THE SIMPLE STRATEGY-ITERATION METHOD IS STRONGLY POLYNOMIAL FOR THE TURN-BASED DETERMINISTIC FORWARD GAME

Anonymous authors
Paper under double-blind review

ABSTRACT

We study Turn-Based Deterministic Forward Games (TBDFGs), the subclass of turn-based deterministic zero-sum games in which no directed cycle contains actions controlled by both players. This forward condition is strictly weaker than acyclicity: recurrent behavior may be arbitrarily rich within one player's states, while mixed-player feedback cycles are excluded. We analyze the simple strategy-iteration method of Hansen et al. (2013); Jia et al. (2020), a generic method for TBSGs whose execution neither tests for nor uses the TBDFG property. We prove that it has a strongly polynomial guarantee on every TBDFG. In particular, it terminates after at most $O(n^6 m^4 \log^4 n)$ simplex pivot steps for any uniform discount factor $\gamma \in (0, 1)$. Thus, the forward property acts as a structural certificate for convergence even when the algorithm is not informed that the input has this property.

1 INTRODUCTION

A Turn-Based Deterministic (Zero-Sum) Game (TBDG) is a special case of a Turn-Based Stochastic (Zero-Sum) Game (TBSG) in which the transition probability distribution associated with every action is deterministic (see Shapley (1953)). A TBSG extends a Markov decision problem (MDP) by partitioning the states between two players: one player minimizes a common linear objective, while the other maximizes it. Although finite MDPs are polynomial-time solvable through linear programming and its deterministic version is strongly polynomial time solvable Post & Ye (2015), the existence of strongly polynomial algorithms for broad classes of TBSGs and TBDGs remains open.

As in an MDP, at each time step the Markov process is in a state $s = 1, \ldots, n$, and the decision maker may choose an action a belonging to state s, denoted by $a \in \mathcal{A}_s$. Such a selection of one action per state is called a strategy π. After an action is chosen, the process moves to the next state and incurs an immediate cost c_a. The transition probabilities are determined by the chosen action a through a probability distribution vector $\mathbf{p}_a \geq \mathbf{0} \in \mathbb{R}^n$. If $\mathbf{p}_a$ is deterministic, or equivalently a point-mass distribution, for every a, then the TBSG reduces to a TBDG. In this case, the process moves along a directed graph whose nodes are states and whose directed edges are chosen actions. In the strategy graph, each action is either on a path, called a path action, or on a cycle, called a cycle action.

In a TBSG, the total set of n states and their actions are partitioned into two sets, $\mathcal{S}^1$ and $\mathcal{S}^2$. The player controlling $\mathcal{S}^1$ minimizes the common total discounted cost over the infinite horizon, while the player controlling $\mathcal{S}^2$ maximizes it. We refer to these players as Player 1 and Player 2, respectively, and let $|\mathcal{S}^1| = n_1$ and $|\mathcal{S}^2| = n_2$ with $n_1 + n_2 = n$. Furthermore, let k_s be the number of actions available in state s, and suppose the actions are sorted as

$$\mathcal{A}_s = \left\{ a \in \mathbb{N} : \sum_{j=0}^{s-1} k_j < a \leq \sum_{j=0}^{s} k_j \right\} \quad \forall s = 1, \ldots, n.$$

Here $\mathbb{N}$ is the set of natural numbers and $k_0 = 0$. We denote $\mathcal{A}^i = \cup_{s \in \mathcal{S}^i} \mathcal{A}_s$ for $i = 1, 2$. Then the total number of actions is $m = \sum_{s=1}^{n} k_i$, where Player 1 controls $m_1 = \sum_{s=1}^{n_1} k_s$ actions $\mathcal{A}^1$ and Player 2 controls $m_2 = m - m_1$ actions in $\mathcal{A}^2$.

We now give the formal definition of TBDGs.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Definition 1 (**TBDG).** *A* 2-Player Deterministic Zero-Sum Game *is* defined by a tuple $\mathcal{G} = (\mathcal{S}, \mathcal{A}, \mathbf{c}, \mathbf{P}, \gamma)$. *The elements of the tuple are defined as follows.*

1. $\mathcal{S} = \mathcal{S}^1 \cup \mathcal{S}^2 = [n]$ *denotes the state space, where* $\mathcal{S}^1$ *and* $\mathcal{S}^2$ *are sets of states controlled by Players 1 and 2, respectively, with* $\mathcal{S}^1 \cap \mathcal{S}^2 = \phi$.
2. $\mathcal{A} = \cup_{s\in\mathcal{S}}\mathcal{A}_s = [m]$ *denotes the action space, where* $\mathcal{A}_s$ *is the set of actions playable at state* s. *We assume* $\mathcal{A}_s \cap \mathcal{A}_{s'} = \phi$ *for all* $s, s' \in \mathcal{S}$ *and* $s \neq s'$, *and denote* $\mathcal{A}^i = \cup_{s\in\mathcal{S}^i}\mathcal{A}_s$ *as the set of actions available to Player* i, *for* $i = 1, 2$.
3. $\mathbf{c}$ *is the cost vector, where* c_a *denotes the immediate cost when action* $a \in \mathcal{A}$ *is taken.*
4. P *is the probability transition matrix. After action* $a \in \mathcal{A}_s$ *is taken at state* $s \in \mathcal{S}$, *the next state is chosen according to* $\mathbf{p}_a$, *where* $\mathbf{p}_a$ *is point-mass distribution for all* $a \in \mathcal{A}$.
5. $\gamma \in (0, 1)$ *is a fixed discount factor. If an infinite sequence of actions* $\{a_k\}_{k=0}^{\infty}$ *is taken, the total discounted cost of this action sequence is* $\sum_{k=0}^{\infty}\gamma^k c_{a_k}$.

We next define strategies and optimal strategies.

Definition 2 (**Strategy & Strategy profile).** *A strategy for Player* i *is a vector* π^i *such that Player* i *chooses action* π^i_s *at state* $s \in \mathcal{S}^i$, *for* $i = 1, 2$. *A strategy profile* $\pi = (\pi^1, \pi^2)$ *is a pair of strategies, one for each player.*

Each strategy profile can be viewed as a subgraph of the underlying directed graph of the corresponding TBDG. Specifically, each state $s \in \mathcal{S}$ is a vertex in the subgraph, and the corresponding action π_s is an out-going edge. The out-degree of each vertex is 1, and each vertex and edge lies either on a cycle or on a path in this subgraph. For clarity, we specify the following definitions.

Definition 3 (**Directed Cycle).** *A directed cycle in a directed graph is a non-empty directed trail in which only the first and last vertices are equal.*

Definition 4. *For any strategy profile* $\pi = (\sigma, \tau)$, *we denote by* $\mathrm{Path}(\pi)$ *the set of paths and by* $\mathrm{Cyc}(\pi)$ *the set of cycles in the corresponding subgraph. With a slight abuse of notation, we also regard the states and actions appearing on these paths and cycles as elements of* $\mathrm{Path}(\pi)$ *and* $\mathrm{Cyc}(\pi)$, *respectively.*

By definition, a strategy profile $\pi = (\pi^1, \pi^2)$ of a TBSG specifies one action for Player 1 in each state of $\mathcal{S}^1$, denoted by $\pi^1 = \{\pi_1, \ldots, \pi_{n_1}\}$, and one action for Player 2 in each state of $\mathcal{S}^2$, denoted by $\pi^2 = \{\pi_{n_1+1}, \ldots, \pi_n\}$. An optimal strategy profile is one in which no player can improve by switching actions at any state when optimizing the expected discounted sum over the infinite horizon with discount factor γ. Here, $\gamma > 0$ is assumed to be strictly less than one but may be arbitrarily close to one. It is known that an optimal strategy profile exists for TBSGs.

Let v_s denote the cost-to-go value of state s under a strategy profile π of a TBSG. Then π is an optimal strategy profile if and only if $\mathbf{v}^* = (v_1^*, \ldots, v_n^*)$ is a fixed point of the following operator Hansen et al. (2013); Shapley (1953):

$$v_s^* := \min_{a\in\mathcal{A}_s}\{c_a + \gamma\mathbf{p}_a^T\mathbf{v}^*\} \quad (s \in \mathcal{S}^1), \qquad v_s^* := \max_{a\in\mathcal{A}_s}\{c_a + \gamma\mathbf{p}_a^T\mathbf{v}^*\} \quad (s \in \mathcal{S}^2),$$

and the corresponding optimal strategy profile is

$$\pi_s^* := \arg\min_{a\in\mathcal{A}_s}\{c_a + \gamma\mathbf{p}_a^T\mathbf{v}^*\} \quad (s \in \mathcal{S}^1), \qquad \pi_s^* := \arg\max_{a\in\mathcal{A}_s}\{c_a + \gamma\mathbf{p}_a^T\mathbf{v}^*\} \quad (s \in \mathcal{S}^2).$$

The following lemma and definitions are known; see, for example, Hansen et al. (2013); Shapley (1953).

Lemma 1 (**Value vectors).** *Let* π *be any strategy profile, let* P^π *be the* n *by* n *matrix where* $P^\pi_{s,s'}$ *is the probability of transitioning from* s' *to* s *using the actions in* π, *and let* $\mathbf{c}^\pi$ *be the vector of costs for each state under* π. *Then the state-indexed cost-to-go value vector* $\mathbf{v}^\pi$ *satisfies* $\mathbf{v}^\pi = \mathbf{c}^\pi + \gamma(P^\pi)^T\mathbf{v}^\pi$ *or* $\mathbf{v}^\pi = (I - \gamma P^\pi)^{-T}\mathbf{c}^\pi$.

