure stationary strategies (f^0, g^0) (e.g. $f^0(t) = g^0(t) = 1$ for $t = 1, \ldots, s$) and set $\alpha = 0$.
2. Search lexicographically for an improvement $(f^{\alpha+1}, g^{\alpha+1})$ of (f^α, g^α) *always looking first for player I* and then only for player II. There are three cases:
   Case 1: An improvement for player I is found. In this case let $\alpha = \alpha + 1$ and repeat step 2.
   Case 2: There are no improvements for player I, but there is an improvement for player II. In this case let $\alpha = \alpha + 1$ and repeat step 2.
   Case 3: There are no improvements . Go to step 3.
3. The pair $(f^*, g^*) = (f^\alpha, g^\alpha)$ is an optimal pure stationary strategy pair for the two players.

Remark 1. The claim that a lexicographically arrived locally optimal pair is optimal for the stochastic game. It depends on some intrinsic properties of stochastic games of perfect information.

In an ordinary matrix game $A = (a_{ij})$ with value v, if $a_{pq} = v$, it does not mean p, q are good pure strategies. However for the case of games with perfect information, we have the following

Lemma 1. *In a zero-sum, perfect information, stochastic game Γ, a pair of pure stationary strategies (f°, g°) are optimal if and only if $\phi_\beta(f^\circ, g^\circ) = \phi_\beta(\Gamma)$, the value of the stochastic game.*

Proof. Now suppose $\phi_\beta(f^\circ, g^\circ) = \phi_\beta(\Gamma)$ for some pair of pure stationary strategies (f°, g°). Let (f^*, g^*) be an optimal pair. When player II's strategy is fixed at g^*, we are in an MDP situation and f^* is optimal for the MDP with immediate payoffs $r(f, g^*)$ and transitions $Q(f, g^*)$. Thus for any pure stationary strategy h for player I we have from MDP arguments, the vector inequality,

$$r(h, g^*) + \beta Q(h, g^*)\phi_\beta(f^*, g^*) \leq \phi_\beta(f^*, g^*). \tag{3.5}$$

Now consider the above inequality on those coordinates $k \in S$ which correspond to states controlled by player I. In these states g^* is simply a dummy. Here, the t-th coordinate of the reward vector $r(h, g^*)(t)$, $1 \leq t \leq t_1$ is independent of g^* and we can substitute any g for player II. Thus we have $r(h, g^*)(t) = r(h, g^\circ)(t)$. Similarly for $1 \leq t \leq t_1$ the t-th row $Q(h, g^*)_t$ of $Q(h, g^*)$ is independent of g^* and we can

replace $Q(h, g^*)_t$ with $Q(h, g^\circ)_t$. Since we are given for all t and in particular for states t controlled by player I, $\phi_\beta(f^\circ, g^\circ)(t) = \phi_\beta(\Gamma)(t) = \phi_\beta(f^*, g^*)(t)$, we can conclude

$$r(h, g^\circ)(t) + \beta Q(h, g^\circ)_t \phi_\beta(f^\circ, g^\circ) \leq \phi_\beta(f^\circ, g^\circ)(t). \tag{3.6}$$

Let t be a state controlled by player II. Now for any pure stationary h for player I, the immediate reward $r(h, g^\circ)(t) = r(f^\circ, g^\circ)(t)$ (player I is just a dummy in that state t controlled by player II), the transition probability vector $Q(h, g^\circ)_t$ corresponding to the t-th row of $Q(h, g^\circ)$ is independent of h, namely $Q(h, g^\circ)_t = Q(f^\circ, g^\circ)_t$. Also we know that $\phi_\beta(f^\circ, g^\circ)$ satisfies the vector equation

$$\phi_\beta(f^\circ, g^\circ) = r(f^\circ, g^\circ) + \beta Q(f^\circ, g^\circ)\phi_\beta(f^\circ, g^\circ) \tag{3.7}$$

In particular for those states t controlled by player II,

$$\begin{aligned} \phi_\beta(f^\circ, g^\circ)(t) &= r(f^\circ, g^\circ)(t) + \beta Q(f^\circ, g^\circ)_t \phi_\beta(f^\circ, g^\circ) \\ &= r(h, g^\circ)(t) + \beta Q(h, g^\circ)_t \phi_\beta(f^\circ, g^\circ). \end{aligned} \tag{3.8}$$

Thus from (3.6) and (3.8) we have

$$r(h, g^\circ) + \beta Q(h, g^\circ)\phi_\beta(f^\circ, g^\circ) \leq \phi_\beta(f^\circ, g^\circ).$$

For all states controlled by player I or by player II. In states k of nature the immediate reward $r(f^\circ, g^\circ)(t)$ is independent of f°, g° and the same way with the transitions at states of nature. Thus the inequality holds for all states. The map: $L_{h,g^\circ} : x \to r(h, g^\circ) + \beta Q(h, g^\circ)\, x$ is a contraction and monotone map with the unique fixed point $\phi_\beta(h, g^\circ)$. The above inequality shows that for any arbitrary pure stationary h for player I, $\phi_\beta(h, g) \leq \phi_\beta(f^\circ, g^\circ)$. For any arbitrary pure stationary strategy u of player II, we can use a similar argument to establish the vector inequality

$$r(f^\circ, u) + \beta Q(f^\circ, u)\phi_\beta(f^\circ, g^\circ) \geq \phi_\beta(f^\circ, g^\circ). \tag{3.9}$$

Thus (f°, g°) is optimal for the stochastic game. □

For what follows we require some notation. Let $t \in S$ be a fixed state. For any $X \subset A(t)$ we write Γ^t_X be the subgame in which only the actions in X are allowed in state t. The corresponding pure stationary strategy sets will be denoted by F^t_X and G^t_X. For the original game Γ we write F and G for the pure stationary strategy sets of players I and II respectively.

When one of the players, say player I restricts his actions in state t to only those in the set X, while player II has no restrictions at all in any state, we reach the subgame Γ^t_X. The pure stationary strategy space G^t_X for player II for this subgame is the same as G in the original game as player II's strategy is not constrained.

Lemma 2. *For $t \in S$, $X \subset A(t)$, $Y \subset A(t)$, $X \cap Y = \emptyset$ we have for each starting state k, $\phi(\Gamma^t_{X\cup Y})(k) = \max\{\phi_\beta(\Gamma^t_X)(k), \phi_\beta(\Gamma^t_Y)(k)\}$. In fact as vectors either $\phi_\beta(\Gamma^t_X) \geq \phi_\beta(\Gamma^t_Y)$ or $\phi_\beta(\Gamma^t_X) \leq \phi_\beta(\Gamma^t_Y)$.*

Proof. The restrictions of player I in state t have no relevance to player II as he is a dummy (has just one action) in that state even otherwise. Thus $G^t_X = G^t_Y = G^t_{X\cup Y} = G$. Also the set $F^t_X \cup F^t_Y = F^t_{X\cup Y}$.

