Improved Bounds for Strategy Improvement Algorithms for Energy Games

Dani Dorfman
Max Planck Institute for Informatics, Saarbrücken, Germany

Haim Kaplan
Tel Aviv University, Israel

Uri Zwick
Tel Aviv University, Israel

Abstract

Strategy improvement is a natural and well-studied family of algorithms for solving various classes of stochastic and deterministic graph games. We present an improved upper bound of $O(n2^n)$ on the number of iterations performed by the most natural, and most greedy, variant of the algorithm when applied to n-vertex *Energy Games*. We also obtain a similar upper bound of $O(\mathrm{poly}(n) \cdot 2^n)$ on the expected number of iterations performed by *Random-Edge*, one of the most natural randomized variants of the algorithm. To the best of our knowledge, these are the first bounds for natural strategy-improvement algorithms on non-binary energy games that beat the trivial $n^n = 2^{n \log n}$ bound obtained by enumerating all strategies. The proof is based on a new adaptation of the layering technique of [Dorfman, Kaplan, Zwick, ICALP 2019].

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Graph Games, Energy Games, Strategy improvement

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.140

Funding *Haim Kaplan*: This work was supported in part by Israel Science Foundation grant no. 1156-23 and the Blavatnik Family Foundation.
Uri Zwick: This work was supported in part by Israel Science Foundation grant no. 2854/20.

1 Introduction

2-player Turn-Based Deterministic Games (2TBDG), are zero sum, infinite-duration, turn-based, deterministic games played by two players, usually referred to as Min, or player 0, and Max, or player 1, on a finite directed graph. Each vertex of the graph is controlled by one of the players. Each edge has a real cost associated with it. A token is initially placed at one of the vertices of the graph. The player controlling the vertex in which the token resides chooses an outgoing edge and moves the token along it. (Each vertex is assumed to have at least one outgoing edge.) Play goes on indefinitely producing an infinite sequence $c_0, c_1, \ldots$ composed of the costs of the edges traversed. The outcome of the game is $OBJ(c_0, c_1, \ldots)$, where $OBJ : \mathbb{R}^\omega \to \mathbb{R}$ is the *objective function*. Min wants to minimize the outcome while Max wants to maximize it. The definition can be extended to *2-player Turn-Based Stochastic Games* (2TBSG) in which some of the vertices are randomization vertices. (see, e.g., [9, 21].)

Discounted Payoff Games (DPG) (see, e.g., [44]) use the objective function $DIS_\lambda(c_0, c_1, \ldots) = (1-\lambda)\sum_{i=0}^{\infty} \lambda^i c_i$, where $0 < \lambda < 1$ is a *discount factor*. *Mean Payoff Games* (MPG) (see, e.g., [12, 19, 44]) use the objective function $AVG(c_1, c_2, \ldots) = \liminf_{n\to\infty} \frac{1}{n}\sum_{i=1}^{n} c_i$. *Energy Games* (EG) (see, e.g., [8, 4]) use the objective function $ENG(c_1, c_2, \ldots) = \sup_{n\geq 0} \sum_{i=1}^{n} c_i$. (When $n = 0$ the sum is defined to be 0.)

Parity Games (PG) (see, e.g., [13, 39]), which may be seen as very special cases of MPGs or EGs (see, e.g., [40]), assume that all costs, usually referred to as *priorities*, are in $\{1, 2, \ldots, d\}$ and $PAR(c_0, c_1, \ldots) = (\limsup_{n\to\infty} c_n) \bmod 2$, i.e., $PAR(c_0, c_1, \ldots) = 1$ if and only if the parity of the largest priority that appears infinitely often is odd, and 0 otherwise.

It is well known [12, 40, 15] that all these classes of games, and many more, are *positionally determined*, i.e., have *positional* optimal strategies. (A strategy is positional if it is deterministic and memoryless.) In particular, each vertex has a *value* that both players ensure if they play optimally. (If the value of a vertex is ν, then Min has a strategy that ensures that the outcome of the game will be at most ν, no matter what Max does, and Max has a strategy that ensures that the outcome will be at least ν, no matter what Min does.)

It is a major open problem whether any of the games defined above can be solved in polynomial time.[1] Solving a game means finding the values of all vertices, and sometimes also optimal (positional) strategies for the two players. MPGs and EGs are equivalent under polynomial time reductions. Both can be reduced to DSGs with discount factors arbitrarily close to 1. PGs seem to be much easier than DSGs, MPGs and EGs. It is not difficult to see that decision versions of these problems are in NP $\cap$ co-NP [44], this follows easily from the fact that the 1-player versions of these games are in P [5]. Furthermore, Jurdziński [27] proved that these problems are actually in UP $\cap$ co-UP.

The fastest randomized algorithms for 2TBSGs, and in particular, MPGs, EGs, run in subexponential $2^{\tilde{O}(\sqrt{n})}$ time, where n is the number of vertices in the game graph [34, 3, 20, 23]. These algorithms are based on randomized versions of the simplex algorithm for solving linear programs [29, 36, 30].[2]

MPGs and EGs with integer costs in $\{-M, \ldots, 0, \ldots, M\}$ can be solved in a pseudopolynomial running time of $O(mnM)$, where n is the number of vertices and m is the number of edges [6]. The fastest deterministic algorithm whose running time does not depend on M runs in $O(m2^{n/2})$ time [11, 31].[3]

PGs can be solved much faster in quasipolynomial time [7, 28, 14, 32]. Unfortunately, the techniques used to obtain these quasipolynomial time algorithms for PGs do not seem to apply to MPGs or EGs [16].

One of the most natural algorithms for solving various versions of games is *Strategy Improvement* (SI). When applied to 1-player versions of such games, also known as *Markov Decision Processes* (MDPs), the algorithm is also called *Policy Iteration* and was first proposed by Howard [26]. The algorithm starts with an arbitrary strategy of one of the players and improves it by making local improving changes until no further local improvements are possible. It can then be shown that the final strategy obtained is optimal. Note that this is reminiscent of the simplex algorithm for solving linear programs. One important difference, however, is that the strategy improvement algorithm may perform several improving changes simultaneously, which is usually not possible in the context of general linear programs.

Strategy improvement is actually a family of algorithms parameterized by a *switching rule* that specifies which improving changes, also known as *switches*, to perform in each situation. (In typical situations there are many possible improving switches some of which may be performed simultaneously.) This is analogue to a *pivoting rule* used by the simplex algorithm. The subexponential algorithms mentioned above are actually randomized versions of the strategy improvement algorithm with a switching rule called Random-Facet.[4]

---

1 DPG and stochastic extensions of them can be solved in polynomial time if the discount factor λ is assumed to be a constant. See [43, 21].