Definition 5 (**Optimal counter strategy).** *Let* τ *be a strategy of Player 2. A strategy* σ *for Player 1 is an optimal counter strategy against* τ *if and only if* $\mathbf{v}^{(\sigma,\tau)} \leq \mathbf{v}^{(\sigma',\tau)}$ *for every possible strategy* σ' *of Player 1. Similarly, a strategy* τ *for Player 2 is an optimal counter strategy against* σ *if and only if* $\mathbf{v}^{(\sigma,\tau')} \leq \mathbf{v}^{(\sigma,\tau)}$ *for every possible strategy* τ' *of Player 2. For any given strategy* σ *of Player 1, let* $\pi^2(\sigma)$ *denote the optimal counter strategy of Player 2. Similarly, for any given strategy* τ *of Player 2, let* $\pi^1(\tau)$ *denote the optimal counter strategy of Player 1.*

Definition 6 (Optimal strategy profile). *A strategy profile $\pi = (\sigma, \tau)$ is optimal if and only if σ is an optimal counter strategy against τ and τ is an optimal counter strategy against σ. In this case, σ is an optimal strategy for Player 1 and τ is an optimal strategy for Player 2.*

In addition to the value vector, a strategy profile induces an associated flux vector $\mathbf{x}^\pi$, which plays a critical role in our analysis. The flux can be interpreted as a "discounted flow." Suppose that we start with one unit of mass at every state and then run the Markov chain. At each time step, we remove a fraction $1 - \gamma$ of the mass from each state and redistribute the remaining mass according to the strategy π. Summing over all time steps, the total amount of mass passing through each action is its flux. More formally, we have the following lemma (see Post & Ye (2015)).

Lemma 2. *Let π be a strategy profile and let P^π be the n by n matrix where $P^\pi_{s',s}$ is the probability of transitioning from s to s' using the actions in π. The action-indexed flux vector $\mathbf{x}^\pi$ has the following properties: if action a is not taken in the strategy profile, then $x_a = 0$; if π takes a in state s, then the nonzero state-indexed flux vector $\mathbf{x}^\pi_+$ satisfies $\mathbf{x}^\pi_+ = (I - \gamma P^\pi)^{-1}\mathbf{e}$, where $\mathbf{e}$ is the vector of all ones in dimension n. The flux represents the total discounted number of times each action is used when the TBSG starts from all states and evolves according to the Markov chain P^π with discounting at each iteration.*

A flux vector $\mathbf{x}^\pi$ can be viewed as a basic feasible solution to the following minimax "linear programming (LP)" problem:

$$\begin{array}{ll} \min\max & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = \mathbf{e}, \quad \mathbf{x} \geq \mathbf{0}, \end{array} \quad \text{or} \quad \begin{array}{ll} \min\max & (\mathbf{c}^1)^T\mathbf{x}^1 + (\mathbf{c}^2)^T\mathbf{x}^2 \\ \text{subject to} & A^1\mathbf{x}^1 + A^2\mathbf{x}^2 = \mathbf{e}, \quad \mathbf{x} \geq \mathbf{0}, \end{array} \tag{1}$$

where $\mathbf{c} \in \mathbb{R}^m$ and $A \in \mathbb{R}^{n \times m}$. The a-th entry of $\mathbf{c}$ is c_a, and the a-th column of A is $\mathbf{e}_s - \gamma \mathbf{p}_a$ where $a \in \mathcal{A}_s$, that is, action a belongs to state s. Here, $\mathbf{e}_s$ is the vector whose s-th entry is one and whose other entries are zero. The flux vector $\mathbf{x} \in \mathbb{R}^m$ is partitioned as $\mathbf{x} = (\mathbf{x}^1; \mathbf{x}^2)$, where $\mathbf{x}^1 = (x_1; \ldots; x_{m_1})$ and $\mathbf{x}^2$ contains the remaining components; similarly, $\mathbf{c} = (\mathbf{c}^1; \mathbf{c}^2)$ and $A = [A^1, A^2]$. The minimax representation associates $\mathbf{x}^1$ with minimization and $\mathbf{x}^2$ with maximization of the displayed linear objective, subject to the joint flow and nonnegativity constraints.

Under the above "LP-like" representation, a flux vector $\mathbf{x}$ is optimal, or equivalently corresponds to an optimal strategy profile, if and only if, for a value vector $\mathbf{v}$,

$$(\mathbf{r}^1)^T\mathbf{x}^1 = (\mathbf{r}^2)^T\mathbf{x}^2 = 0, \quad \mathbf{r} = \mathbf{c} - A^T\mathbf{v}, \quad \mathbf{r}^1 \geq \mathbf{0}, \quad \text{and} \quad \mathbf{r}^2 \leq \mathbf{0}, \tag{2}$$

where $\mathbf{r}$ is the reduced-cost vector associated with the value vector $\mathbf{v}$, $\mathbf{r}^1 = \mathbf{c}^1 - (A^1)^T\mathbf{v}$, and $\mathbf{r}^2 = \mathbf{c}^2 - (A^2)^T\mathbf{v}$.

1.1 Literature and background

Stochastic games were introduced by Shapley as a dynamic extension of matrix games in which the state evolves according to actions chosen by the players Shapley (1953). The model has since developed along several related lines, including recursive games and undiscounted stochastic games Gillette (1957), as well as the broader theory of competitive Markov decision processes Filar & Vrieze (2012). Markov decision processes form the one-player specialization of this framework and provide the classical algorithmic baseline through dynamic programming, policy iteration, and linear programming Howard (1960); Puterman (2014).

The computational complexity of MDPs has been studied from several complementary viewpoints. Papadimitriou and Tsitsiklis analyzed the complexity of finite-horizon, discounted, and average-cost MDPs Papadimitriou & Tsitsiklis (1987), while Littman, Dean, and Kaelbling surveyed complexity issues surrounding MDP solution methods in planning and artificial intelligence Littman et al. (1995). For discounted MDPs, the complexity landscape is governed by two closely related distinctions: polynomial versus strongly polynomial complexity, and a fixed discount factor versus one supplied as part of the input. Although finite discounted MDPs are polynomial-time solvable through linear programming, strong polynomiality requires a bound on the number of arithmetic operations that is independent of the numerical encoding, including how close γ is to one. Ye obtained a strongly polynomial algorithm for discounted MDPs with a fixed discount factor using an interior-point method Ye (2005), and subsequently established strongly polynomial iteration bounds for the simplex

and policy-iteration methods in the same setting Ye (2011). The iteration complexity of policy-improvement methods is nevertheless more subtle: Mansour and Singh provided early worst-case upper bounds for policy iteration Mansour & Singh (1999), whereas Melekopoglou and Condon, Fearnley, and subsequent work established exponential lower bounds for several policy-improvement variants Melekopoglou & Condon (1994); Fearnley (2010). Hollanders, Delvenne, and Jungers further demonstrated exponential behavior for discounted MDPs when the discount factor is part of the input Hollanders et al. (2012). Stronger results become possible under additional structural restrictions. For deterministic MDPs, Post and Ye proved that the simplex method with the most-negative-reduced-cost rule is strongly polynomial even with an arbitrary uniform discount factor, obtaining an $O(n^3m^2 \log^2 n)$ iteration bound Post & Ye (2015). More generally, Scherrer's structural bounds for policy iteration show that restrictions on transient and recurrent behavior can eliminate explicit dependence on the discount factor for broader classes of MDPs Scherrer (2013). These results provide an important precedent for the present paper: recurrence need not obstruct strong polynomiality when the underlying transition structure is sufficiently controlled.

The two-player setting is considerably more delicate. Hansen, Miltersen, and Zwick proved that strategy iteration is strongly polynomial for turn-based stochastic games when the discount factor is fixed, although their iteration bound depends explicitly on $1/(1-\gamma)$ and therefore does not yield strong polynomiality when γ is part of the input Hansen et al. (2013). Jia, Wen, and Ye subsequently developed the simplex strategy-iteration framework adopted in this paper, establishing geometric convergence and corresponding complexity guarantees for general TBSGs in the fixed-discount regime Jia et al. (2020). Complementary exact and complexity-theoretic analyses further illuminate the difficulty of solving stochastic games on graphs Andersson & Miltersen (2009); Hansen et al. (2011), while exponential lower bounds for strategy-improvement procedures demonstrate that local pivoting rules can perform poorly on unrestricted two-player game classes Friedmann (2011). Thus, when the discount factor is part of the input, strongly polynomial algorithms are not known for broad classes of TBSGs or TBDGs. Against this backdrop, we ask whether strong polynomiality can be recovered by imposing a natural restriction on the recurrent structure of the game graph.

Motivated by this complexity landscape, we study a special class of turn-based deterministic games, called *Turn-Based Deterministic Forward Games* (TBDFGs). In a TBDFG, the underlying directed game graph contains no directed cycle involving actions from both players. This forward-progression condition does not require the graph to be acyclic: recurrent behavior is allowed, but only within states controlled by a single player. Consequently, TBDFGs form a natural intermediate class between acyclic deterministic games and general cyclic turn-based deterministic games. They are substantially more general than finite-horizon or tree-structured models, while still excluding mixed-player feedback loops in which both players can repeatedly influence the same directed cycle. This balance makes TBDFGs both practically relevant and mathematically tractable.