An optimal f^* for player I in the game $\Gamma^t_{X\cup Y}$ is found either in F^t_X or in F^t_Y. Suppose it is found in F^t_X. Thus, player I can guarantee for each starting state k in Γ^t_X what he could guarantee in $\Gamma^t_{X\cup Y}$ against any pure stationary strategy g of player II and conversely. Also, any optimal g^* for player II for the game $\Gamma^t_{X\cup Y}$ is found in Γ^t_X. Thus the two games have the same value. Trivially $\phi_\beta(\Gamma^t_{X\cup Y}) \geq \phi_\beta(\Gamma^t_Y)$. Thus as vectors $\phi_\beta(\Gamma^t_X) \geq \phi_\beta(\Gamma^t_Y)$. In case f^* lies in Γ^t_Y we can repeat the proof verbatim and conclude $\phi_\beta(\Gamma^t_Y) = \phi_\beta(\Gamma^t_{X\cup Y}) \geq \phi_\beta(\Gamma^t_X)$. Hence the last part of the assertion follows.

An obvious player II analog of Lemma 2 exists using $B(t)$ instead of $A(t)$. □

Theorem 1. *The strategy pairs* $(f^\alpha, g^\alpha), \alpha = 0, 1, \ldots$ *obtained at step 2 along the algorithmic path never cycle and hence the algorithm must terminate. The terminal pair* (f^*, g^*) *is locally optimal, in the sense that no adjacent improvement is possible for either player. It is also a globally optimal strategy pair for the stochastic game.*

Remark 2. Unlike in the policy improvement algorithm of MDP, in our case we cannot expect any monotonicity property of the payoffs along the algorithmic path.

Proof. The proof is by induction on the total number n of actions available for the two players in all states, that is $n = \sum_{i=1}^{s}(a_i + b_i)$. Trivially $n \geq 2$. If $n = 2$ the algorithm obviously terminates at (f_0, g_0). By induction we will assume that the algorithm terminates at a saddle point policy (pure stationary optimal pair) whenever $n = 2, \ldots, k$. Let $n = k + 1$. If $a_i = b_i = 1$ for all i then again there is nothing to prove. So without loss of generality suppose there is a state where one of the players, has more than one action. Let τ be the largest value of t for which in state τ say player I, has more than one action. In state τ player II has exactly one action. We will prove the theorem for this case as the proof for the second case is almost identical.

We now split the game at state τ. The algorithm will pass through a sequence $c_1, c_2, \ldots, c_m$ of actions in state τ. The first such action is $c_1 = f_0(\tau)$ whereas $c_2, \ldots, c_m$ will be written later on. Let $X_i \subset A(t)$ be defined by $X_i = \{c_1, c_2, \ldots, c_i\}$. Now suppose the algorithm is initiated at (f_0, g_0). By construction the algorithm will reach a policy (f_{n_1}, g_{n_1}) which has no improvements in $\Gamma^\tau_{X_1}$. By our induction assumption (f_{n_1}, g_{n_1}) is an optimal pair for $\Gamma^\tau_{X_1}$.

By Lemma 2, $\phi_\beta(\Gamma^\tau_{X_1})$, the value of the game $\Gamma^\tau_{X_1}$, is at most $\phi_\beta(\Gamma)$, the value of the original game. If the two value vectors coincide, by Lemma 1, (f_{n_1}, g_{n_1}) is also optimal for the original game. Otherwise, $\phi_\beta(\Gamma^\tau_{X_1})(t) < \phi_\beta(\Gamma)(t)$ for some state t. Now consider the MDP induced on the original game, when player II restricts to g_{n_1}. One can use policy improvement algorithm on this MDP by starting from f_{n_1} and making *strict* improvements in some state only via adjacent policies. If no adjacent policy of f_{n_1} gives strict improvement in any state, then f_{n_1} is itself optimal for the MDP which means $\phi_\beta(\Gamma^\tau_{X_1})(t) = \phi_\beta(\Gamma)(t)$ for all states t. However, any adjacent policy of f_{n_1} which changes the action at a state other than state τ is a policy available for player I in $\Gamma^\tau_{X_1}$ and is not better than f_{n_1}. Thus the only way any strict improvement occurs via some adjacent policy of f_{n_1} has to be one which changes the action differently at state τ. Let $c_2 = f_{n_1+1}(\tau) \neq c_1$, be such an action chosen at state τ.

After this improvement, the algorithm continues and by the induction hypotheses we will reach a saddle point (f_{n_2}, g_{n_2}) of the subgame $\Gamma^{\tau}_{\{c_2\}}$. Applying Lemma 2 with $X = \{c_1\}$ and $Y = \{c_2\}$ we get $\phi_\beta(f_{n_2}, g_{n_2}) = \phi_\beta(\Gamma^{\tau}_{X_2})$. By Lemma 1 we can conclude that there are no improvements of (f_{n_2}, g_{n_2}) in $\Gamma^{\tau}_{X_2}$, and further by Lemma 2 that $\phi_\beta(f_{n_1}, g_{n_1}) \leq \phi_\beta(f_{n_2}, g_{n_2})$ with strict inequality in some coordinate. If there are no improvements in Γ as well then $(f^*, g^*) = (f_{n_2}, g_{n_2})$ is a saddle point of Γ and the algorithm terminates. Otherwise an improvement in state τ to an action outside X_2 is available. Just as before, let c_3 be this action. Because there is only a finite number of actions in state τ (as in all states), by repeating the same arguments as before, the algorithm will pass through saddle points $(f_{n_1}, g_{n_1}), (f_{n_2}, g_{n_2}), \ldots, (f_{n_m}, g_{n_m})$ of the respective games $\Gamma^{\tau}_{X_1}, \Gamma^{\tau}_{X_2}, \ldots, \Gamma^{\tau}_{X_m}$, and (f_{n_m}, g_{n_m}) a pair with no improvements. That is, $(f^*, g^*) = (f_{n_m}, g_{n_m})$ is a saddle point of Γ.

Since improvements in state j travel strictly through the sequence $c_1, c_2, c_3, \ldots$, by our induction we have no cycling (repetition) in the sequence. To prove the theorem in the case of player II having more than one action in state j is no more than using the player II-analog of Lemma 2. □

4. Hypercubes

In the policy improvement algorithm for the discounted MDP, the key ideas are

- If two policies f, f' are adjacent (differ in the choice of action at just one state) then the discounted payoffs $\phi_\beta(f)$, $\phi_\beta(f')$ using them are *totally ordered* as vectors, namely $\phi_\beta(f) \geq \phi_\beta(f')$ or $\phi_\beta(f') \geq \phi_\beta(f)$.
- If a policy has no strict improvement with any of its adjacent policies then the policy we started with is optimal.
- Given any two optimal policies f and f^*, any path of adjacent policies from one to the other consist of only optimal policies.
- Given any policy f and an optimal policy f^*, any path of adjacent policies $f^0 = f, f^1, \ldots, f^k = f^*$ consist of initially strict improvements followed by no improvements (weak improvements).