2 It is important to note that there are no known ways of solving 2TBSG using linear programming. It is known, however, that solving MPGs is equivalent to solving *tropical* linear programs [1].

3 Some small mistakes in the pseudocode of [11] are corrected in [2].

4 The names Random-Facet and Random-Edge are borrowed from the names of closely related pivoting rules for the simplex algorithm.

Perhaps the most natural switching rule, known as Switch-All, chooses the best improving switch available at each vertex, if any, and performs all these switches simultaneously. In a sense, this is the most *greedy* variant of strategy improvement.[5] It is also known to behave very well in practice [5].

Although Switch-All is probably not the most efficient deterministic variant of the strategy improvement algorithm, we believe that it is important to understand its theoretical behavior. Our main result is an improved upper bound of $O(n2^n)$ on the number of iterations that it performs on any n-vertex energy game. We also obtain a similar upper bound on the expected number of iterations performed when using the Random-Edge switching rule, one of the most natural randomized switching rules. Comparable results were known only for *binary* games, i.e., games in which the outdegree of each vertex is at most 2.

Friedmann [17] obtained a lower bound of $\Omega(2^{cn})$, for some $c > 0$, on the number of iterations performed by the Switch-All algorithm for PGs suggested by [42]. This lower bound can also be adapted to hold for EGs. Thus, our $O(n2^n)$ upper bound is tight up to a constant in the exponent.

Acyclic Unique Sink Orientations (AUSOs) of the Boolean hypercube [41] may be seen as combinatorial abstractions of binary versions, i.e., outdegree 2, of the games we are considering, as well as of other problems.[6] The strategy improvement family of algorithms can also be expressed in this abstract setting. Mansour and Singh [35] obtained an $O(2^n/n)$ upper bound on the number of iterations performed by Switch-All on any AUSO. (see also [25].) Matoušek and Szabó [37] obtained an $\Omega(2^{n/2})$ lower bound on the number of iterations performed in the worst case by Switch-All. Hansen et al. [22] gave an $O(1.7985^n)$ upper bound on the expected number of steps performed by Random-Edge on AUSOs. Hansen and Zwick [24] obtained a $2^{\Omega\left(\sqrt{n\log n}\right)}$ lower bound for Random-Edge. Gartner [18] obtained an $O(\mathrm{e}^{2\sqrt{n}})$ upper bound on the expected number of steps performed by Random-Facet. An almost matching $2^{\Omega(\sqrt{n})}$ was obtained by Matoušek [38].

The strategy improvement algorithm for energy games with the Switch-All switching rule was previously considered by Brim and Chalupka [5]. They showed that the algorithm performs at most $O(mnM)$ steps, when all edge costs are in $\{-M,\ldots,0,\ldots,M\}$. They did not present any bound for energy games with arbitrarily large integer edge costs, let alone arbitrary real edge costs. They also did not consider Random-Edge. Our $O(n2^n)$ and (expected) $O(mn^2\log n\cdot 2^n)$ upper bounds for Switch-All and Random-Edge hold for energy games with arbitrary edge costs.

Our new analysis of strategy improvement algorithms is based on the *layering technique* of Dorfman, Kaplan and Zwick [10] who used it to analyze their deterministic $O(m2^{n/2})$-time algorithm for solving energy games. (Note, however, that their algorithm is not a strategy improvement algorithm.) The technique was also used by Kozachinskiy [31] to obtain a deterministic $O(3.214^n)$-time algorithm for solving DPGs. New ideas are needed to adapt the layering technique to the analysis of strategy improvement algorithms.

On a final note, there is a simple reduction [31] from energy games to bipartite energy games (i.e., $E\subseteq V_1\times V_2\cup V_2\times V_1$). Although this reduction will somewhat simplify part of our proofs, this reduction only preserves the winning vertices but not the values.

---

[5] Performing more locally improving switches does not necessarily imply a larger improvement. Thus, Switch-All is not necessarily the best strategy improvement algorithm.

[6] A non-binary game can be converted into a binary game, but the number of vertices in the new game becomes proportional to the number of edges in the original game. AUSOs can also be considered over non-Boolean “cubes”.

2 Preliminaries

An *energy game* is played on a graph $G = (V_0, V_1, E, c)$, where $V_0 \cap V_1 = \emptyset$. $V = V_0 \cup V_1$ is the set of vertices, $E \subseteq V \times V$ is the set of edges and $c : E \to \mathbb{R}$ is the cost function. It is assumed that every vertex has at least one outgoing edge.

The game is played by two players, Player 0 that controls the vertices in V_0, and Player 1 that controls the vertices in V_1. Starting from an initial vertex $u_0 \in V$, the players construct an infinite path $u_0, u_1, \ldots$. If $u_i \in V_0$, then Player 0 chooses the edge $(u_i, u_{i+1}) \in E$. If $u_i \in V_1$, then Player 1 chooses the edge $(u_i, u_{i+1}) \in E$. Let $c_i = c(u_{i-1}, u_i)$, for $i \geq 1$. The value of this play from u_0 is the smallest $c \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ such that $c \geq \sum_{i=1}^{k} c_i$ for every $k \geq 1$. Equivalently $c = \sup_{k \geq 0} \sum_{i=1}^{k} c_i$ (when $k = 0$ the sum equals 0). Player 0 wants to minimize c while Player 1 wants to maximize it. I.e., Player 0 wants to choose short paths while Player 1 wants to choose long paths. The *value* of the game is a function $val(G) : V \to \mathbb{R}^{\geq 0}$ mapping every vertex to the value of the game starting at that vertex (assuming both players play optimally). For ease of notation we abbreviate $val(G)$ to val.

Let $\Pi_p = \{u_1 u_2 \ldots u_k \in V^* V_p \mid \forall i\ (u_i, u_{i+1}) \in E\}$, for $p = 0, 1$, be the set of finite paths (walks) in G that end in a vertex of V_p. The paths are not required to be simple.

▶ **Definition 2.1** (Strategies, positional strategies). *A* strategy *for player p, where $p = 0, 1$, is a mapping $\sigma : \Pi_p \to E$ such that $\sigma(u_1 \ldots u_k) = (u_k, v) \in E$ for every $u_1 \ldots u_k \in \Pi_p$. A* positional *strategy for Player p is a strategy σ such that $\sigma(u_1 \ldots u_k) = \sigma(u_k)$ for every $u_1 \ldots u_k \in \Pi_p$.*

It is well known [15] that energy games are determined by positional strategies, i.e., each player has a positional strategy that guarantees him at least the optimal value, no matter what the initial vertex is. Therefore throughout this paper we will consider only positional strategies. We denote by Σ_p the set of all positional strategies of player p, where $p = 0, 1$.