Board games such as Chess and Go motivate a staged interpretation: play progresses through successive decision stages, with terminal outcomes represented by absorbing states. In the representation discussed in Appendix A, prohibited repetitions in Go are modeled as terminal invalid moves rather than transitions to earlier stages. TBDFGs allow richer recurrence than a game tree: local cycles may occur within a stage, provided that each cycle is controlled by one player. The same forward principle applies to hierarchical decision processes with progress measured by time, resources, or task completion. AlphaGo and AlphaGo Zero illustrate the algorithmic value of policy improvement, value estimation, and search Silver et al. (2016; 2017). Our goal is complementary: rather than designing a learning system for Go, we establish a discount-independent complexity guarantee for simple strategy iteration on this structural subclass. Additional discussion and examples are given in Appendix A.

We make the formal definition of TBDFGs as the following.

Definition 7 (**TBDFG**). *A* Turn-Based Deterministic Forward Game (TBDFG) *is a* TBDG *satisfying the additional* forward-progression *property that the underlying game graph contains no directed cycle involving actions from both players.*

The forward game setting also covers hierarchical decision processes with single-player local recurrence. Further graph-game motivation and examples are given in Appendix A.

The main challenge is to control the interaction between the players during strategy iteration. Unlike an acyclic game, a TBDFG may contain recurrent components, and Player 1's updates can change

Player 2's optimal response along transient paths. We show that the forward condition nevertheless suffices to bound the number of simplex pivots independently of the discount factor. Our contributions are as follows.

(i) We formalize the TBDFG, a structural subclass of TBDG in which no directed cycle contains actions from both players. This class is motivated by forward-progression structures in staged games, deterministic planning, and graph games with local recurrence.

(ii) We analyze the simple strategy-iteration method of Hansen et al. (2013); Jia et al. (2020), which is not tailored to TBDFGs but a general strategy iteration for solving TBSGs. We prove that this general algorithm converges in strongly polynomial time on TBDFGs, with an bounded of $O(n^6 m^4 \log^4 n)$ on total simplex pivot steps for any uniform discount factor $\gamma \in (0, 1)$.

The rest of this paper is organized as follows. In Subsection 1.2, we introduce notation, terminology, and known properties for TBDGs. In Section 2, we study the general simple strategy-iteration algorithm and prove that it is strongly polynomial for TBDFGs. Section 3 summarizes our results. Additional game discussion and examples are provided in Appendix A, and deferred proofs are given in Appendix B.

1.2 Preliminaries

We use the following notation throughout the paper. Let $\mathbb{R}^n$ denote the Euclidean space of dimension n, and let $\mathbb{N}$ denote the set of all natural numbers. For any $n \in \mathbb{N} \setminus \{0\}$, let $[n]$ denote the set $\{1, \dots, n\}$. Let $\mathbf{e}$ and $\mathbf{0}$ be the all-ones and all-zeros vectors, respectively, with dimensions clear from the context. To simplify the discussion, we use the following equivalent scaled "LP-like" representation.

$$\begin{array}{ll} \min\max & \mathbf{c}^T\mathbf{x} \\ \text{subject to} & A\mathbf{x} = (1-\gamma)\mathbf{e} := \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}, \end{array} \quad \text{or} \quad \begin{array}{ll} \min\max & (\mathbf{c}^1)^T\mathbf{x}^1 + (\mathbf{c}^2)^T\mathbf{x}^2 \\ \text{subject to} & A^1\mathbf{x}^1 + A^2\mathbf{x}^2 = \mathbf{b}, \quad \mathbf{x} \geq \mathbf{0}. \end{array} \tag{3}$$

The formulation in (3) is equivalent to (1) up to the scaling factor $(1-\gamma)$. This notation simplifies the discussion below, while all previous results, namely Lemmas 1 and 2 and (2), continue to hold after replacing $\mathbf{e}$ with $\mathbf{b}$. The following results follow from these and by applying the $(1-\gamma)$ scaling to the proofs in Hansen et al. (2013); Shapley (1953).

Proposition 1. *For any TBDG given by $\mathcal{G} = (\mathcal{S}, \mathcal{A}, \mathbf{c}, \mathbf{P}, \gamma)$ and any strategy profile π of the game, the following properties hold.*

(i) *The state-indexed cost-to-go value vector $\mathbf{v}^\pi$ satisfies $\mathbf{v}^\pi = \mathbf{c}^\pi + \gamma(P^\pi)^T\mathbf{v}^\pi$ or $\mathbf{v}^\pi = (I - \gamma P^\pi)^{-T}\mathbf{c}^\pi$, where P^π is the n by n matrix where $P^\pi_{s,s'}$ is the probability of transitioning from s' to s using the actions in π and $\mathbf{c}^\pi$ be the vector of costs for each state under π.*

(ii) *The associated (scaled) flux vector $\mathbf{x}^\pi$ satisfies $\mathbf{x}^\pi_a = 0$ for all $a \notin \pi$, and $\mathbf{x}^\pi_+ = (I - \gamma P^\pi)^{-1}\mathbf{b}$, where $\mathbf{b} = (1-\gamma)\mathbf{e}$ and $\mathbf{x}^\pi_+$ is the nonzero state-index flux vector*

(iii) *The feasible polytope of* (3) *is nondegenerate for $0 < \gamma < 1$, and there is a bijection between vertices and strategy profiles of the TBDG.*

(iv) *For the (scaled) flux vector, we have $\sum_{a\in\pi} \mathbf{x}^\pi_a = n$ and $\mathbf{x}^\pi_a = 0$ for all $a \notin \pi$. Furthermore, we have $1 - \gamma \leq x^\pi_a \leq (1-\gamma)n$ if $a \in \pi$ is a path-action and $1 \leq x^\pi_a \leq n$ if $a \in \pi$ is a cycle-action.*

(v) *The reduced-cost vector is $\mathbf{r}^\pi = \mathbf{c} - A^T\mathbf{v}^\pi$ where $A = I - \gamma P$. Moreover, we have $(\mathbf{r}^\pi)^T\mathbf{x}^\pi = 0$ and $\mathbf{c}^T\mathbf{x}^\pi = (\mathbf{c}^\pi)^T\mathbf{x}^\pi_+ = \mathbf{b}^T\mathbf{v}^\pi$. Furthermore, for any two strategy profiles π, π', we have*

$$\mathbf{b}^T(\mathbf{v}^{\pi'} - \mathbf{v}^\pi) = (\mathbf{r}^\pi)^T\mathbf{x}^{\pi'} = (\mathbf{r}^\pi_{\pi'})^T\mathbf{x}^{\pi'}_+.$$

where $\mathbf{r}^\pi_{\pi'}$ is the truncation of $\mathbf{r}^\pi$ on the set of actions used in π'.

(vi) *In particular, let $\pi^* = (\pi^1_*, \pi^2_*)$ be the optimal strategy profile and denote the objective value by ν^*. Then we have*

$$\nu^* := \mathbf{c}^T\mathbf{x}^{\pi_*} = (\mathbf{c}^{\pi_*})^T x^{\pi_*}_+ = \mathbf{b}^T\mathbf{v}^{\pi_*} \quad \textit{and} \quad (\mathbf{r}^{\pi_*})^T\mathbf{x}^{\pi_*} = 0. \tag{4}$$

(vii) *A flux vector $\mathbf{x}$ is optimal, or equivalently, corresponds to an optimal strategy profile, if and only if, for a value vector $\mathbf{v}$,*

$$(\mathbf{r}^1)^T\mathbf{x}^1 = (\mathbf{r}^2)^T\mathbf{x}^2 = 0, \quad \mathbf{r} = \mathbf{c} - A^T\mathbf{v}, \quad \mathbf{r}^1 \geq \mathbf{0}, \quad \textit{and} \quad \mathbf{r}^2 \leq \mathbf{0}, \tag{5}$$

where $\mathbf{r}$ is the reduced-cost vector associated with the value vector $\mathbf{v}$, $\mathbf{r}^1 = \mathbf{c}^1 - (A^1)^T\mathbf{v}$, and $\mathbf{r}^2 = \mathbf{c}^2 - (A^2)^T\mathbf{v}$.

For any fixed strategy of either player, finding the other player's optimal counter strategy in a TBDG reduces to a deterministic MDP. The following result is a corollary of (Post & Ye, 2015, Theorem 3.7).

Proposition 2. *For a TBDG, using the simplex method with the simple pivoting rule, for any fixed strategy σ of Player 1, the optimal counter strategy $\pi^2(\sigma)$ can be found in $O(n^3m^2\log^2 n)$ pivot steps of simplex method; and for any fixed strategy τ of Player 2, the counter optimal strategy $\pi^1(\tau)$ can be found in $O(n^3m^2\log^2 n)$ pivot steps of simplex method.*

2 Simple strategy-iteration algorithm and complexity results

In this section, we study the simple strategy-iteration method for TBSGs and establish its strongly polynomial complexity on TBDFGs. The algorithm does not test the forward condition or alter its pivot rule for this class; the condition is used only in the convergence analysis.