In the case of a stochastic game of perfect information we have pairs of policies (f, g) (pure stationary strategies), one for each player. If two pairs of policies (f, g), and (f', g') differ at exactly one state, we call them adjacent. This means either $f = f'$ and g differs from g' in some state or $g = g'$ and f differs from f' in some state. If two policy pairs (f, g) , (f', g') are adjacent (differ in the choice of action at just one state) then the discounted payoffs $\phi_\beta(f,\ g)$ and $\phi_\beta(f',\ g')$ are *totally ordered* as vectors, namely $\phi_\beta(f, g) \geq \phi_\beta(f', g')$ or $\phi_\beta(f', g') \geq \phi_\beta(f, g)$. Thus this key property is common to both MDP and stochastic games of perfect information. The closest to an MDP are stochastic games of perfect information where exactly one state, is controlled by player II and the rest are controlled by player I. By renumbering states, we can assume this to be the last state. If we replace improvements by *lexicographic improvements*, via adjacent policies, we can (not obvious) lift the second property above of MDP also to such stochastic games of perfect information.

Motivated by these we will study the directed graph for a class of hypercubes which will eventually capture some intrinsic combinatorial properties of perfect information

stochastic games. A *pair* of pure stationary strategies for the two players (f, g) is essentially a function $\theta : S \to \cup_{t=1}^{s}(A(t) \cup B(t))$ with $\theta(i) \in A(i)$ for those states i belonging to player I, and $\theta(j) \in B(j)$ for those states j belonging to player II. In other words, θ just chooses an action in each state. In MDP terminology, θ is just a policy (for the single player). In defining our hypercubes, the idea is to associate with each pair (f, g) of strategies of Γ with a vertex of a graph and join each adjacent pair of strategy pairs with a directed edge. In the next section we connect the two.

Let s be a fixed positive integer. We define a class of digraphs Ω_s as follows. Let $a_1, a_2, \ldots, a_s$ be integers with $a_i \geq 2$ for all i and set $A(i) = \{1, 2, \ldots, a_i\}$. Consider a digraph C with the properties (P1), (P2), and (P3) that follow.

(P1): The vertex set of C is the set $\prod_{i=1}^{s} A(i)$.

Two vertices $\alpha, \theta \in \prod_{i=1}^{s} A(i)$ are called *neighbors* if and only if they differ in exactly one coordinate.

(P2): For any pair of neighboring vertices α and θ, at least one of (α, θ) or (θ, α) is an edge. We denote the directed edge (α, θ) by $\alpha \to \theta$.

(P3): If (α, θ) is an edge of C then α and θ are neighbors.

The number s will be called the dimension of C. The set of all such digraphs C (which result from some $s, a_1, a_2, \ldots, a_s$) will be denoted by Ω_s, and we define $\Omega = \cup_{s=1}^{\infty}\Omega_s$. From here on we write $\alpha \in C$ to mean α is a vertex of C. If (α, θ) and (θ, α) are both edges of C then we write $\alpha \leftrightarrow \theta$. Temporarily we will assume that $a_i = 2$ for all i and refer to this restricted class of digraphs by Ω' $(= \cup_{s=0}^{\infty}\Omega'_s)$. In this case, C can be identified with the well-known s-dimensional hypercube. Given any fixed vertex $\alpha \in C$, there are exactly s vertices adjacent to α. Let them be $\theta_1, \theta_2, \ldots, \theta_s$. Without loss of generality, assume that α and θ_i differ in the ith coordinate. If (α, θ_i) is an edge then we say that α is a *max in dimension* i. If (θ_i, α) is an edge then we say that α is a *min in dimension* i. Note that $\alpha \leftrightarrow \theta_i$ implies that α is both a max and a min in dimension i.

Any vertex α is a max in some (may be empty) dimensions and a min in rest of the dimensions. All together there are 2^s such possible max/min configurations for α. If $\alpha \leftrightarrow \theta$ for some vertex θ then α will have multiple configurations. Each such θ will double the number of configurations α has.

Let $S = \{1, 2, \ldots, s\}$ and let T be an arbitrary subset of S. For any $b \in \prod_{i \in T} A(i)$, we can define the subgraph $C_{b,T}$ of C by restricting all coordinates in T to b. If $|T| = k$ then $C_{b,T}$ can be identified with an $(s-k)$-dimensional hypercube in its own right, and thus we have $C_{b,T} \in \Omega'_{s-k}$. $C_{b,T}$ will be called a *subcube* of C. Furthermore, given any two vertices $\alpha, \theta \in C$, there is a unique subcube of smallest dimension which contains them. We call this subcube $C(\alpha, \theta)$. A single max/min configuration for a vertex α in a t-dimensional subcube D can be thought of as a binary t-tuple $([\alpha]_{i_1}, \ldots, [\alpha]_{i_t})$ where $[\alpha]_{i_q}$ equaling zero (one) means that in this configuration, α is a min(max) in dimension q of D (here we need to write $i_1, \ldots, i_t$ to denote the t non-constant coordinates on which D is defined.) We say α and θ *share a max/min configuration* in a subcube D to mean that $([\alpha]_{i_1}, \ldots, [\alpha]_{i_t})$ and $([\theta]_{i_1}, \ldots, [\theta]_{i_t})$ are max/min configurations of α and θ respectively in the sub cube D, and they are equal as binary vectors.

A digraph $C \in \Omega'_s$ will be called *complete* if all 2^s max/min configurations are present in C. C will be called *balanced* if the following properties hold:

(B1): If two vertices $\alpha, \theta \in C$ have a max/min configuration in common, then for every pair of neighboring vertices $h, k \in C(\alpha, \theta)$ we have $h \leftrightarrow k$.

(B2): If $\alpha \leftrightarrow \theta$ for some pair of neighboring vertices $\alpha, \theta \in C$ then α and θ have the same max-min configurations, i.e. for all i, α is a max(min) in dimension i if and only if θ is a max(min) in dimension i.

A digraph C will be said to be *full* if the following two conditions hold:

(F1) The digraph C and all its sub cubes are complete.

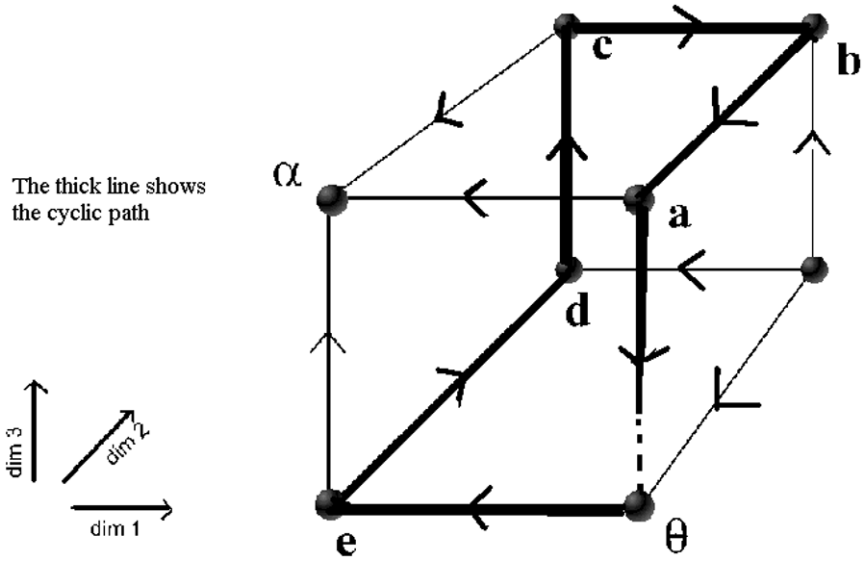
(F2) The digraph C and all its sub cubes are balanced.