▶ **Definition 2.2** ($play^{\sigma_0,\sigma_1}$ and val^{σ_0,σ_1}). *Let $\sigma_0 \in \Sigma_0$ and $\sigma_1 \in \Sigma_1$ be strategies of Players 0 and 1 and let $u_1 \in V$. Let $play^{\sigma_0,\sigma_1}(u_1)$ be the infinite path $u_1 u_2 \ldots$ defined by $(u_k, u_{k+1}) = \sigma_p(u_k)$, if $u_k \in V_p$, for $p = 0, 1$. Let $c_i = c(u_i, u_{i+1})$, for $i \geq 1$. Let $val^{\sigma_0,\sigma_1}(u_1) = \max\{0, \sup_{k \geq 1} \sum_{i=1}^{k} c_i\}$.*

The following lemma follows immediately from Definition 2.2.

▶ **Lemma 2.3.** *Let $\sigma \in \Sigma_0, \pi \in \Sigma_1$. The following holds for every $u \in V$ with $val^{\sigma,\pi}(u) > 0$*

- *If $u \in V_0$ then $val^{\sigma,\pi}(u) = c(u, v) + val^{\sigma,\pi}(v)$, where $\sigma(u) = (u, v)$.*
- *If $u \in V_1$ then $val^{\sigma,\pi}(u) = c(u, v) + val^{\sigma,\pi}(v)$, where $\pi(u) = (u, v)$.*

▶ **Definition 2.4** (f-tight edges). *Let $f : V \to \mathbb{R}^{\geq 0}$. We say that an edge $(u, v) \in E$ is f-*tight *if $c(u, v) - f(u) + f(v) = 0$.*

Given positional strategies $\sigma \in \Sigma_0, \pi \in \Sigma_1$, we define the *induced games* $G(\sigma, \cdot) = (\emptyset, V, E(\sigma, \cdot), c)$ and $G(\cdot, \pi) = (V, \emptyset, E(\cdot, \pi), c)$, where

$$E(\sigma, \cdot) = E \cap \{(u, v) \mid u \in V_1 \vee \sigma(u) = (u, v)\},$$

$$E(\cdot, \pi) = E \cap \{(u, v) \mid u \in V_0 \vee \pi(u) = (u, v)\}.$$

The values of the induced games are denoted $val^{\sigma,\cdot}, val^{\cdot,\pi}$, respectively. The game $G(\sigma, \pi)$ is defined in the obvious way. Clearly, $val(G(\sigma, \pi)) = val^{\sigma,\pi}$.

▶ **Definition 2.5.** *Given a strategy $\pi \in \Sigma_1$, the set $BR(\pi) \subseteq \Sigma_0$ of best-response strategies to π is defined as $BR(\pi) = \{\sigma \in \Sigma_0 \mid \forall \sigma' \in \Sigma_0,\ val^{\sigma,\pi} \leq val^{\sigma',\pi}\}$. (Here, and in what follows, $val^{\sigma,\pi} \leq val^{\sigma',\pi}$ means that $val^{\sigma,\pi}(u) \leq val^{\sigma',\pi}(u)$, for every $u \in V$.)*

▶ Remark. The set $BR(\pi)$ consists of all optimal strategies for player-0 in $G(\cdot, \pi)$. In particular, $BR(\pi) \neq \emptyset$. Also, $val^{\sigma,\pi} = val^{\cdot,\pi}$ for every $\sigma \in BR(\pi)$. Brim and Chalupka [5] show that $val^{\cdot,\pi}$ and a strategy $\sigma \in BR(\pi)$ can be found in polynomial time.

▶ **Lemma 2.6** ([5]). *Let $G = (V_0, \emptyset, E, c)$ be an energy game controlled by player 0. There is an $O(mn + n^2 \log n)$ algorithm that computes $val(G)$ and a positional optimal strategy $\sigma \in \Sigma_0$.*

▶ **Lemma 2.7.** *For any $u \in V$ and strategies $\sigma \in \Sigma_0, \pi \in \Sigma_1$, if $val^{\sigma,\pi}(u) < \infty$, then there exists a vertex $v \in play^{\sigma,\pi}(u)$ such that $val^{\sigma,\pi}(v) = 0$. Moreover, there is a possibly empty path of $val^{\sigma,\pi}$-tight edges from u to v in $G(\sigma, \pi)$.*

Proof. By contradiction, assume that $val^{\sigma,\pi}(v) > 0$ for every $v \in play^{\sigma,\pi}(u)$. Since $play^{\sigma,\pi}(u)$ is an infinite walk, it eventually cycles. That is, $play^{\sigma,\pi}(u) = u_0 ... u_k C^\star$, where $C = u_{k+1} \ldots u_{k+\ell}$. Since $val^{\sigma,\pi}(u) < \infty$ we have that $val^{\sigma,\pi}(u_i) < \infty$ for every $i \geq 0$ and by Lemma 2.3, $val^{\sigma,\pi}(u_i) = c(u_i, u_{i+1}) + val^{\sigma,\pi}(u_{i+1})$ for every $i \geq 0$. Summing these equations for $i = k+1, \ldots, k+\ell$ yields that $c(C) = 0$. A standard argument shows that at least one vertex (the vertex of minimal value) $v \in C$ has $val^{\sigma,\pi}(v) = 0$, a contradiction. (This is essentially the *gasoline puzzle* of Lovász [33, p. 31].)

Therefore, there exists a vertex $v \in play^{\sigma,\pi}(u)$ with $val^{\sigma,\pi}(v) = 0$. Let v be the first such vertex. By Lemma 2.3, every edge on the path from u to v is $val^{\sigma,\pi}$-tight, and this path is contained in $G(\sigma, \pi)$. ◀

The following lemma states that the value function of an Energy Game can be attained by solving a non-linear system of equations. This is the basis of many value-iteration algorithms for Mean-Payoff games and Energy games. It is well known [6] that this system of equations has a minimal solution (coordinate-wise).

▶ **Lemma 2.8** ([6]). *Let $G = (V_0, V_1, E, c)$ be an energy game. Then $val : V \to \mathbb{R}^{\geq 0} \cup \{\infty\}$ is the minimal function satisfying the following system of equations:*

(i) *For every $u \in V_0$, $val(u) = \max(0, \min\{c(u, v) + val(v) \mid (u, v) \in E\})$.*

(ii) *For every $u \in V_1$, $val(u) = \max(0, \max\{c(u, v) + val(v) \mid (u, v) \in E\})$.*

▶ **Corollary 2.9.** *Let $\pi \in \Sigma_1$, and let u be such that $val^{\cdot,\pi}(u) > 0$. Then:*

(i) *If $u \in V_0$ then $val^{\cdot,\pi}(u) \leq c(u, v) + val^{\cdot,\pi}(v)$ for every $(u, v) \in E$.*

(ii) *If $u \in V_1$ then $val^{\cdot,\pi}(u) = c(u, v) + val^{\cdot,\pi}(v)$ where $\pi(u) = (u, v)$.*

Next, we define greedy strategy improvement.