The strategy-iteration method extends the original policy-iteration method for MDPs Howard (1960). It updates the action at a state $s \in \mathcal{S}^1$ with the most negative reduced cost in $\mathbf{r}$ under the strategy profile $\pi = (\sigma, \pi^2(\sigma))$. Specifically, for every state $s \in \mathcal{S}^1$, if $\delta_s = \min\{r_a : a \in \mathcal{A}_s\} < 0$, then the current action in strategy σ is switched to $a_s = \arg\min\{r_a : a \in \mathcal{A}_s\}$ at state s; ties may be broken arbitrarily in each state. The method then repeats with the new strategy profile $\pi^+ = (\sigma^+, \pi^2(\sigma^+))$, where possibly multiple actions in σ are replaced. Each iteration can be completed in strongly polynomial time (see Proposition 2). The difference between the standard strategy-iteration algorithm and the simple strategy-iteration algorithm is that the latter replaces the action in only one state $s \in \mathcal{S}^1$, namely, the state with the smallest negative reduced cost.

We now state the simple strategy-iteration algorithm, which coincides with Algorithm 1 in Jia et al. (2020). In each iteration, Algorithm 1 updates Player 1's strategy through a simple pivoting step, with ties broken arbitrarily, and then finds Player 2's optimal strategy against the updated Player 1 strategy using simplex method with simple pivoting rule. The algorithm can be described as follows.

Algorithm 1 Simple strategy-iteration algorithm

Input: Initialize $\pi_0 = (\sigma_0, \tau_0)$, where $\tau_0 = \pi^2(\sigma_0)$ and $\mathbf{v}^{\pi_0}$ are as defined in Proposition 1. Set $k = 1$.

1: Run one simplex step for Player 1, and denote the updated Player 1 strategy by σ_k; namely, pivot in the action with the smallest negative modified cost for Player 1.
2: Call the Simplex method with the simple pivoting rule in (Post & Ye, 2015, Section 2) with initial strategy τ_{k-1} to find the optimal strategy for Player 2, and set $\tau_k = \pi^2(\sigma_k)$.
3: Update the value function $\mathbf{v}^{\pi_k}$, where $\pi_k = (\sigma_k, \tau_k)$.
4: If $\mathbf{v}^{\pi_k} \neq \mathbf{v}^{\pi_{k-1}}$, set $k \leftarrow k + 1$ and go to Step 1. Otherwise, terminate the algorithm and return $\pi_k = (\sigma_k, \tau_k)$.

Remark 1. *(i) In Step 1 of Algorithm 1, Player 1's strategy is not updated if there is no action with negative modified cost. In particular, if π_0 is already an optimal strategy profile, then Step 1 leaves the strategy profile unchanged, and the algorithm terminates without performing any update. (ii) By Proposition 1(iii), the feasible polytope in* (3) *is nondegenerate, and its vertices are in one-to-one correspondence with strategy profiles of the TBDG. Hence the simplex pivot in Step 1 of Algorithm 1 is well defined as a move between strategy profiles. Moreover, by Proposition 2, once Player 1's strategy σ_k is fixed, Player 2's optimal counter strategy $\tau_k = \pi^2(\sigma_k)$ in Step 2 can be computed by the simplex method with the simple pivoting rule in strongly polynomial iteration time. Therefore each iteration of Algorithm 1 is well defined as a strategy-iteration procedure for TBDGs.*

It is useful to distinguish the algorithmic template from the complexity analysis. Algorithm 1 is the same simple strategy-iteration scheme studied for general TBSGs in Hansen et al. (2013); Jia et al. (2020); the novelty here is not a new pivot rule, but a stronger guarantee for the TBDFG subclass. In particular, the bound proved below is strongly polynomial when the uniform discount factor is part of the input, and it contains no dependence on $1/(1-\gamma)$. The proof is inspired in part by the reduced-cost and flux analysis of the simplex method for deterministic MDPs in Post & Ye (2015), but the two-player setting creates a difficulty that is absent from a one-player minimization problem. After each Player 1 pivot, Player 2 is reoptimized, so the resulting sequence is governed by a

minimax interaction rather than by a single monotone simplex trajectory. The TBDFG condition rules out directed cycles that contain actions of both players, thereby excluding mixed-player recurrent feedback. Nevertheless, Player 1 and Player 2 actions may still alternate along transient paths before the play enters a one-player recurrent component. This transient interaction is the main new obstacle in the proof: unlike the one-player simplex analysis of Post & Ye (2015), progress cannot be measured solely by the pivot sequence of Player 1, since each such pivot may change Player 2's optimal response; the analysis below must therefore track the cumulative effects of these counter-optimizations along mixed-player paths, which requires new estimates beyond those used in the MDP setting in Post & Ye (2015).

We next prove that Algorithm 1 computes optimal strategies for both players in time polynomial in m, n, and $\log n$, as stated in Theorem 1. We begin with a property of optimal counter strategies that holds for all TBDGs.

Lemma 3. *Denote by ν^1 the objective value of any strategy profile $(\sigma, \pi^2(\sigma))$, and denote by ν^2 the objective value of any strategy profile $(\pi^1(\tau), \tau)$. Then $\nu^2 \leq \nu^* \leq \nu^1$. Moreover, the corresponding reduced-cost vector satisfies $r_a^{(\sigma,\pi^2(\sigma))} \leq 0$ for all $a \in \mathcal{A}^2$ and $r_a^{(\pi^1(\tau),\tau)} \geq 0$ for all $a \in \mathcal{A}^1$, and $r_a^\pi = 0$ for all actions taken in the strategy profile.*

Proof. For any strategy σ of Player 1, let the strategy profile be $\pi = (\sigma, \pi_*^2)$. Then we have

$$\mathbf{c}^T\mathbf{x}^\pi - \nu^* = (\mathbf{r}^{\pi^*})^T\mathbf{x}^\pi = \sum\nolimits_{a\in\mathcal{A}^1} r_a^{\pi^*} x_a^\pi + \sum\nolimits_{a\in\mathcal{A}^2} r_a^{\pi^*} x_a^\pi \geq 0,$$

where the first equality follows from Proposition 1(v) and (4), and the inequality follows from $\mathbf{x}^\pi \geq \mathbf{0}$, $r_a^{\pi^*} \geq 0$ for any $a \in \mathcal{A}^1$, $r_a^{\pi^*} = 0$ for any $a \in \pi_*^2$, and $\mathbf{x}_a^\pi = 0$ for any $a \in \mathcal{A}^2/\pi_*^2$. On the other hand, ν^1 is the objective value of the strategy profile $(\sigma, \pi^2(\sigma))$, where $\pi^2(\sigma)$ is the optimal counter strategy of Player 2 against σ and maximizes the objective value. Thus ν^1 should be at least the value of the strategy profile $\pi = (\sigma, \pi_*^2)$, namely $\nu^1 \geq \nu^*$. Similarly, one can prove $\nu^* \geq \nu^2$. Finally, the reduced-cost sign property follows directly from the counter-optimality of the other player, since the resulting problem is an MDP. □

We next prove that the value vector decreases monotonically throughout the iterations. The proof is inspired by (Hansen et al., 2013, Lemma 5.1) and holds for TBDGs.

Lemma 4. *The value function obtained by Algorithm 1 decreases strictly at each iteration, i.e., $\mathbf{v}^{\pi_{k-1}} \geq \mathbf{v}^{\pi_k}$ and $\mathbf{b}^T\mathbf{v}^{\pi_{k-1}} > \mathbf{b}^T\mathbf{v}^{\pi_k}$, unless Algorithm 1 terminates. The proof is given in Appendix B.1.*

2.1 Improvement with a new cycle variable

In this subsection, we show that the objective value $\mathbf{b}^T\mathbf{v}$ improves significantly when a pivoted-in action of Player 1 becomes a cycle action. We first prove the following objective-reduction lemmas for the case in which a new cycle variable is introduced.

Lemma 5. *Let $\mathbf{v}^{\pi^*}$ be given in* (4) *and $\Delta_k = \min\{r_a^{\pi_k} : a \in \mathcal{A}^1\}$ be the minimum modified cost of the strategy profile π_k at the beginning of the $(k+1)$-th iteration. Then, we have*

$$\mathbf{b}^T\mathbf{v}^{\pi^*} \geq \mathbf{b}^T\mathbf{v}^{\pi_k} - n|\Delta_k|. \tag{6}$$

Proof. If $\Delta_k \geq 0$, then $\mathbf{v}^{\pi_k} = \mathbf{v}^{\pi^*}$ is the optimal value function, and this inequality holds automatically. Now, consider the case $\Delta_k < 0$. Let $\sigma_k^* = \pi^1(\tau_k)$. Then, it follows from Lemma 3 that

$$\mathbf{b}^T\mathbf{v}^{(\sigma_k^*,\tau_k)} \leq \mathbf{b}^T\mathbf{v}^{\pi^*}. \tag{7}$$

Additionally, based on the definition of σ_k^*, we have

$$\mathbf{b}^T\mathbf{v}^{(\sigma_k^*,\tau_k)} \geq \mathbf{b}^T\mathbf{v}^{(\sigma_k,\tau_k)} - n|\Delta_k|. \tag{8}$$

Since Δ_k has the same interpretation as the modified cost in linear programming, this inequality follows from the simplex-method analysis for MDPs (see Post & Ye (2015)). Consequently, (6) follows by combining (7) and (8). □

We next show that a substantial improvement is achieved once a pivoted-in action becomes part of a cycle in a TBDFG.