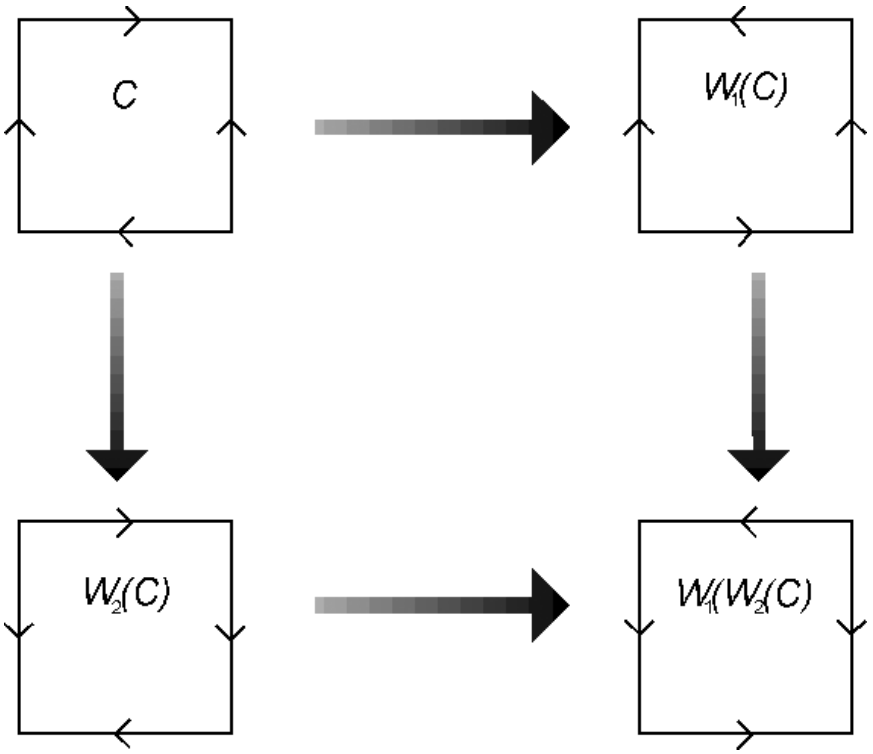
The set of full s-dimensional elements of Ω'_s will be denoted by Λ'_s and we write $\Lambda' = \cup_{s=1}^{\infty} \Lambda'_s$. Next we define the maps $W_i : \Omega'_s \to \Omega'_s$ for $i = 1, 2, \ldots, s$. For $C \in \Omega'_s$ we define $W_i(C) \in \Omega'_s$ to have the same vertex set as C. The edges of $W_i(C)$ are the same as those in C except that those edges whose vertices differ in the ith coordinate are reversed, i.e. for all neighboring pairs $\alpha, \theta \in C$ with α and θ differing in the ith coordinate, (α, θ) is an edge of $W_i(C)$ if and only if (θ, α) is an edge of C (See **Figure 2**). It follows directly from the definition that $W_i(W_i(C)) = C$ and that $W_i(W_j(C)) = W_j(W_i(C))$ for all i, j.

The vertex α in **Figure 1** is a min in all three dimensions, whereas θ is a max in dimension 1 and a min in dimensions 2 and 3. The other six vertices possess the rest of the max/min configurations.

Lemma 3. $W_i(\Lambda'_s) = (\Lambda'_s)$

Proof. Let $C \in \Lambda'_s$. If $s = 1$ then C has exactly two vertices, call them α and θ. If (α, θ) is an edge then (θ, α) is an edge in $W_1(C)$ so that θ is a max and α is a min in dimension 1. Thus all of the $2^s = 2^1$ configurations are present, and therefore $W_1(C)$ is complete. If α and θ share some configuration then they must both be max's and min's in dimension 1.





This just means that (α, θ) as well as (θ, α) are edges in C and by definition are also edges of $W_1(C)$. Thus $\alpha \leftrightarrow \theta$ in $W_1(C)$ as well so that (F2) is also satisfied.

Suppose the lemma holds for $C \in \Lambda'_s$, $s = 1, 2, \ldots, n-1$. If $s = n$ then we proceed as follows. Fix $T = \{i\}$ and write $D = W_i(C)$. Then $D_{0,T}$ and $D_{1,T}$ are the sub cubes of D which correspond to fixing the ith coordinate of all vertices to 0 and 1 respectively. By the induction hypothesis we have that W_i leaves all proper sub cubes complete, in particular, $D_{0,T}$ and $D_{1,T}$. If α and θ, viewed as vertices in C, have the same max-min configurations then viewed as vertices in D they will as well. Also, W_i leaves double edges unaltered so that D satisfies (F2). Thus, we only need check that D itself is complete.

Let α be any vertex in C. We will show that the max/min configuration of α is present in D. Suppose $\alpha \in C$ is a max in dimension i. As C is complete, there must be a vertex $\theta \in C$ which has the same max/min configuration as α except that it is a min in dimension i. Then θ viewed as a vertex in D will have the same max/min configuration in all dimensions as in C except that in dimension i it will be a max. Thus, $\theta \in D$ will have the same max/min configuration as $\alpha \in C$. Similar is the case for $\alpha \in C$ being a min in dimension i. As α was arbitrary, all max/min configurations in C appear in D. Therefore the completeness of C implies that of D. This proves that $D \in \Lambda'_s$ whence $W_i(\Lambda') \subset \Lambda'$. Applying W_i to this last containment yields $\Lambda'_s = W_i(W_i(\Lambda'_s)) \subset W_i(\Lambda'_s) \subset \Lambda'_s$ so that $W_i(\Lambda'_s) = \Lambda'_s$. This completes the proof. □

Lemma 4. *Let $C \in \Lambda'_s$ for some s and let $\alpha \in C$ be any vertex. Then there exists a sequence of vertices $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_k = \alpha^*$ where (α_{i+1}, α_i), $i = 0, 1, \ldots, k-1$, are edges in C and α^* is a max in dimension i for all i.*