▶ **Definition 2.10.** *Given a strategy $\pi \in \Sigma_1$, we define the set $SI(\pi) \subseteq \Sigma_1$ of greedy strategy improvements for π as follows. A strategy $\pi' \in \Sigma_1$ belongs to $SI(\pi)$ if, for every $u \in V_1$,*

$$\pi'(u) = \begin{cases} \pi(u) & \textit{if } val^{\cdot,\pi}(u) \geq \max\{c(u,v) + val^{\cdot,\pi}(v) \mid (u,v) \in E\}, \\ e \in \arg\max\{c(u,v) + val_{\cdot,\pi}(v) \mid (u,v) \in E\}, & \textit{otherwise.} \end{cases}$$

The following theorem is folklore. For completeness, we include its proof in Appendix A.

▶ **Theorem 2.11.** *Let $\pi \in \Sigma_1, \sigma \in BR(\pi)$ and let $\pi' \in SI(\pi), \sigma' \in BR(\pi')$. It holds that $val^{\sigma,\pi} \le val^{\sigma',\pi'}$. Moreover, if $\pi' \neq \pi$ then there exists a vertex $u \in V_1$ such that $val^{\sigma,\pi}(u) < val^{\sigma',\pi'}(u)$.*

Note that Theorem 2.11 proves the termination of the following strategy improvement algorithm for energy games. Start with an arbitrary strategy $\pi_1 \in \Sigma_1$, compute $\pi_2 \in SI(\pi_1), \pi_3 \in SI(\pi_2), \ldots$, and terminate when $\pi_{i+1} = \pi_i$. Indeed, by Theorem 2.11 $val^{\cdot,\pi_i} \le val^{\cdot,\pi_{i+1}}$ for all i, and $val^{\cdot,\pi_i}(u) < val^{\cdot,\pi_{i+1}}(u)$ for at least one[7] vertex $u \in V$. Since the number of strategies is a finite, the process must terminate. We refer to this algorithm as greedy strategy improvement.

▶ **Lemma 2.12.** *Let $\pi \in \Sigma_1$ and $\pi' \in SI(\pi)$. It holds that $val^{\cdot,\pi'}(u) \ge c(u,v) + val^{\cdot,\pi}(v)$ for every $(u,v) \in E$, where $u \in V_1$.*

Proof. By the definition of $SI(\pi)$, we have that $c(u,v') + val^{\cdot,\pi}(v') \ge c(u,v) + val^{\cdot,\pi}(v)$, where $(u,v') = \pi'(u)$. By Lemma 2.8, applied to $G(\cdot,\pi)$, it holds that

$$val^{\cdot,\pi'}(u) = \max\left(0, c(u,v') + val^{\cdot,\pi'}(v')\right).$$

Therefore

$$val^{\cdot,\pi'}(u) \ge c(u,v') + val^{\cdot,\pi'}(v') \ge c(u,v') + val^{\cdot,\pi}(v') \ge c(u,v) + val^{\cdot,\pi}(v),$$

where the second inequality holds since $val^{\cdot,\pi'}(v') \ge val^{\cdot,\pi}(v')$ by Theorem 2.11. ◀

3 Greedy Strategy Improvement

In this section we use the *layering technique* to obtain key structural properties of the Switch-All algorithm. These properties are used in Section 4 to obtain an $O(n2^n)$ upper bound on the number of iterations that Switch-All performs on any n-vertex energy game.

Consider the sequence of strategies computed during the algorithm $(\sigma_1,\pi_1), (\sigma_2,\pi_2), \ldots, (\sigma_N,\pi_N)$. For convenience, from now on we let $f_i := val^{\sigma_i,\pi_i}$. We define the *anchor* of the ith iteration as $A_i = \{v \in V \mid f_i(v) = 0\}$. By Lemma 2.7, for every $i \ge 1$ and $v \in V$ with $f_i(v) < \infty$, there exists a path of f_i-tight edges from v to A_i in $G(\sigma_i,\pi_i)$. We now introduce the concept of layers, a partitioning of the vertices with finite f_i value. This structural definition is crucial for our analysis.

▶ **Definition 3.1** (Layers). *Let $(\sigma,\pi) \in \Sigma_0 \times \Sigma_1$ and $f = val^{\sigma,\pi}$. Let $A = \{v \mid f(v) = 0\}$. We define the following partitioning of $\{v \mid f(v) < \infty\}$ as follows. The first Layer L_1 is defined as*

$$L_1 = A \cup \{v \in V_1 \mid v \text{ has an } f\text{-tight path of vertices in } V_1 \text{ to } A\}.$$

The next layers are defined recursively as follows. For $j \ge 1$ we let

$$L_{2j} = \{v \in V_0 \mid v \text{ has an } f\text{-tight path of vertices in } V_0 \text{ to } L_{2j-1}\} \setminus L_{\le 2j-1},$$

$$L_{2j+1} = \{v \in V_1 \mid v \text{ has an } f\text{-tight path of vertices in } V_1 \text{ to } L_{2j}\} \setminus L_{\le 2j},$$

where $L_{\le j} := \bigcup_{k \le j} L_k$. For convenience we define $L_0 = A$. (L_0 and L_1 are the only layers that intersect.) We denote by L^i_j the jth layer with respect to $f_i = val^{\sigma_i,\pi_i}$.

[7] It can happen at most once that $val^{\cdot,\pi_i} = val^{\cdot,\pi_{i+1}}$ and $\pi_i \neq \pi_{i+1}$. If this happens then $val^{\cdot,\pi_i}(u) = 0$ for every $u \in V$ such that $\pi_i(u) \neq \pi_{i+1}(u)$. Moreover, because $val^{\cdot,\pi_i} = val^{\cdot,\pi_{i+1}}$, after such an iteration it holds that $\pi_{i+1} = \pi_{i+2}$.

▶ Remark. A basic but important observation is the following. Let $u \in L^i_a, v \in L^i_b$ be such that $a, b > 0$ and $(u, v) \in E$ is f_i-tight, then $a \le b + 1$. Moreover, if a, b belong to the same player then $a \le b$.