Lemma 6. *Suppose the game is a TBDFG. Let $\mathbf{v}^{\pi^*}$ be given in* (4)*. Suppose a Player 1's cycle is created on the k-th iteration, then*

$$\mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi^*}) \leq (1 - 1/n)\mathbf{b}^T(\mathbf{v}^{\pi_{k-1}} - \mathbf{v}^{\pi^*}).$$

Proof. Let $a' \in \mathcal{A}^{s'}$ be the newly introduced action of Player 1 corresponding to the minimum modified cost during the k-th iteration. Since a Player 1 cycle is created during this iteration and Player 2's response cannot affect this cycle by the forward-progression property, a' is a cycle action of π_k at the end of this iteration. Observe that

$$\begin{aligned}\mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi^*}) &= \mathbf{b}^T(\mathbf{v}^{\pi_{k-1}} - \mathbf{v}^{\pi^*}) - \mathbf{b}^T(\mathbf{v}^{\pi_{k-1}} - \mathbf{v}^{\pi_k}) \\ &= \mathbf{b}^T(\mathbf{v}^{\pi_{k-1}} - \mathbf{v}^{\pi^*}) + \sum\nolimits_{a\in\pi_k} r_a^{\pi_{k-1}} x_a^{\pi_k} \leq \mathbf{b}^T(\mathbf{v}^{\pi_{k-1}} - \mathbf{v}^{\pi^*}) + \sum\nolimits_{a\in\sigma_k} r_a^{\pi_{k-1}} x_a^{\pi_k} \\ &= \mathbf{b}^T(\mathbf{v}^{\pi_{k-1}} - \mathbf{v}^{\pi^*}) + r_{a'}^{\pi_{k-1}} x_{a'}^{\pi_k} \leq \mathbf{b}^T(\mathbf{v}^{\pi_{k-1}} - \mathbf{v}^{\pi^*}) + r_{a'}^{\pi_{k-1}},\end{aligned}$$

where the first equality is obtained by rearranging the terms, the second equality follows from Proposition 1(v), the first inequality follows from $\tau_{k-1} = \pi^2(\sigma_{k-1})$, namely, $r_a^{\pi_{k-1}} \leq 0$ for all $a \in \mathcal{A}^2$, the last equality follows from the updating scheme of Algorithm 1, namely, $r_a^{\pi_{k-1}} \mathbf{x}_a^{\pi_k} = 0$ for all $a \in \sigma_k$ and $a \neq a'$, and the last inequality follows from $r_{a'}^{\pi_{k-1}} \leq 0$ and Proposition 1(iv) for this cycle action a'. Note that $r_{a'}^{\pi_{k-1}} = \Delta_{k-1}$ in Lemma 5. This together with the above inequality and Lemma 5 yields

$$\mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi^*}) \leq \mathbf{b}^T(\mathbf{v}^{\pi_{k-1}} - \mathbf{v}^{\pi^*}) + r_{a'}^{\pi_{k-1}} \leq (1 - 1/n)\mathbf{b}^T(\mathbf{v}^{\pi_{k-1}} - \mathbf{v}^{\pi^*}). \qquad \square$$

2.2 Improvement before creating cycles

In this subsection, we analyze what happens when no new cycles of Player 1 are created. To simplify the discussion, we make the following definition.

Definition 8 (Optimal Cost without New Cycle). *Let Π_k be the set of strategy profiles such that any $\pi = (\sigma, \tau) \in \Pi_k$ satisfies $\tau = \pi^2(\sigma)$ and every Player 1's cycle in π is also a cycle in π_k. Define $\tilde{\nu}_k^* = \min_{\pi\in\Pi_k} \mathbf{b}^T\mathbf{v}^\pi$ as the optimal cost for π_k without creating any new Player 1's cycle relative to π_k.*

The following lemma gives a lower bound on the optimal cost without a new Player 1's cycle using the forward progression property of TBDFG.

Lemma 7. *Suppose the game is a TBDFG. Let $\Delta_k = \min\{r_a^{\pi_k} : a \in \mathcal{A}^1\}$ be the minimum modified cost of the strategy profile π_k at the beginning of the $(k+1)$-th iteration. Then we have $\tilde{\nu}_k^* \geq \mathbf{b}^T\mathbf{v}^{\pi_k} - (1-\gamma)n^2|\Delta_k|$.*

Proof. In the following, we show that for any $\pi \in \Pi_k$ we have $\mathbf{b}^T\mathbf{v}^\pi \geq \mathbf{b}^T\mathbf{v}^{\pi_k} - (1-\gamma)n^2|\Delta_k|$. The result then follows by taking the minimum over all $\pi \in \Pi_k$. Observe that

$$\begin{aligned}\mathbf{b}^T\mathbf{v}^\pi - \mathbf{b}^T\mathbf{v}^{\pi_k} &\geq \mathbf{b}^T\mathbf{v}^{(\sigma,\tau_k)} - \mathbf{b}^T\mathbf{v}^{\pi_k} = \left(\mathbf{r}_{(\sigma,\tau_k)}^{\pi_k}\right)^T \mathbf{x}_+^{(\sigma,\tau_k)} \\ &= \sum\nolimits_{a\in\sigma\setminus\mathrm{Cyc}(\pi_k)} r_a^{\pi_k} x_a^{(\sigma,\tau_k)} \geq -\sum\nolimits_{a\in\sigma\setminus\mathrm{Cyc}(\pi_k)} x_a^{(\sigma,\tau_k)}|\Delta_k| \geq -(1-\gamma)n^2|\Delta_k|,\end{aligned}$$

where the first inequality follows from the optimality of τ against σ, the first equality follows from Proposition 1(v), and the second equality follows from $r_a^{\pi_k} = 0$ for all $a \in \tau_k$ or $a \in \sigma \cap \mathrm{Cyc}(\pi_k)$, and the second inequalty follows from the definition of Δ_k. To justify the last inequality, note that every action in $\sigma \setminus \mathrm{Cyc}(\pi_k)$ must be a path action under (σ, τ_k). Otherwise, such an action would lie on a cycle of (σ, τ_k); by the TBDFG property, this cycle cannot contain Player 2 actions, and hence it would also be a Player 1 cycle in $\pi = (\sigma, \tau)$. Since $\pi \in \Pi_k$, this cycle would be covered by cycles in π_k, contradicting $a \notin \mathrm{Cyc}(\pi_k)$. The last inequality then follows from Proposition 1(iv), since there are at most n such actions and each path action has flux at most $(1-\gamma)n$. $\square$

We next show that a small improvement is still made when no Player 1's cycles are created for TBDFGs.

Lemma 8. *Suppose the game is a TBDFG. Let $k^+ \geq k$ be any iteration before creating a new Player 1's cycle (or algorithm termination). Then for any $k' \in \mathbb{N} \cap (k, k^+]$ the following inequality holds $\mathbf{b}^T(\mathbf{v}^{\pi_{k'}} - \mathbf{v}^{\pi_{k^+}}) \leq (1 - 1/n^2)\mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi_{k^+}})$. The proof is given in Appendix B.2.*

Combining the previous two lemmas, we analyze the possible long-run behavior of the algorithm for solving TBDFGs.

Lemma 9. *Suppose the game is a TBDFG. Then after running Algorithm 1 for $O(n^2 \log n)$ iterations from iteration k, one of the following must hold: (i) the algorithm terminates, (ii) a new Player 1's cycle is created, (iii) a Player 1's cycle is broken, (iv) some action in π_k will never appear until a new cycle is created. The proof is given in Appendix B.3.*

2.3 Strong polynomial complexity

In this subsection, we prove our main result that Algorithm 1 is strongly polynomial for TBDFGs. Using the previous lemma, we show that the algorithm cannot continue making only slight improvements indefinitely without either creating a new Player 1' cycle or terminating.

Lemma 10. *Suppose the game is a TBDFG. Either the algorithm terminates or a new Player 1's cycle is created after $O(n^2 m \log n)$ iterations of Algorithm 1.*

Proof. Let π_{k_0} be a strategy after a new Player 1's cycle is created, and consider the policies $\pi_{k_1}, \pi_{k_2}, \ldots$, each separated by $O(n^2 \log n)$ iterations. If no Player 1's cycle is created, then by Lemma 9, each of these policies π_{k_i} has either broken another Player 1's cycle in π_{k_0} or contains an action that cannot appear in π_{k_j} for all $j > i$. There are at most n Player 1's cycles in π_{k_0} and at most m actions that can be eliminated. Therefore, after $(m + n)O(n^2 \log n)$ iterations, the algorithm must terminate or create a new cycle. □

Using the significant improvement achieved when a new cycle is created with Player 1's pivoted in action, we establish the following lemma.

Lemma 11. *Suppose the game is a TBDFG. Let π be a strategy profile at some iteration of Algorithm 1. Starting from π, after $O(n \log n)$ Player 1's cycle-creating rounds, some action in $\pi \cap \mathcal{A}^1$ is either eliminated from cycles for the remainder of the algorithm or entirely eliminated from policies for the remainder of the algorithm. The proof is given in Appendix B.4.*

We close the section with the complexity bound for Algorithm 1 and show that TBDFG is strongly polynomial-time solvable.