Proof. If $s = 1$ then C has exactly two distinct vertices, and it is trivial to check that the lemma holds. Suppose the lemma holds for $s = 1, 2, \ldots, n-1$. If $s = n$ then we proceed as follows. Let $T = \{n\}$ and let α be any vertex in C. Without loss of generality, assume that $\alpha \in C_{0,T}$. By the induction hypothesis there exists a sequence $\alpha = \alpha_0, \alpha_1, \ldots, \alpha_j = \theta$ of vertices in $C_{0,T}$ such that (α_{i+1}, α_i), $i = 0, 1, \ldots, j-1$ are edges in $C_{0,T}$ and θ is a max in dimensions i for $i = 1, 2, \ldots, s-1$. If θ is also a max in dimension s then setting $k = j$ and $\alpha^* = \theta$ yields the desired sequence. So assume that θ is not a max in dimension s whence there must be a vertex h in $C_{1,T}$ such that (h, θ) is an edge in C. Set $\alpha_{j+1} = h$. Again by the induction hypothesis there exists a sequence of adjacent vertices $\alpha_{j+1}, \alpha_{j+2}, \ldots, \alpha_k$ in $C_{1,T}$, where $(\alpha_{j+l+1}, \alpha_{j+l})$ are edges of $C_{1,T}$, where α_k is a max in dimension i for $i = 1, 2, \ldots, s-1$. We claim that α_k must be a max in dimension s as well. If α_k were a min in dimension s then it would have the same max/min configuration as θ. As C is full, this would imply that for all neighbors $u, v \in C(\theta, \alpha_k)$ we have $u \leftrightarrow v$. As θ differs from both h and α_k in the sth coordinate and as $\alpha_k \in C(\theta, \alpha_k)$, we must have that $\theta \leftrightarrow h$. But this contradicts our assumption that θ is not a max in dimension s. Setting $\alpha^* = \alpha_k$ yields the sequence required in the lemma. This completes the proof. □

Any sequence $\alpha_0, \alpha_1, \ldots, \alpha_k$ where (α_{i+1}, α_i), $i = 0, 1, \ldots, k-1$ are edges in C will be called an *increasing sequence*. An increasing sequence $\alpha_0, \alpha_1, \ldots, \alpha_k$ in C is called *strictly increasing* if (α_i, α_{i+1}) is not an edge of C for $i = 1, \ldots, k-1$. A vertex which is max in dimension i for all i will be called a *max-vertex*. These definitions will also be used in the general class Ω.

Corollary 1. *Let $C \in \Lambda'_s$ for some s. Let $\alpha \in C$ be any vertex and let $\alpha^* \in C$ be any max-vertex. Then there exists an increasing sequence in C starting at α and terminating at α^*.*

Proof. By Lemma 4 there exists an increasing sequence starting at α and terminating at some max-vertex θ. As α^* is also a max-vertex, it has the same max/min configuration as θ. Therefore, F2) guarantees an increasing sequence from θ to α^*. Concatenating the latter sequence with the former gives an increasing sequence from α to α^*. This completes the proof. □

Lemma 5. *Let $C \in \Lambda'_s$ for some s. Let $\alpha \in C$ be any vertex. Then there exists a strictly increasing sequence in C starting at α and terminating at some max-vertex.*

Proof. The following inductive proof is almost a repeat of Lemma 4. For $s = 1$, this is a trivial assertion. We will assume the assertion for $s = 1, \ldots, n-1$ and prove for n. Given a vertex α, we can fix $T = \{n\}$ and consider without loss of generality that $\alpha \in C_{0,T}$. By induction we have a strict improvement via adjacent vertices from α to some max vertex $\alpha_j = \theta$ of $C_{0,T}$. If θ is also a max vertex in dimension s we are done with $\alpha^* = \theta$. Otherwise, starting with a vertex $h = \alpha_{j+1}$ which is max in dimension n and adjacent to θ there must be a path of strict improvements along adjacent vertices

$h = \alpha_{j+1}, \ldots, \alpha_k$ in $C_{1,T}$ terminating in a max vertex of $C_{1,T}$. If $\alpha_k = \lambda$ is not a max vertex in all dimensions, the only possibility is that it is a min vertex in dimension s. If α_k were a min in dimension s then it would have the same max/min configuration as θ. As C is full, this would imply that for all $u, v \in C(\theta, \alpha_k)$ we have $u \leftrightarrow v$. As θ differs from both h and α_k in the sth coordinate and as $\alpha_k \in C(\theta, \alpha_k)$, we must have that $\theta \leftrightarrow h$. But this contradicts our assumption that θ is not a max in dimension s. Setting $\alpha^* = \alpha_k$ yields the sequence required in the lemma. This completes the proof. □

We next return to the original class of digraphs Ω and refer to the elements of Ω' as *2-cubes*. Let $C \in \Omega_s$ for some $s \geq 1$. If we restrict the sets $A(i), i = 1, \ldots, s$, corresponding to the digraph C, to some (non-empty) subsets $A(i)' \subset A(i)$, we get a subgraph C' with vertex set $\prod_{i=1}^{s} A(i)'$. Dropping those $A(i)'$ which contain only one element yields a subgraph which can be considered an element of Ω in its own right. Just as before, we will call C' a *sub cube* of C. If we have $|A(i)'| \leq 2$ for all i then C' will be a 2-cube. If exactly k of the sets $A(i)'$ have only one element, then $C' \in \Omega_{s-k}$. Furthermore, given any two vertices $\alpha, \theta \in C$, there is a unique smallest sub cube, again denoted by $C(\alpha, \theta)$, containing both α and θ (i.e. no proper sub cube of $C(\alpha, \theta)$ contains both α and θ). It is easy to see that $C(\alpha, \theta)$ must be a 2-cube.

Let $C \in \Omega_s$ and let $\alpha \in C$ be an arbitrary vertex. Consider all s-dimensional 2-cubes which are sub cubes of C and which contain the vertex α. If α is a max in dimension i in all these 2-cubes then we say that $\alpha \in C$ is a *max in dimension i*. Identical is the definition of $\alpha \in C$ being a *min in dimension i*. Unlike the case of the 2-cubes, the vertex α can be a max, min, both or neither in dimension i. If the vertex α is either a min or max in every dimension then we can speak of α having a max/min configuration. If all 2^s possible max/min configurations are present in C then C is said to be *complete*. The definitions of *balanced* and *full* for 2-cubes can be taken verbatim for $C \in \Omega$. The set of full s-dimensional elements of Ω will be denoted by Λ_s and we write $\Lambda = \cup_{s=1}^{\infty} \Lambda_s$. We can apply the definition of the maps $W_i, i = 1, \ldots, s$ to the general set Ω_s, and again we get

Lemma 6. $W_i(\Lambda_s) = \Lambda_s$

Proof. The proof is similar to that of Lemma 3 except that the induction is on $n = \sum_{i=1}^{s} a_i$. It mainly requires noticing the fact that if a vertex α is a min(max) in dimension i, then W_i changes it to a max(min) in dimension i. □

Lemma 7. *Let $C \in \Lambda_s$ for some s. Let $\alpha \in C$ be any vertex and let $\alpha^* \in C$ be any max-vertex. Then there exists an increasing sequence in C starting at α and terminating at α^*.*

Proof. Given α and α^* consider the sub cube $D = C(\alpha, \alpha^*)$. Since D is a 2-cube we can apply Corollary 1 to prove the existence of the desired sequence. This completes the proof. □

Lemma 8. *For any $C \in \Lambda$ and any vertex $\alpha \in C$, there exists a strictly increasing sequence starting at α and terminating at some max-vertex $\alpha^* \in C$.*