▶ **Definition 3.2** (Static layers). *Let $f_i = val^{\sigma_i,\pi_i}$ as above and let L^i_j be the layers as defined in Definition 3.1 with respect to f_i, where $i \ge 2$. We say that layers 0 to $k \ge 0$ are* static *at iteration i if $L^{i-1}_j = L^i_j \ne \emptyset$ for every $0 \le j \le k$ and $f_{i-1}(v) = f_i(v)$ for every $v \in L^i_{\le k}$.*

▶ **Definition 3.3** (Stable layers). *Let $f_i = val^{\sigma_i,\pi_i}$ and let the layers L^i_j be as above. We say that layers 0 to $k \ge 1$ are* stable *at iteration i if layers 0 to $k-1$ are static at iteration i and*
- *If k is even then $\emptyset \ne L^{i-1}_k \subseteq L^i_k$ and $f_{i-1}(v) = f_i(v)$ for every $v \in L^{i-1}_k$.*
- *If k is odd then $L^{i-1}_k \supseteq L^i_k \ne \emptyset$ and $f_{i-1}(v) = f_i(v)$ for every $v \in L^i_k$.*

We define layer 0 to be stable at all iterations. We let $S(i)$ be the maximal k such that layers 0 to k are stable at iteration i. We call $S(i)$ the stability *at iteration i. Note that $S(i) \ge 0$ for all i.*

The following lemma will be a key tool throughout this section.

▶ **Lemma 3.4.** *Let $(u, v) \in E$ and assume $f_{i+1}(u) > 0$. Then*
- *If $u \in V_0$, (u, v) is f_i-tight and $f_i(v) = f_{i+1}(v)$, then $f_i(u) = f_{i+1}(u)$.*
- *If $u \in V_1$, (u, v) is f_{i+1}-tight and $f_{i-1}(v) = f_i(v) = f_{i+1}(v)$, then $f_i(u) = f_{i+1}(u)$.*

Proof. We begin with the first statement. Let $u \in V_0$ and $(u, v) \in E$ such that (u, v) is f_i-tight and $f_i(v) = f_{i+1}(v)$. By Corollary 2.9, we get that

$$f_{i+1}(u) \le c(u, v) + f_{i+1}(v) = c(u, v) + f_i(v) = f_i(u),$$

where the last equality holds since (u, v) is f_i-tight. By Theorem 2.11, it holds that $f_{i+1}(u) \ge f_i(u)$ and therefore $f_{i+1}(u) = f_i(u)$.

For the second statement, Let $u \in V_1$ and $(u, v) \in E$ such that (u, v) is f_{i+1}-tight and $f_{i-1}(v) = f_i(v) = f_{i+1}(v)$. By Lemma 2.12, we get that

$$f_i(u) \ge c(u, v) + f_{i-1}(v) = c(u, v) + f_{i+1}(v) = f_{i+1}(u),$$

where the last equality holds since (u, v) is f_{i+1}-tight. By Theorem 2.11 we conclude that $f_{i+1}(u) = f_i(u)$. ◀

▶ **Lemma 3.5.** *Assume that layers 0 to k are stable at iteration i. Let ℓ be minimal such that $L^{i+1}_\ell \ne L^i_\ell$.*[8] *Then, layers 0 to $\min\{k, \ell\}$ are stable at iteration $i + 1$.*

Proof. We prove by induction on $1 \le j \le \min\{k, \ell\}$ that:
- If j is even and $v \in L^i_j$, then $f_{i+1}(v) = f_i(v)$ and $v \in L^{i+1}_j$.
- If j is odd and $v \in L^{i+1}_j$, then $f_{i+1}(v) = f_i(v)$ and $v \in L^i_j$.

We start with the base case $j = 1$. Recall that the first layer also contains the anchor and therefore might have vertices from V_0. Let $v \in L^{i+1}_1$. If $v \in V_0$ then $v \in A_{i+1}$, so $f_{i+1}(v) = 0$. By Theorem 2.11, we get that $f_i(v) = 0$ and therefore $v \in L^i_j$. The analysis of the case $v \in V_1$ is identical to the inductive step and therefore omitted.

We now move to the inductive step. Let $j \ge 2$, we divide the proof into two cases as follows.

[8] We define $\ell = \infty$ if $L^{i+1}_j = L^i_j$, for all j. In Figure 1 we give an example in which this can happen.

Case 1: j is even. Thus $v \in V_0$. By the definition of layer L^i_j, v has an f_i-tight path $P = v_0(= v), v_1, ..., v_r$, where $v_0, v_1, \dots, v_{r-1} \in L^i_j$, to $v_r \in L^i_{j-1} = L^{i+1}_{j-1}$.

Since P is f_i-tight, we get $f_i(v_0) = \sum_{i=0}^{r-1} c(v_i, v_{i+1}) + f_i(v_r)$. By Corollary 2.9, with $\pi = \pi_{i+1}$, it holds that $f_{i+1}(v_i) \leq c(v_i, v_{i+1}) + f_{i+1}(v_{i+1})$ for every $0 \leq i \leq r-1$. Therefore

$$f_{i+1}(v_0) \leq \sum_{i=0}^{r-1} c(v_i, v_{i+1}) + f_{i+1}(v_r) = \sum_{i=0}^{r-1} c(v_i, v_{i+1}) + f_i(v_r) = f_i(v_0), \tag{1}$$

where the first equality holds since $v_r \in L^i_{j-1}$ (and by the main induction hypothesis on j). Hence, by Theorem 2.11 ($f_{i+1} \geq f_i$), $f_{i+1}(v_0) = f_i(v_0)$. Thus, all the inequalities in Equation 1 are equalities, so P is f_{i+1}-tight and therefore $v_0 \in L^{i+1}_j$.

Case 2: j is odd. Thus $v \in V_1$. By the definition of layer L^{i+1}_j, v has an f_{i+1}-tight path $P = v_0(= v), v_1, ..., v_r$, where $v_0, v_1, \dots, v_{r-1} \in L^{i+1}_j$, to $v_r \in L^{i+1}_{j-1}$. We prove that for every $s < r$, it holds that $v_s \in L^{i-1}_j \cap L^i_j \cap L^{i+1}_j$ and $f_{i+1}(v_s) = f_{i-1}(v_s)$. The case $s = 0$ finishes the proof of the lemma. The proof is by a backward induction on $s = r-1, r-2, \dots, 1$.

We begin by showing that $f_{i+1}(v_{r-1}) = f_i(v_{r-1})$. The edge $(v_{r-1}, v_r) \in E$ is f_{i+1}-tight. Since $v_r \in L^{i+1}_{j-1}$, then by the main induction hypothesis on j, it holds that $f_{i+1}(v_r) = f_i(v_r)$. Moreover, since $j-1 < \min\{k, \ell\}$ we have that $f_i(v_r) = f_{i-1}(v_r)$. By Lemma 3.4, we get that $f_{i+1}(v_{r-1}) = f_i(v_{r-1})$.