Theorem 1. *The simple strategy-iteration method in Algorithm 1 converges in at most $O(n^6 m^4 \log^4 n)$ pivot steps of simplex method on TBDFGs with uniform discount $\gamma \in (0, 1)$. Therefore, the simple strategy-iteration method is strongly polynomial for the TBDFG.*

Proof. Consider the policies $\pi_{k_0}, \pi_{k_1}, \ldots$ such that $O(n \log n)$ new Player 1's cycles are created between π_{k_i} and $\pi_{k_{i+1}}$. By Lemma 11, each π_{k_i} contains an action that is either eliminated entirely from π_{k_j} for $j > i$ or eliminated from cycles. Each action can be eliminated from both cycles and paths, so after $2m$ such rounds of $O(n \log n)$ new Player 1's cycles, the algorithm has converged. By Proposition 2 and 10, each outer iteration takes $O(n^3 m^2 \log^2 n)$ pivot steps of simplex method, and Player 1's cycles are created every $O(n^2 m \log n)$ iterations. Thus the algorithm terminates in $O(n^6 m^4 \log^4 n)$ total pivot steps of simplex method.

□

3 Summary of contributions

This paper studies TBDFGs, a class of deterministic turn-based games in which every directed cycle is controlled by a single player. We show that the general strategy-iteration method in Algorithm 1 follows a discount-independent strongly polynomial complexity on every TBDFG, even though the algorithm neither recognizes nor explicitly exploits the forward property. This is a structure-oblivious guarantee for a standard general purpose strategy-iteration method.

AI USE STATEMENT

We used ChatGPT to assist with language polishing and LaTeX formatting. The authors take full responsibility for the final manuscript, including the accuracy of all statements, mathematical results, and proofs.

REFERENCES

Daniel Andersson and Peter Bro Miltersen. The complexity of solving stochastic games on graphs. In *International Symposium on Algorithms and Computation*, pp. 112–121. Springer, 2009.

Cristian S Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pp. 252–263, 2017.

Anne Condon. The complexity of stochastic games. *Information and Computation*, 96(2):203–224, 1992.

Andrzej Ehrenfeucht and Jan Mycielski. Positional strategies for mean payoff games. *International Journal of Game Theory*, 8(2):109–113, 1979.

E. Allen Emerson and Charanjit S. Jutla. Tree automata, μ-calculus and determinacy. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science*, pp. 368–377. IEEE Computer Society, 1991. doi: 10.1109/SFCS.1991.185392.

John Fearnley. Exponential lower bounds for policy iteration. In *International Colloquium on Automata, Languages, and Programming*, pp. 551–562. Springer, 2010.

Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

Oliver Friedmann. An exponential lower bound for the latest deterministic strategy iteration algorithms. *Logical Methods in Computer Science*, 7, 2011.

Dean Gillette. Stochastic games with zero stop probabilities. *Contributions to the Theory of Games III*, 39:179–187, 1957.

Kristoffer Arnsfelt Hansen, Michal Koucky, Niels Lauritzen, Peter Bro Miltersen, and Elias P Tsigaridas. Exact algorithms for solving stochastic games. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pp. 205–214, 2011.

Thomas Dueholm Hansen, Peter Bro Miltersen, and Uri Zwick. Strategy iteration is strongly polynomial for 2-player turn-based stochastic games with a constant discount factor. *Journal of the ACM (JACM)*, 60(1):1–16, 2013.

Romain Hollanders, Jean-Charles Delvenne, and Raphaël M Jungers. The complexity of policy iteration is exponential for discounted markov decision processes. In *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*, pp. 5997–6002. IEEE, 2012.

Ronald A Howard. *Dynamic programming and markov processes.* John Wiley, 1960.

Zeyu Jia, Zaiwen Wen, and Yinyu Ye. Towards solving 2-TBSG efficiently. *Optimization Methods and Software*, 35(4):706–721, 2020.

Michael L. Littman, Thomas L. Dean, and Leslie Pack Kaelbling. On the complexity of solving markov decision problems. In *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 394–402. Morgan Kaufmann, 1995. doi: 10.5555/2074158.2074203.

Walter Ludwig. A subexponential randomized algorithm for the simple stochastic game problem. *Information and computation*, 117(1):151–155, 1995.

Yishay Mansour and Satinder Singh. On the complexity of policy iteration. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, pp. 401–408. Morgan Kaufmann, 1999. doi: 10.5555/2073796.2073842.

Mary Melekopoglou and Anne Condon. On the complexity of the policy improvement algorithm for markov decision processes. *ORSA Journal on Computing*, 6(2):188–192, 1994.

Christos H Papadimitriou and John N Tsitsiklis. The complexity of markov decision processes. *Mathematics of operations research*, 12(3):441–450, 1987.

Ian Post and Yinyu Ye. The simplex method is strongly polynomial for deterministic markov decision processes. *Mathematics of Operations Research*, 40(4):859–868, 2015.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Bruno Scherrer. Improved and generalized upper bounds on the complexity of policy iteration. *Advances in Neural Information Processing Systems*, 26, 2013.

Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10): 1095–1100, 1953.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

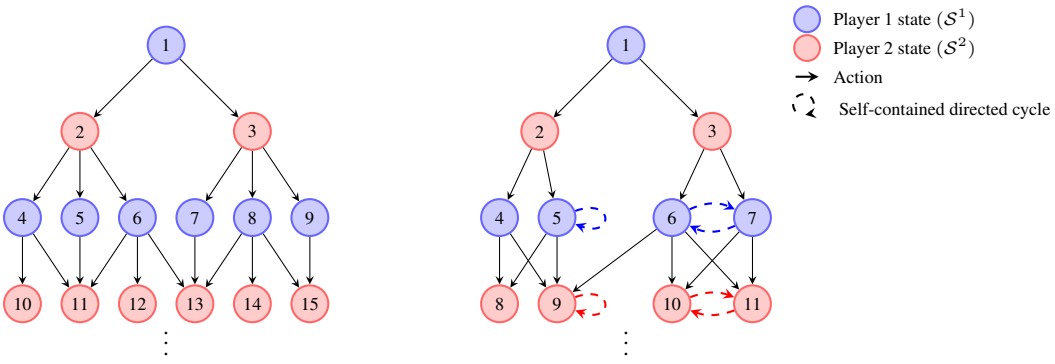
Yinyu Ye. A new complexity result on solving the markov decision problem. *Mathematics of Operations Research*, 30(3):733–749, 2005.

Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Mathematics of Operations Research*, 36(4):593–603, 2011. doi: 10.1287/moor.1110.0516.

Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1-2):343–359, 1996.

A ADDITIONAL GAME DISCUSSION AND EXAMPLES

Board games provide important motivating examples as a subclass of TBDFG. The game is divided into finitely many stages, where each stage consists of states of one of the two players and the immediate following stage consists of the states of the other player. The game is forward in the sense that no state-action transition can return to a state in an earlier stage. At the same time, the model does not require each stage to be acyclic: states controlled by the same player in the same stage may still transition among one another and form local cycles. These self-contained cycles represent local interactions within the decision structure of a single player, while the absence of backward transitions across stages rules out cyclic strategic feedback between the two players. Termination states can be incorporated as absorbing states, namely, single-edge self-loops with arbitrary terminal rewards or costs; see Figure 1.



(a) Acyclic game.

(b) Two-player staged game.

Figure 1: Special cases of TBDFG.

This staged interpretation appears naturally in progression-based games and decision processes, such as Chess and Go. In these games, stages can be viewed as successive decision steps from the initial position to termination. The game progresses forward because each move advances the play to a later decision stage, while terminal outcomes such as checkmate, resignation, draw, or scoring can be represented by absorbing termination states. In Go, for instance, a board position that would repeat an earlier position is prohibited by the ko or superko rule. Equivalently, such a prohibited repetition can be modeled as reaching a terminal invalid-move state, rather than as a transition back to an earlier stage. The simplest such instance in Go is shown in Figure 2 where Black plays at x and captures the white stone at y, producing position T. If White were allowed to immediately play at y, White would capture the black stone at x and return the board to position S. The ko rule prevents this infinite two-player cycle. Under this representation, the game has no mixed-player interaction cycle: repetitions are absorbed by terminal states, and the remaining legal transitions progress forward through the staged game tree. Thus, Chess and Go illustrate the practical relevance of the TBDFG framework, whose essential structure is forward progression together with possible local self-contained cycles.

The success of AlphaGo further highlights the algorithmic importance of exploiting structure in large deterministic games. AlphaGo and AlphaGo Zero combine policy improvement, value estimation, and search to make the enormous game tree of Go computationally tractable Silver et al. (2016; 2017). Although these methods are learning-based and heuristic rather than complexity-theoretic, they demonstrate that structural information and iterative value improvement can be powerful tools for solving large turn-based games. Our work is complementary to this line of research. Instead of designing a practical learning system for Go, we identify a structural subclass of the game that captures the forward-progression principle in Go and other board games, and prove that a simple strategy-iteration method converges in strongly polynomial time for this class.