Proof. Let $C \in \Lambda$ and $\theta \in C$ be any max-vertex in C. Given any vertex $\alpha \in C$, consider the 2-cube $C(\alpha, \theta)$. We know from Lemma 5 that in this sub cube there is a strictly

increasing sequence from α to some max vertex $\theta' \in C(\alpha, \theta)$. As θ is also a max-vertex of $C(\alpha, \theta)$, by property (B1) of $C(\alpha, \theta)$ we must have a sequence $\theta' = \theta_0, \theta_1, \ldots, \theta_k = \theta$ in $C(\alpha, \theta)$ with $\theta_i \leftrightarrow \theta_{i+1}$ for $i = 0, \ldots, k-1$. By property (B2) of C, θ' and θ have the same max-min configurations (as well as the intermediate θ_i). As θ is a max-vertex of C, so is θ'. This completes the proof. □

5. The clear connection

Given a stochastic game Γ of perfect information, we define a digraph $C_\Gamma \in \Omega$ as follows. Let $S = \{1, 2, \ldots, s\}$ be the state space of Γ where we leave out those states which have only one action for the player of that state. As before let a_i be the number of actions in state i. We define a digraph $C^0_\Gamma \in \Omega_s$ just as before using the data $s, a_1, \ldots, a_s$. Each vertex of C^0_Γ corresponds to a pair of pure stationary strategies for the two players. If α is a vertex, we write (f_α, g_α) for the corresponding pair of pure stationary strategies. A pair of vertices α and θ will comprise an edge (α, θ) of C^0_Γ if and only if (f_α, g_α) and (f_θ, g_θ) differ in exactly one state and $\phi_\beta(f_\alpha, g_\alpha) \geq \phi_\beta(f_\theta, g_\theta)$. The following is the combinatorial connection promised.

Theorem 2. *The digraph* $C^0_\Gamma \in \Lambda_s$.

Proof. The following observations are direct consequences of Blackwell [1962]. For any two adjacent vertices α and β, the respective payoff vectors $\phi_\beta(f_\alpha, g_\alpha)$ and $\phi_\beta(f_\theta, g_\theta)$ are comparable so that C^0_Γ satisfies properties P1, P2, and P3. The subgames of Γ correspond to sub cubes of C^0_Γ. The key property that the digraph C^0_Γ is complete follows from the existence of optimal pure stationary strategies for our games. Suppose we want a vertex λ which has the predefined configuration, say max, max, min, min, ... , max, then all we have to do is to consider a new game with perfect information with the same immediate payoffs and same transitions, except that we reallocate the states to player I or player II depending on at which states we need a max and in which states we need a min. For example if total number of states is say 5 and say (max, max, min, min, max) is the vertex we are looking for, we just assign states 1, 2, 5 to player I and 3, 4 to player II. The set of pure stationary pairs available for the new game is the same as the old game and the optimal pure stationary pair (f^*, g^*) of this new game will correspond to a vertex of C^0_Γ with the specified max/min configurations. In Lemma 1 we showed that any strategy pair for perfect information games giving a payoff equal to the value vector are themselves optimal. If two vertices have the same max/min configuration, then we know the strategy pairs corresponding to the two vertices are optimal for the game defined according to the max/min configuration. Let (f_α, g_α) and (f_θ, g_θ) be the two optimal pairs for the game. They have the same max/min configurations. Let $\pi \in C(\alpha, \theta)$. Let (f_τ, g_τ) be the strategy pair corresponding to π. As an element of the smallest two dimensional cube containing the vertices α and θ the strategy (f_τ, g_τ) for each state t of player I prescribes either the action prescribed by f_α or the action prescribed by f_θ . From the policy improvement step suppose we take state 1 to be a state for, say player I and f_τ prescribes the action $f_\theta(1)$. We already have $\phi_\beta(f_\tau, g_\tau)$ as the unique fixed point of the contraction operator

$$u \to L_\tau u = r(f_\tau, g_\tau) + \beta Q(f_\tau, g_\tau)u \tag{5.10}$$

We will show that the value vector $\phi_\beta(f_\theta, g_\theta) = \phi_\beta(f_\alpha, g_\alpha) = v$ is also a fixed point of the same operator. Observe that

$$r(f_\tau, g_\tau)(1) = r(f_\theta, g_\tau)(1) = r(f_\theta, g_\theta)(1)$$

Similarly the first row vector of the transitions Q satisfies

$$Q(f_\tau, g_\tau)_1 = Q(f_\theta, g_\tau)_1 = Q(f_\theta, g_\theta)_1$$

Thus

$$r(f_\tau, g_\tau)(1) + \beta Q(f_\tau, g_\tau)_1 v = r(f_\theta, g_\theta)(1) + \beta Q(f_\theta, g_\theta)_1 v = v(1)$$

Similarly we can prove for any state t

$$r(f_\tau, g_\tau)(t) + \beta Q(f_\tau, g_\tau)_t v = v(t)$$

Thus v is another fixed point of the operator L_τ. The uniqueness of the fixed point of L_τ shows that $v = \phi_\beta(f_\tau, g_\tau)$. Thus for all s-dimensional 2-cubes which posses the same max/min configuration at two vertices α, θ, we have shown that for any other vertex $\tau \in C(\alpha, \theta)$, $\tau \leftrightarrow \alpha$ and the digraph C_Γ^0 satisfies condition (**B1**):

Suppose $\theta \leftrightarrow \alpha$ for some pair of neighboring vertices of the digraph C_Γ^0. This means the associated strategy pairs (f_α, g_α) and (f_θ, g_θ) are adjacent strategy pairs. The assumption that $\theta \leftrightarrow \alpha$ is equivalent to saying: if one pair is optimal for a perfect information game with a specific configuration of states as max and min states, then the other pair is also optimal for the same game and hence preserves the same max/min configuration as the first pair. Thus the digraph C_Γ^0 satisfies condition (**B2**). □

If i is a state of player I, then $b_i = 1$. If i is a state of player II, then $a_i = 1$. Taking this into account we next define a sequence of digraphs C_Γ^i, $i = 1, 2, \ldots, s$ recursively. Set

$$C_\Gamma^i = \begin{cases} W_i(C_\Gamma^{i-1}) & \text{if } a_i = 1 \\ C_\Gamma^{i-1} & \text{if } b_i = 1 \end{cases}$$

and define $C_\Gamma = C_\Gamma^s$. As $C_\Gamma^0 \in \Lambda_s$ we can conclude, by Lemma 3, that $C_\Gamma \in \Lambda_s$ as well. The application of the functions W_i on C_Γ^0 result in the max-vertex of C_Γ corresponding to the saddle point policy of Γ. Let α_0 be an arbitrary vertex corresponding to a fixed pair of pure-stationary policies for both players. By Lemma 7 we know that in C_Γ there exists an increasing sequence of vertices $\alpha_0, \alpha_1, \ldots, \alpha_k$ where α_k is a max-vertex whence $(f_{\alpha_k}, g_{\alpha_k})$ is a saddle point of Γ.