Thus, (v_{r-1}, v_r) is also f_i-tight, and therefore $v_{r-1} \in L^i_j$. By stability, $L^i_j \subseteq L^{i-1}_j$ and $f_i(v_{r-1}) = f_{i-1}(v_{r-1})$. We conclude that $v_{r-1} \in L^{i-1}_j \cap L^i_j \cap L^{i+1}_j$ and $f_{i+1}(v_{r-1}) = f_{i-1}(v_{r-1})$.

The proof of the induction step is similar. We use the inductive assumption $f_{i+1}(v_{s+1}) = f_{i-1}(v_{s+1})$ and apply Lemma 3.4. ◀

As a direct Corollary we get

▶ **Corollary 3.6.** *If $k := S(i+1) < S(i)$ then $L^{i+1}_k \neq L^i_k$.*

Proof. By contradiction, assume $L^{i+1}_k = L^i_k$, thus the minimal ℓ for which $L^{i+1}_\ell \neq L^i_\ell$ satisfies $\ell > k$. By Lemma 3.5 we get $S(i+1) > k$, a contradiction. ◀

The next two lemmas establish a connection between the behavior of layers 0 to k and the next layer L_{k+1}. They state that if the kth layer is static for sufficiently many iterations, then the $(k+1)$th layer becomes stable.

The next lemma states that even layers stabilize immediately. Intuitively, the reason for this is that player 0 always plays according to a best response to the strategy of player 1. Hence, as long as player 1 does not change his strategy at layers 0 to $2k$, neither does player 0.

▶ **Lemma 3.7.** *If layers 0 to $2k-1$ are static at iteration $i+1$, then layers 0 to $2k$ are stable at iteration $i+1$.*

Proof. Let $v \in L^i_{2k}$. By the definition of the layers, v has an f_i-tight path $P = v_0(= v), v_1, ..., v_r$, containing vertices of L^i_{2k} to $v_r \in L^i_{2k-1}$. As shown in the proof of Lemma 3.5, $f_{i+1}(v_s) = f_i(v_s)$ for every $s \leq r$ and all edges on this path are also f_{i+1}-tight. We conclude that $v_s \in L^{i+1}_{2k}$ for all $s < r$. Hence, $f_{i+1}(v) = f_i(v)$ and $v \in L^{i+1}_{2k}$. ◀

The following lemma states that if layers 0 to $2k$ remain static for sufficiently many consecutive iterations (see Example C.1 in Appendix C), then the next odd layer L_{2k+1} can only lose vertices and the f-values of its vertices will not change, i.e., layer $2k+1$ is stable.

▶ **Lemma 3.8.** *If layers* 0 *to* $2k$ *are static at iterations* [9] $i+1,\dots,i+n$*, then layers* 0 *to* $2k+1$ *are stable at iteration* $i+n$*.*

Proof. Let $v \in L^{i+n}_{2k+1}$. By the definition of the layers, v has an f_{i+n}-tight path[10] $P = v_r(= v), v_{r-1}, ..., v_0$, containing vertices of L^{i+n}_{2k+1}, to $v_0 \in L^{i+n}_{2k}$. Note that $r < n$. By the settings of the lemma, $L^{i+t}_{\le 2k} = L^{i+n}_{\le 2k}$ for every $0 \le t \le n$, and $f_{i+n}(v_0) = f_i(v_0)$ (since $v_0 \in L^{i+n}_{\le 2k}$).

We show by induction on $0 \le s \le r$ that $f_{i+s}(v_s) = f_{i+n}(v_s)$. The base case $s = 0$ was mentioned above. Let $s > 0$, by the inductive assumption $f_{i+s-1}(v_{s-1}) = f_{i+n}(v_{s-1})$. Consider the edge (v_s, v_{s-1}), this edge is f_{i+n}-tight. Therefore, by Lemma 3.4, $f_{i+s}(v_s) = f_{i+n}(v_s)$, concluding the inductive step.

Since $r < n$ and $f_{i+s}(v_{s-1}) = f_{i+n}(v_{s-1})$ for every $0 \le s \le r$, we get that $f_{i+n-1}(v_s) = f_{i+n}(v_s)$ for every $0 \le s \le r$. Thus, P is also f_{i+n-1}-tight, so $v \in L^{i+n-1}_{2k+1}$. The case $s = r$ also gives us $f_{i+n-1}(v) = f_{i+n}(v)$, thus, layers 0 to $2k+1$ are stable at iteration $i+n$. ◀

In particular, if layers 0 to $2k$ are static for n iterations then at the last iteration it is guaranteed that layers 0 to $2k+1$ are stable.

To summarize the results of this section, we analyzed the behaviour of $S(i)$. In Corollary 3.6 we showed that when $S(i)$ decreases then an appropriate stable layer shrinks/expands. In Lemmas 3.7 and 3.8 we showed that $S(i)$ should increase if it remained unchanged for several iterations. Thus, in a span of n strategy improvements, either a stable layer shrinks/-expands or the stability $S(i)$ increases.

4 Bounding the number of iterations

We prove Theorem 4.1 which is our main theorem.

▶ **Theorem 4.1.** *Let* $(\sigma_1, \pi_1), (\sigma_2, \pi_2), ..., (\sigma_N, \pi_N)$ *be the sequence of strategy pairs computed by greedy strategy improvement, then* $N = O(n2^n)$*.*

We carefully define a binary counter and show that it increases every n iterations of greedy strategy improvement. Let $k = S(i)$. We define the *stable binary counter*, denoted as b^i, as follows

$$b^i = \begin{cases} \underbrace{0.....0}_{|L^i_0|}1\,\underbrace{0.....0}_{|L^i_1 \setminus L^i_0|}\,\underbrace{1.....1}_{|L^i_2|}.....\underbrace{1.....1}_{|L^i_k|}\;\;\underbrace{00.........01}_{n+1-\left|\bigcup_{j\le k} L^i_j\right|} & k \text{ is even} \\ \underbrace{0.....0}_{|L^i_0|}1\,\underbrace{0.....0}_{|L^i_1 \setminus L^i_0|}\,\underbrace{1.....1}_{|L^i_2|}.....\underbrace{0.....0}_{|L^i_k|}\;\;\underbrace{10.........00}_{n+1-\left|\bigcup_{j\le k} L^i_j\right|} & k \text{ is odd.} \end{cases}$$

That is, except for layer 0, even layers correspond to a sequence of 1s and odd layers correspond to a sequence of 0s. We denote the last $n+1-\left|\bigcup_{j\le k} L^i_j\right|$ bits as the *padding* bits. Note that the padding depends on the parity of k.