The forward game setting is naturally suited to broader classes of games and decision processes with hierarchical, staged, or predominantly forward dynamics. In many such models, the state space can be organized by a progress variable, such as time, level, resource depletion, accumulated information, or position in a task hierarchy. Most actions advance the system with respect to this variable, while

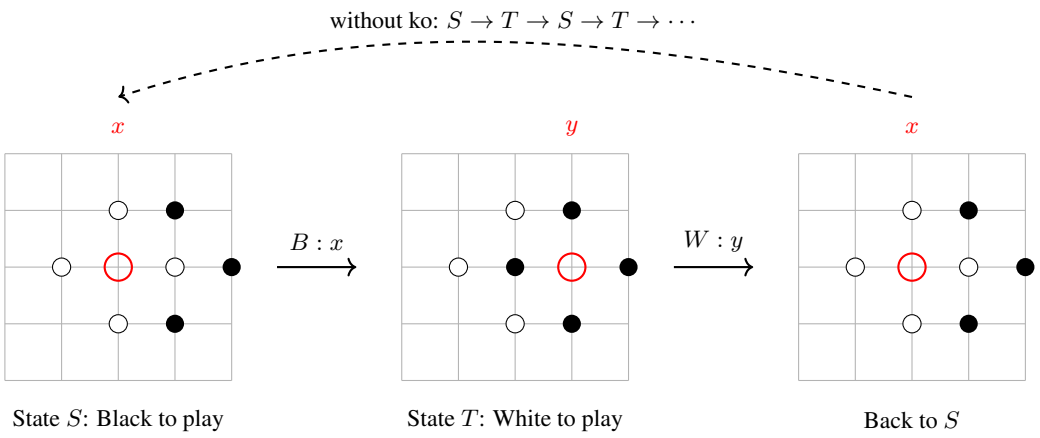
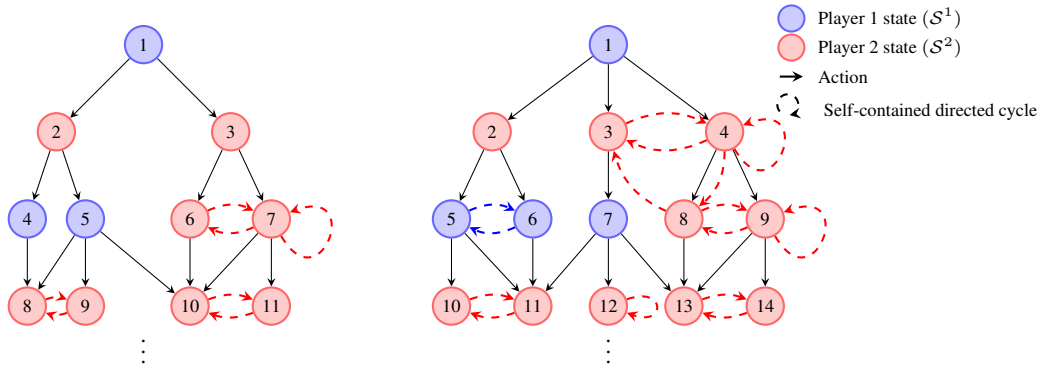


Figure 2: A simplest ko instance in Go.

local loops may occur within a stage when a player repeats a decision, retries an action, waits, refines a subtask, or cycles among internal configurations before the game proceeds. When these local cycles are controlled by a single player, the resulting game satisfies the TBDFG condition. This setting captures deterministic planning problems with adversarial choices, hierarchical game trees with local repetitions, verification and synthesis games with layered progress measures, and restricted graph games in which cycles are confined to one controller.

The assumption is also motivated by graph games, where the transition graph plays a central role in both modeling and complexity analysis. Simple stochastic games Condon (1992); Ludwig (1995), mean-payoff games Ehrenfeucht & Mycielski (1979); Zwick & Paterson (1996), and parity games Emerson & Jutla (1991); Calude et al. (2017) demonstrate how graph structure can strongly influence algorithmic behavior. Mean-payoff games are especially relevant here as turn-based weighted graph games with long-run average objectives. Zwick and Paterson gave a constructive pseudo-polynomial algorithm for such games, computing exact vertex values from sufficiently accurate finite-horizon approximations and then deriving positional optimal strategies Zwick & Paterson (1996). Their result illustrates that even when optimal strategies are memoryless, recurrent weighted graph structure can still be algorithmically delicate. Since cycles are often the main source of long-run strategic complexity, the forward-game condition isolates this difficulty by allowing only player-internal cycles. Consequently, TBDFGs preserve nontrivial recurrent behavior while ruling out directed feedback cycles controlled jointly by both players.

These models can have substantially richer structures than special cases such as acyclic games and two-player staged games, illustrated in Figure 1. Figure 3 presents examples of more general TBDFGs. Figure 3(a) contains cycles in Player 2's states and does not exhibit the alternating stage-like structure typical of board games. Figure 3(b) shows that multiple single-player cycles may coexist and form more complicated recurrent structures. The defining feature of a TBDFG is that every directed cycle remains internal to one player, allowing rich local recurrence while preserving forward progression in the strategic interaction between the two players.



(a) Cycle contained in Player 2's states.

(b) Multiple self-contained cycles in both players.

Figure 3: Examples of general TBDFG.

B Deferred Proofs

B.1 Proof of Lemma 4

Proof. Without loss of generality, we only prove the statement for the first iteration.

First, we show the monotonicity of the value vector. By Lemma 3 and $\tau_0 = \pi^2(\sigma_0)$ (see Input of Algorithm 1), we have $r_a^{\pi_0} \leq 0$ for all $a \in \mathcal{A}^2$ and $r_a^{\pi_0} = 0$ for all $a \in \pi_0$. Moreover, according to the updating rule used in Step 1 of Algorithm 1, we have $r_a^{\pi_0} = \min\{r_{a'}^{\pi_0} : a' \in \mathcal{A}^1\}$ for $a \in \sigma_1 \setminus \sigma_0$. This implies that $r_a^{\pi_0} \leq 0$ for all $a \in \pi_1$. Using this together with Proposition 1(i) and (v), we have

$$\mathbf{v}^{\pi_1} - \mathbf{v}^{\pi_0} = (I - \gamma P^{\pi_1})^{-T} \left(\mathbf{c}^{\pi_1} - (I - \gamma P^{\pi_1})^T \mathbf{v}^{\pi_0}\right) = (I - \gamma P^{\pi_1})^{-T} \mathbf{r}_{\pi_1}^{\pi_0} \leq 0$$

where the first equality follows from Proposition 1(i), the second equality follows from the definition of $\mathbf{r}$ in Proposition 1(v), and the inequality follows from $r_a^{\pi_0} \leq 0$ for all $a \in \sigma_1$ and the fact that P^{π_1} is the probability transition matrix. Consequently, the obtained value function decreases.

Next, we show that $\mathbf{b}^T \mathbf{v}^{\pi_0} > \mathbf{b}^T \mathbf{v}^{\pi_1}$ unless Algorithm 1 terminates. If not, we have $\min\{r_a^{\pi_0} : a \in \mathcal{A}^1\} < 0$. Denoting $a' = \arg\min\{r_a^{\pi_0} : a \in \mathcal{A}^1\}$, we have

$$\mathbf{b}^T \mathbf{v}^{\pi_0} - \mathbf{b}^T \mathbf{v}^{\pi_1} = -(\mathbf{r}_{\pi_1}^{\pi_0})^T \mathbf{x}_+^{\pi_1} = -\sum_{a \in \sigma_1} r_a^{\pi_0} x_a^{\pi_1} - \sum_{s \in \tau_1} r_a^{\pi_0} x_a^{\pi_1} \geq -\sum_{a \in \sigma_1} r_a^{\pi_0} \mathbf{x}_a^{\pi_1} = -r_{a'}^{\pi_0} x_{a'}^{\pi_1} > 0$$

where the first equality follows from Proposition 1(v), the first inequality follows from $\tau_0 = \pi^2(\sigma_0)$ which implies that $r_a^{\pi_0} \leq 0$ for all $a \in \tau_1$, the last equality follows from the updating scheme in Step 1 of Algorithm 1, and the last inequality follows from the fact that $r_{a'}^{\pi_0} < 0$. Hence the result follows. □

B.2 Proof of Lemma 8

Proof. If $k^+ = k$, then $(k, k^+]$ contains no integer, and the proof is complete.

If $k^+ > k$, let $\Delta_k = \min\{r_a^{\pi_k} : a \in \mathcal{A}^1\}$ be the modified cost computed at Step 1 of Algorithm 1 at the $(k+1)$-th iteration, and a' be the action corresponding to this smallest modified cost, which is a path action on π_{k+1} by the definition of k^+. Then, for any $k' \in (k, k^+]$ we have

$$\mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi_{k'}}) \geq \mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi_{k+1}}) = -\left(\mathbf{r}_{\pi_{k+1}}^{\pi_k}\right)^T \mathbf{x}_+^{\pi_{k+1}} \geq -r_{a'}^{\pi_k} x_{a'}^{\pi_{k+1}} \geq (1-\gamma)|\Delta_k|, \quad (9)$$

where the first inequality follows from Lemma 4, the first equality follows from Proposition 1(v), the second equality follows from the definition of a' and the updating scheme (see Step 1 of Algorithm 1), and the last inequality follows from the definition of Δ_k and Proposition 1(iv). Then we have

$$\begin{aligned}
\mathbf{b}^T(\mathbf{v}^{\pi_{k'}} - \mathbf{v}^{\pi_{k^+}}) =& \mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi_{k^+}}) - \mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi_{k'}}) \\
\leq& \mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi_{k^+}}) - (1-\gamma)|\Delta_k| \\
\leq& (1 - 1/n^2)\mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi_{k^+}}),
\end{aligned}$$

where the first equality follows by rearranging the terms, the first inequality follows from (9), and the second inequality follows from $\pi_{k^+} \in \Pi_k$ and Lemma 7. □

B.3 Proof of Lemma 9

Proof. Let $k^+ \geq k$ be any iteration before creating a new Player 1's cycle (or algorithm termination or $k^+ \geq k$), and let k' be any round between k and k^+. By Proposition 1(v), we have $(\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k} = \mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi_{k^+}})$.