In the representation of the game Γ as a hypercube C_Γ, we ignored those states for which only one action was present. Including those states would have increased the dimension of C_Γ unnecessarily. After all, consider a game Γ with ten states which has two actions in state 1 and one action in the other nine states. This game, viewed as an MDP, has only two distinct policies so that its corresponding hypercube C_Γ would be two vertices connected by one or two edges. That is, C_Γ is a one-dimensional hypercube (and not a ten dimensional one). However, in computing the required sequence of policies, we cannot ignore these states.

To proceed directly from these results we would start with a policy and generate a sequence of improvements. The natural way to search for improvements is to do so lexicographically. This amounts to restricting the search to the smaller sub cubes of the graph C_Γ first and as the saddle points of the subgames are determined, the larger sub cubes would be entered. This is exactly what our algorithm does. From a computational aspect, computing $\phi_\beta(f, g)$ initially involves solving a linear system of equations. Once done, the later computations of $\phi_\beta(f', g')$ corresponding to adjacent pairs for each new improvement involves a single column pivot and a matrix multiplication.

6. An example

Next we present a run of the algorithm on a stochastic game with 5 states. States 1, 2 are for player I, the maximizer. States 3, 4 are for player II. State 5 is for nature. In states 1 to 2 player I has 3 available actions. In states 3, 4 player II has 3 actions. In all there are 81 pure stationary pairs . Each box represents a state. The first entry of the box gives the immediate reward for each action while the rest of the entries give the transition probabilities to states 1, ... 5. The discount factor for the game is set at $\beta = 0.999$.

	1	5.0	0.0	0.0	0.0	0.0	1.0 ←
State 1: Player I	2	4.0	0.0	0.0	0.2	0.0	0.8
	3	3.0	0.0	0.0	0.6	0.0	0.4
	1	6.0	0.0	0.0	0.0	0.0	0.1 ←
State 2: Player I	2	1.0	1.0	0.0	0.0	0.0	0.0
	3	0.0	0.0	0.0	0.1	0.0	0.0
	1	4.0	0.0	0.0	0.0	0.9	0.1
State 3: Player II	2	2.0	0.1	0.0	0.0	0.0	0.0
	2	0.0	0.3	0.0	0.2	0.5	0.0 ←
	1	2.0	0.0	0.1	0.6	0.3	0.0 ←
State 4: Player II	2	2.0	0.2	0.0	0.4	0.4	0.0
	3	3.0	0.0	0.0	0.0	0.9	0.1
State 5: Nature	1	0.0	0.0	0.10	0.20	0.3	0.4 ←

What follows is the sequence of pure stationary strategy pairs $(f_0, g_0), (f_1, g_1), \ldots$ generated by the algorithm along with their respective values $\phi_\beta(f_i, g_i)$. $\phi_\beta(f_i, g_i)$ is actually a vector with five coordinates since it is indexed by the starting state.

$(f_0, g_0) = (1, 1; 1, 1)$
$\phi_\beta(f_0, g_0) = (25623.75545, 25624.75545, 25626.37590, 25625.34070, 25621.31758)$
$(f_1, g_1) = (3, 1; 1, 1)$
$\phi_\beta(f_1, g_1) = (25624.79013, 25624.75545, 25626.37590, 25625.34070, 25621.31758)$
$(f_2, g_2) = (3, 1; 2, 1)$
$\phi_\beta(f_2, g_2) = (19259.87802, 19261.15600, 19259.95203, 19260.22973, 19257.08171)$
$(f_3, g_3) = (1, 1; 2, 1)$
$\phi_\beta(f_3, g_3) = (19771.24430, 19772.24430, 19771.26717, 19771.43943, 19768.22112)$
$(f_4, g_4) = (1, 1; 2, 2)$
$\phi_\beta(f_4, g_4) = (19601.15390, 19602.15390, 19601.19378, 19601.24696, 19598.11371)$
$(f_5, g_5) = (1, 1; 3, 2)$

$\phi_\beta(f_5, g_5) = (15060.09179, 15061.09179, 15057.74716, 15059.35223, 15056.59745)$
$(f_6, g_6) = (1, 1; 3, 1)$
$\phi_\beta(f_6, g_6) = (14128.31247, 14129.31247, 14125.83080, 14127.16724, 14124.72494)$

Here, computing $\phi_\beta(f, g)$ for each pair (f, g) amounts to solving a 5×5 linear equation. Since we are looking for improvements via adjacent policies, and since they are totally ordered, in our lexicographic search, if (f, g) and (f', g') differ in just the state τ, of player II say, then we can simply evaluate

$$r(g')(\tau)) + \beta \sum_t q(t/g'\tau)\phi_\beta(f', g')(t)$$

and look for strict improvement (in case of player I, larger than $\phi_\beta(f, g)(\tau)$ and in case of player II it must be a smaller than $\phi_\beta(f, g)(\tau)$). Only when such an improvement is found, we compute the discounted payoff vector at the new adjacent policy pair. This way our algorithm took just 6 steps in the above example to arrive at the optimal solution $(f_6, g_6) = (1, 1; 3, 1)$ with value (14128.31247, 14129.31247, 14125.83080, 14127.16724, 14124.72494). In all states the players choose the action which is myopically optimal. Invariably the myopic strategy is almost close to the optimal solution.

Remark. Suppose at each state, the immediate payoffs are the same for all actions of that state, then we have no way of taking advantage of the myopic optimal strategies.

7. An open question

It is only natural to try and extend this algorithm to the average reward payoff criterion

$$\phi(\pi, \rho)(t_0) = \lim_{N\to\infty} \frac{1}{N} \sum_{n=0}^{N} r_n(t_0, \pi, \rho).$$

This payoff was first introduced by Gillette [1957]. Liggett and Lippman [1969] showed that for perfect information games an optimal pair in pure stationary strategies exists under the average reward criterion so that one would expect there to be a policy-improvement approach to solving such a game. Using the average reward MDP policy improvement methods of Blackwell [1962] on many randomly generated stochastic games with perfect information showed finite termination every time, but the authors were unable to prove that cycling could never occur. For undiscounted games, even though the same vertex could be reached as part of a subgame, in all the examples we solved, it never did. Blackwell's policy improvement for undiscounted MDP's has the interesting property that each policy (in the sequence generated by policy-improvement) is better than the previous policy for all β close to 1. More precisely, Blackwell's algorithm to find average reward optimal policies produces a sequence $f_0, f_1, \ldots, f_k$ with $\phi(f_k) \geq \phi(f)$ for all f. In addition, for all β close to 1, $\phi_\beta(f_m)$ is strictly greater than $\phi_\beta(f_{m-1})$ in at least one coordinate. This is the key step in his proof that the procedure must terminate. Thus, if one were to prove the following conjecture of ours, the average reward policy

improvement path would never cycle and hence, would always find an average reward optimal pair:

Conjecture. Given a stochastic game Γ and a fixed $\beta \in [0, 1)$, the graph C_Γ contains no strict cycles, i.e. there does not exist a strictly increasing sequence $\alpha_0, \alpha_1, \ldots, \alpha_k$ with $\alpha_k = \alpha_0$.