▶ **Lemma 4.2.** *For every* $i \ge 1$, $b^{i+1} \ge b^i$*. Moreover,* $b^{i+1} = b^i$ *if and only if* $S(i+1) = S(i) := k$ *and layers* 0 *to* k *are static at iteration* $i+1$*.*

[9] A tighter bound would be to say: at iterations $i+1,\dots,i+a$, where $a = |V_1| - |V_1 \cap L^i_{\le 2k}|$.

[10] Note the change in indices, $v_r = v$ is the first vertex of the path.

Proof. Let $k_i = S(i), k_{i+1} = S(i+1)$. We divide the proof into three cases as follows.

Case $k_{i+1} > k_i$. This means that layers 0 to k_i are static at iteration $i+1$ and therefore the most-significant bits of b^i, b^{i+1}, corresponding to layers 0 to k_i, are the same. Thus, it suffices to prove that the suffix of b^{i+1} is larger than the padding of b^i. If k_i is even then the padding of b^i equals 1 (as a binary number). Since the length of the binary counter is $n+1$, then every suffix of the binary counter that has all the bits corresponding to at least one layer (and the padding bits) has a value of at least 2. It follows that the suffix of b^{i+1} that start at layer k_i+1 has a value of at least 2. Assume k_i is odd. Then the padding of b^i equals $10\cdots0$. Since $k_{i+1} > k_i$, it holds that $|L^{i+1}_{k_i+1}| \geq 1$. Thus, the suffix of b^{i+1} also starts with 1 as the most significant bit (actually $|L^{i+1}_{k_i+1}|$ many 1s) but then it is followed by the padding which is at least 1. Hence $b^{i+1} > b^i$.

Case $k_{i+1} = k_i$. Let $k = k_i$. If $L^{i+1}_k = L^i_k$ then $b^{i+1} = b^i$ and layers 0 to k are static in iteration $i+1$. Assume $L^{i+1}_k \neq L^i_k$. If k is odd then $L^{i+1}_k \subsetneq L^i_k$ then $b^{i+1} > b^i$, since the padding became longer. If k is even then $L^i_k \subsetneq L^{i+1}_k$ then again $b^{i+1} > b^i$, since there are now more leading bits of 1 before the padding.

Case $k_{i+1} < k_i$. By Corollary 3.6 we get that $L^{i+1}_{k_{i+1}} \neq L^i_{k_{i+1}}$. As in the previous case, this means $b^{i+1} > b^i$. ◀

We are ready to prove Theorem 4.1.

Proof of Theorem 4.1. Since b^i is a binary number with $n+1$ digits, it can increase at most $O(2^n)$ times. By Lemma 4.2, $b^{i+1} = b^i$ if and only if $S(i+1) = S(i)$ and layers 0 to $S(i)$ are static at iteration $i+1$. By Lemmas 3.7 and 3.8 this can happen for at most n consecutive iterations. Thus, at least once in every n consecutive iterations, b^i strictly increases, resulting in our stated upper bound of $O(n \cdot 2^n)$ iterations for greedy strategy improvement. ◀

5 Random-Edge

Random-Edge is perhaps the most natural choice for a random strategy improvement algorithm. Given a strategy $\pi \in \Sigma_1$ (to which player 0 chooses an optimal counter strategy $\sigma \in BR(\pi)$), Random-Edge considers the set $E_I = \{(u,v) \in E \mid u \in V_1 \wedge c(u,v) + val^{\cdot,\pi}(v) > val^{\cdot,\pi}(u)\}$ of improving switches. Then a random edge $(u,v) \in E_I$ is chosen and the new strategy for player 1 is defined as π', where $\pi'(w) = (u,v)$ if $w = u$ and $\pi'(w) = \pi(w)$, otherwise. A similar theorem to Theorem 2.11 shows that this strategy improvement algorithm terminates. We say that an edge $(u,v) \in E$ is f-valid (or just valid if f is clear from context) if $c(u,v) - f(u) + f(v) \leq 0$. In particular, an edge $e \in E(V_1, V)$ is an improving switch if it is not $val^{\cdot,\pi}(u)$-valid (invalid).

▶ **Theorem 5.1.** *Let $(\sigma_1,\pi_1),(\sigma_2,\pi_2),...,(\sigma_N,\pi_N)$ be the sequence of strategies computed during Random-Edge, then $\mathbb{E}[N] = O(mn^2 \log n \cdot 2^n)$.*

In our analysis (and hence also definition) of stable/static layers, we heavily relied on Lemma 2.12, i.e., that if $u \in V_1$ has an improving switch (u,v) then greedy strategy improvement will increase the potential of u enough to make this edge tight (as long as $f(v)$ does not change). This does not hold anymore when using Random-Edge since only one vertex in V_1 switches its strategy. Therefore we need to strengthen the definition of stability in order to achieve the same structural properties we had (to the current case in which only a single edge is switched at each iteration). Let E^i_I be the set of improving switches with respect to f_i (i.e., the set of outgoing edges of V_1 that are invalid with respect to f_i).

We now define *strongly static* layers and *strongly stable* layers. Layers 0 to k are strongly static (stable) if they are static (stable) and there are no improving switches (for vertices in V_1) that can change the structure of those layers because of the improving switch. This means that no vertex in $V_1 \cap L_{\leq k}$ has an improving switch and that no vertex of $V_1 \backslash L_{\leq k}$ has an improving switch that connects it to $L_{\leq k}$. In the case of strong stability, when k is odd, we allow vertices in the last layer to have improving switches to $V \setminus L_{\leq k}$. (We allow this because odd layers are supposed to shrink.) We get the following definitions.

▶ **Definition 5.2** (Strongly static layers). *Layers* 0 *to* $k \geq 0$ *are* strongly static *at iteration* i *if they are static at iteration* i *and*

- *If* k *is even then* $E_I^{i-1} \cap \big(E(L_{\leq k-1}^i \cap V_1, V) \cup E(V_1, L_{\leq k-1}^i)\big) = \emptyset$.
- *If* k *is odd then* $E_I^{i-1} \cap \big(E(L_{\leq k}^i \cap V_1, V) \cup E(V_1, L_{\leq k}^i)\big) = \emptyset$.

▶ **Definition 5.3** (Strongly stable layers). *Layers* 0 *to* $k \geq 1$ *are* strongly stable *at iteration* i *if they are stable at iteration* i*, layers* 0 *to* $k-1$ *are strongly static at iteration* i*, and if* k *is odd then* $E_I^{i-1} \cap \big(E(L_k^i, L_{\leq k}^i) \cup E(V_1, L_k^i)\big) = \emptyset$.

We define $S'(i)$ with respect to the strongly stable layers similarly to $S(i)$. We also define the strongly stable binary counter b_s^i with respect to $S'(i)$, similarly to b^i.