We divide the analysis into two cases. First suppose that there exists an action a used in state s on a path in π_k such that $r_a^{\pi_{k^+}} x_a^{\pi_k} \geq (\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k}/n$ (note that $(\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k} \geq 0$). Since a is on a path $x_a^{\pi_k} \leq (1-\gamma)n$, which implies that $(1-\gamma) r_a^{\pi_{k^+}} n^2 \geq (\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k}$. Now if strategy $\pi_{k'}$ uses action a, then

$$\begin{aligned}\mathbf{b}^T(\mathbf{v}^{\pi_{k'}} - \mathbf{v}^{\pi_{k^+}}) &\geq (1-\gamma)\left(v_s^{\pi_{k'}} - v_s^{\pi_{k^+}}\right) = (1-\gamma)\left((c_a + \gamma \mathbf{p}_a^T \mathbf{v}^{\pi_{k'}}) - v_s^{\pi_{k^+}}\right) \\ &\geq (1-\gamma)\left((c_a + \gamma \mathbf{p}_a^T \mathbf{v}^{\pi_{k^+}}) - v_s^{\pi_{k^+}}\right) = (1-\gamma) r_a^{\pi_{k^+}} \geq (\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k}/n^2,\end{aligned}$$

where the first equality follows from Proposition 1(i), the first and second inequalities follow from Lemma 4, the second equality follows from (5), and the last inequality follows from $(1-\gamma) r_a^{\pi_{k^+}} n^2 \geq (\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k}$.

In the second case, there is no action a on a path in π_k satisfying $r_a^{\pi_{k^+}} x_a^{\pi_k} \geq (\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k}/n$. The remaining portion of $(\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k}$ is due to cycles, so there must be some cycle C consisting of actions $\{a_1, \ldots, a_t\}$ and corresponding states $\{s_1, \ldots, s_t\}$ such that $\sum_{a \in C} r_a^{\pi_{k^+}} x_a^{\pi_k} \geq (\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k}/n$. Note that $(\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k} \geq 0$ and it follows from the optimality of the Player 2's strategy in k^+ that $r_a^{\pi_{k^+}} \leq 0$ for all $a \in \mathcal{A}^2$. These together implies that C can only be Player 1's cycle.

All flux in C first enters C either from a path ending at C or from the initial unit of flux placed on some state s in C. If $y_s \geq 1-\gamma$ units of (scaled) flux first enter C at state s in strategy π_k, then that flux incurs $y_s(v_s^{\pi_k} - v_s^{\pi_{k^+}})$ cost with respect to $\mathbf{r}^{\pi_{k^+}}$, so $\sum_{a \in C} r_a^{\pi_{k^+}} x_a^{\pi_k} = \sum_{s \in C} y_s (v_s^{\pi_k} - v_s^{\pi_{k^+}})$. Moreover, each term $v_s^{\pi_k} - v_s^{\pi_{k^+}}$ is nonnegative (see Lemma 4). Under the scaled flux convention $\mathbf{b} = (1-\gamma)\mathbf{e}$, each of the n starting states contributes at most $(1-\gamma)$ units of first-entry mass to any fixed state of the cycle, because the deterministic trajectory can first enter the cycle only once, we have $y_s \leq n(1-\gamma)$. Also note that $\sum_{s \in C}(v_s^{\pi_k} - v_s^{\pi_{k^+}}) = \sum_{a \in C} r_a^{\pi_{k^+}}/(1-\gamma)$. Therefore, $n \sum_{a \in C} r_a^{\pi_{k^+}} \geq \sum_{a \in C} r_a^{\pi_{k^+}} x_a^{\pi_k}$ implying $n^2 \sum_{a \in C} r_a^{\pi_{k^+}} \geq (\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k}$.

As long as cycle C is intact, each $a \in C$ has 1 (scaled) flux from states in C (see Proposition 1(iv)), so if C is in strategy $\pi_{k'}$ then

$$\mathbf{b}^T(\mathbf{v}^{\pi_{k'}} - \mathbf{v}^{\pi_{k^+}}) \geq (1-\gamma) \sum_{s \in C} \left(v_s^{\pi_{k'}} - v_s^{\pi_{k^+}}\right) = \sum_{a \in C} r_a^{\pi_{k^+}} \geq (\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k}/n^2.$$

Now if $\log_{n^2/(n^2-1)} n^2$ iterations occur between π_k and $\pi_{k'}$, Lemma 8 implies that

$$\mathbf{b}^T(\mathbf{v}^{\pi_{k'}} - \mathbf{v}^{\pi_{k^+}}) < \mathbf{b}^T(\mathbf{v}^{\pi_k} - \mathbf{v}^{\pi_{k^+}})/n^2 = (\mathbf{r}^{\pi_{k^+}})^T \mathbf{x}^{\pi_k}/n^2,$$

where the equality follows from Proposition 1(v). In the first case action a cannot appear in $\pi_{k'}$, and in the second case Player 1's cycle C must be broken in $\pi_{k'}$. This takes $\log_{n^2/(n^2-1)} n^2 = O(n^2 \log n)$ iterations if no new cycles interrupt the process. □

B.4 Proof of Lemma 11

Proof. Consider a strategy profile $\pi = (\sigma, \tau)$ where $\tau = \pi^2(\sigma)$ with respect to the optimal gains $\mathbf{r}^{\pi^*}$ where $\pi^* = (\pi_*^1, \pi_*^2)$ (see Proposition 1(vi)).

There is an action such that $r_a^{\pi^*} x_a^{\pi} \geq (\mathbf{r}^{\pi^*})^T \mathbf{x}^{\pi}/n = \mathbf{b}^T(\mathbf{v}^{\pi} - \mathbf{v}^{\pi^*})/n \geq 0$. Notice that $\mathbf{r}^{\pi^*}$ is the gain vector for the optimal strategy, $r_{a'}^{\pi^*} \geq 0$ for all $a' \in \mathcal{A}^1$ and $r_{a'}^{\pi^*} \leq 0$ for all $a' \in \mathcal{A}^2$, thus we have $a \in \mathcal{A}^1$. If a is on a path in π, then $1-\gamma \leq x_a^{\pi} \leq (1-\gamma)n$, so $(1-\gamma) r_a^{\pi^*} \geq (\mathbf{r}^{\pi^*})^T \mathbf{x}^{\pi}/n^2$, and if a is on a cycle in π, then $1 \leq x_a^{\pi} \leq n$, so $r_a^{\pi^*} \geq (\mathbf{r}^{\pi^*})^T \mathbf{x}^{\pi}/n^2$.

If $\pi' = (\sigma', \tau')$ where $\tau' = \pi^2(\sigma')$ is any strategy containing $a \in \mathcal{A}^1$ and a is on a path in π, then we have

$$\mathbf{b}^T(\mathbf{v}^{\pi'} - \mathbf{v}^{\pi^*}) \geq \mathbf{b}^T \left(\mathbf{v}^{(\sigma', \pi_*^2)} - \mathbf{v}^{\pi^*}\right) \geq (\mathbf{r}^{\pi^*})^T \mathbf{x}^{(\sigma', \pi_*^2)}$$
$$= \sum_{a' \in \sigma'} r_{a'}^{\pi^*} x_{a'}^{(\sigma', \pi_*^2)} \geq r_a^{\pi^*} x_a^{(\sigma', \pi_*^2)} \geq (1-\gamma) r_a^{\pi^*} \geq (\mathbf{r}^{\pi^*})^T \mathbf{x}^{\pi} / n^2,$$

where the first inequality follows from the optimality of τ', the second inequality follows from Proposition 1(v), the first equality and the third inequality follow from Lemmas 3 and Proposition 1(iv), the second last inequality follows from $x_a^{\pi} \geq (1-\gamma)$ and the forward progression property of TBDFG, and the last inequality follows from $(1-\gamma) r_a^{\pi^*} \geq (\mathbf{r}^{\pi^*})^T \mathbf{x}^{\pi} / n^2$. Similarly, if π' contains $a \in \mathcal{A}^1$ on a cycle and a is on a cycle in π, we have $\mathbf{b}^T(\mathbf{v}^{\pi'} - \mathbf{v}^{\pi^*}) \geq (\mathbf{r}^{\pi^*})^T \mathbf{x}^{\pi} / n^2$

Now by Lemmas 4 and 6, and there are more than $\log_{n/(n-1)} n^2 = O(n \log n)$ new Player 1's cycles created between policies π and π' then

$$\mathbf{b}^T(\mathbf{v}^{\pi'} - \mathbf{v}^{\pi^*}) < \left(1 - \frac{1}{n}\right)^{\log_{n/(n-1)} n^2} \mathbf{b}^T(\mathbf{v}^{\pi} - \mathbf{v}^{\pi^*}) \leq (\mathbf{r}^{\pi^*})^T \mathbf{x}^{\pi} / n^2,$$

where the first inequality follows from Lemmas 4 and 6, and the equality follows from Proposition 1(v). Therefore, if π contained a on a path, then a cannot appear in any strategy after π' for the remainder of the algorithm. If π contained a on a cycle, then a cannot appear in a cycle after π', although it may still appear on a path later in the algorithm. □