For if a cycle occurred using Blackwell's average reward policy-improvement, then for all β close to 1 we would have a cycle in the induced graph of the game.

We conclude with a remark that the properties **F1** and **F2** are not sufficient to prove the conjecture. For example the 3-dimensional full cube in Figure 3 contains a strict cycle (Think of this as the induced graph of an MDP for cost minimization. Suppose we started with the vertex θ. This vertex is not optimal. Now moving along the edge in dimension 1, we reach the vertex e which fails to be a min vertex in both dim 2 and dim 3. In our search we could choose as the next strict improvement in dimension 2 vertex d. From d we pass through c, b, a improving systematically in dimensions, 3, 1, 2 and finally entering θ from a strictly improving in dimension 3 (See the darkly shaded cyclic path in Figure 1).

Complexity issues. In the case of simple stochastic games the expected number of operations required for the algorithm of Ludwig is subexponential. One can see some analogies between Ludwig's algorithm for simple stochastic games and Vrieze's algorithm (Vrieze [1983], Algorithm 6.3.2, page 95, Filar and Vrieze [1996], Algorithm 6.3.1, page 313) for discounted switching control games. Finite termination of Vrieze's algorithm depends on the monotonicity of the sum of values at all states for certain induced discounted single controller games. Since discounted single controller games are reducible to linear programming (Parthasarathy and Raghavan [1981]), the polynomial bound at each inductive step follows from Kachian's algorithm.

Our algorithm has no such monotonicity properties on any value function of any subgame. Even for the regular MDP case the policy iteration algorithm has exponential complexity (Condon [1994]). As we see, we use policy iteration on the states of player I each time we move to a new pure stationary strategy of player II in the lexicographic search. Allowing for moving through all pure stationary strategies of player II in our lexicographic search our algorithm is at worst exponential in time. We are unable to prove that our policy improvement algorithm will not cycle in the undiscounted case. On the other hand, our empirical study of the algorithm has shown quite favorable results. Namely, we randomly generated 10,000 perfect information stochastic games, each with at most 8 states. On the average, the number of iterations per game was no more than the number of states. To improve the results even further, we started at myopic optimal strategies (those with $\beta = 0$) and the algorithm terminated in at most 3 iterations. We also notice that as β tends to 1, the number of interactions increase.

The efficiency of our algorithm in practice is mostly due to the fact that often certain actions totally dominate other actions. This exponentially reduces the policy space in which the algorithm needs to search. For example, in a 4 state game with two actions in each state, the first action in state 1 might dominate all the other actions regardless of the actions in other states. Thus, once policy improvement chooses action 1 in state 1, it will never choose another action in that state. In the perfect information stochastic games which we have examined more closely, there have always been many dominating actions.

Acknowledgements. The authors would like to sincerely thank the anonymous referees for drawing our attention to papers on simple stochastic games studied extensively by computer scientists. We came to know of them for the *first time*. We felt happy that besides Game theorists, perfect information stochastic games have attraction also for computer scientists. Thanks to their thorough and extensive queries we were able to incorporate many of their detailed suggestions in the revised version.

References

Blackwell, D. [1962]: *Discrete Dynamic Programming.* Annals of Mathematical Statistics **33**, 719–726.
Condon, A. [1992]: *The Complexity of Stochastic Games.* Information and Computing **96**. 203–224.
Derman, C. [1970]: *Finite State Markovian Decision Processes.* Academic Press, New York.
Everett, H. [1957]: *Recursive Games*, In: Dresher, M., A.W. Tucker & P. Wolfe (eds.), pp 47–78. *Contribution to the theory of games, Vol. III*, Ann. of Math. Stud. 39, Princeton Univ. Press, Princeton.
Filar, J.A. and O.J. Vrieze [1996]: *Competitive Markov Decision Processes*, Springer Verlag, Berlin.
Gillette, D. [1957]: *Stochastic games with zero stop probabilities*, In: Dresher, M., A.W. Tucker & P. Wolfe (eds.), pp 179–188. *Contribution to the theory of games, Vol. III*, Ann. of Math. Stud. 39, Princeton Univ. Press, Princeton.
Gurwich, V.A. Karzanov, A.V. and L.G. Khachiyan [1988]: *Cyclic games and an algorithm to find minimax cycle means in directed graphs*, USSR Computational Mathematics and Mathe,matical Physics, **28**, 85–91.
Hoffman, A.J. and Karp, R.M. [1966]: *On Non-Terminating Stochastic Games.* Management Science, **12**, 359–370.
Hordijk, A. and Kallenberg, L.C.M. [1979]: *Linear Programming and Markovian Decision Chains.* Management Science **25**, 352–362.
Howard, R.A. [1960]: *Dynamic Programming and Markov Processes*, Wiley, New York.
Kallenberg, L.C.M. [1983]: *Linear Programming and Finite Markovian Control Problems.* Mathematical Centre Tract 148, Centre for Mathematics and Computer Science, Amsterdam.
Kachian L.G. [1979]: A polynomial algorithm in linear programming. *Soviet Math. Doklady*, **20**, 191–194.
Liggett, T.M. and S.A. Lippman [1969]: *Stochastic Games with Perfect Information and Time Average Payoff.* SIAM Review **11**, 604–607.
Ludwig, W. [1995]: *A subexponential Randomized Algorithm for the Simple Stochastic Game Problem.* Information and Computation **117**, 151–155.
Melekopoglou, M. and A. Condon [1994]: *On the Complexity of the Policy Improvement Algorithm for Markov Decision Processes.* ORSA Journal on Computing **6**, 188–192.
Miller, B and A. Veinott Jr. [1969]: *Discrete Dynamic Programming with a Small Interest Rate* Ann. Math. Statistics **40**, 366–370.
Parthasarathy, T. and T.E.S. Raghavan [1981]: An order field property for stochastic games when one player controls transition probabilities, *J. Optimization Theory and Appl.* **33**:375–392.
Pollatschek and Avi-Itzhak [1969]: *Algorithms for Stochastic Games with Geometrical Interpretation* Management Science **15**, 399–415.
Raghavan T.E.S. and J.A. Filar [1991]: *Algorithms for Stochastic Games – A Survey* Methods and Models of Operations Research ZOR **35**:437–472.
Shapley, L.S. [1953]: *Stochastic Games* Proceedings of the National Academy of Sciences U.S.A. **39**, 1095–1100.
Van der Waal, J. [1977]: *Discounted Markov Games: Successive Approximations and Stopping Times* International J. Game Theory **6**, 11–22.
Veinott, A. Jr. [1966]: *On Finding Optimal Policies in Discrete Dynamic Programming with no Discounting*, **37**, 1284–1294.
O.J. Vrieze [1983]: *Stochastic Games with Finite State and Action Spaces*, (Ph.D Thesis), Free University, Math. Centrum, Amsterdam.
Zwick, U and M.S. Paterson [1996]: *The complexity of mean payoff games on graphs.* Theoretical Computer Science, **158**, 343–359.