Lemmas 3.5 and 3.7 transfer immediately to the strong stability settings. Their proofs are almost identical to the original ones, hence deferred to Appendix B and are regarded as Lemmas 3.5' and 3.7'.

Unlike Corollary 3.6 for greedy strategy improvement, if $k := S'(i+1) < S'(i)$, we cannot claim that the kth layer changed. The following lemma states that if k is even then the claim holds. For odd values of k, the kth layer can remain the same (this also means that b_s decreases!) and therefore we need a more delicate analysis which is given in Lemmas 5.5 and 5.6.

▶ **Lemma 5.4.** *Assume* $k := S'(i+1) < S'(i)$*. If* k *is even then* $L_k^i \subsetneq L_k^{i+1}$.

Proof. By contradiction, assume $L_k^i = L_k^{i+1}$. Therefore layers 0 to k are strongly static at iteration $i+1$. We show that $S'(i+1) \geq k+1$ (a contradiction). Let $v \in L_{k+1}^{i+1}$ and let $P = v_0(=v), v_1, ..., v_r$ be an f_{i+1}-tight path, where $v_0, \ldots, v_{r-1} \in L_{k+1}^{i+1}$, and $v_r \in L_k^{i+1}$. We prove by backward induction on $s < r$ that $v_s \in L_{k+1}^i (\subseteq L_{k+1}^{i-1}$ by stability) and $f_{i+1}(v_s) = f_{i-1}(v_s)$. We only prove the base case $s = r-1$ as the inductive step is similar. Since $S'(i) > S'(i+1) = k$, we get that $v_r \in L_k^{i-1}$ and $f_{i+1}(v_r) = f_{i-1}(v_r)$. We get

$$0 = c(v_{r-1}, v_r) + f_{i+1}(v_r) - f_{i+1}(v_{r-1}) \leq c(v_{r-1}, v_r) + f_{i-1}(v_r) - f_{i-1}(v_{r-1}).$$

By strong stability at iteration i, (v_{r-1}, v_r) is f_{i-1}-valid and therefore, by the last inequality, it is also tight. Hence $f_{i+1}(v_{r-1}) = f_{i-1}(v_{r-1})$. Thus, (v_{r-1}, v_r) was also tight at iteration i and therefore $v_{r-1} \in L_{k+1}^i$. Since $S'(i) > k$, we get $v_{r-1} \in L_{k+1}^{i-1}$, ending the inductive proof.

We have shown that layers 0 to $k+1$ are stable. It remains to prove that at iteration i there are no improving switches in $E(L_{k+1}^{i+1}, L_{\leq k+1}^{i+1}) \cup E(V_1, L_{k+1}^{i+1})$. By our previous observations, $f_{i+1}(v) = f_{i-1}(v)$ for every $v \in L_{\leq k+1}^{i+1}$. Therefore if one of these edges is an improving switch at iteration i, then it was also an improving switch at iteration $i-1$ (a contradiction to $S'(i) > k$). ◀

The following lemma addresses the case in which layers 0 to k, where k is odd, are static and strongly stable, but not strongly static. The lemma states that in this case, S' cannot increase in the next iteration. Note that this cannot occur with greedy strategy improvement (Lemma 3.7), since an even layer stabilizes immediately when the previous layer becomes static.

▶ **Lemma 5.5.** *Assume $k := S'(i)$ is odd and $L_k^i = L_k^{i-1}$. Then $S'(i+1) \leq k$.*

Proof. Note that layers 0 to k are not strongly static at iteration i since otherwise, by Lemma 3.7', we get $S'(i) \geq k+1$. By contradiction, assume $S'(i+1) > k$.

Since $S'(i) = k$ and $L_k^i = L_k^{i-1}$, then by strong stability (and the fact that the layers are not strongly static), there is an edge $e \in E(L_k^{i-1}, V \setminus L_{\leq k}^{i-1})$ which is an improving switch with respect to f_{i-1}. Thus, e is also an improving switch with respect to f_i (f did not change inside $L_{\leq k}^{i-1}$ during these iterations). This contradicts the assumption $S'(i+1) > k$, since the first k layers must be strongly static at iteration $i+1$. ◀

▶ Remark. As seen in the last proof, as long as the last layer does not change and $e = (u, v)$ remains an improving switch, S' cannot increase. Hence, an improving switch at u is crucial in order to make progress. Luckily, in Random-Edge, this event has high probability of happening over a span of $O(m \log n)$ strategy improvements.

▶ Remark. Note that unlike the proof of Theorem 4.1, b_s can decrease when $S'(i+1) < S'(i)$ and $k := S'(i+1)$ is odd and $L_k^{i+1} = L_k^i$. (as stated in the previous remark this happens since there is a vertex in L_k^i with an improving switch to $V \setminus L_{\leq k}^i$ which Random-Edge did not choose yet.) The following lemma states that the next time b_s does increase, it will be larger than b_s^i (i.e., highest so far).

▶ **Lemma 5.6.** *Assume $S'(i+1) := k < S'(i)$ for an odd k. Let $k_j = S'(i+j)$ for $j \geq 1$. Let $\ell \geq 1$ be minimal such that either k_ℓ is even or $L_{k_\ell}^{i+\ell} \neq L_{k_\ell}^{i+\ell-1}$. Then $b_s^{i+\ell} > b_s^i$*

Proof. Assume k_ℓ is odd (and that $L_{k_\ell}^{i+\ell} \neq L_{k_\ell}^{i+\ell-1}$). By the assumption, for every $1 \leq s < \ell$, k_s is odd and $L_{k_s}^{i+s} = L_{k_s}^{i+s-1}$. By Lemma 5.5 this means that $k_1, \ldots, k_\ell$ is monotone non-increasing. Therefore, b_s^i and $b_s^{i+\ell}$ share the most significant bits corresponding to the layers $L_{<k_\ell}^i = L_{<k_\ell}^{i+\ell}$. Since $L_{k_\ell}^{i+\ell} \neq L_{k_\ell}^{i+\ell-1}$ (and by stability this means $L_{k_\ell}^{i+\ell} \subsetneq L_{k_\ell}^{i+\ell-1}$), we get that $b_s^{i+\ell} > b_s^i$.

Assume k_ℓ is even. By the assumption, $k_{\ell-1}$ is odd. By Lemma 5.5 it holds that $k_\ell \leq k_{\ell-1}$, so $k_\ell < k_{\ell-1}$ (different parity). Hence, by Lemma 5.4 it holds that $L_{k_\ell}^{\ell-1} \subsetneq L_{k_\ell}^{\ell}$. By stability, during iterations $i, \ldots i+\ell$, the first $k_\ell - 1$ layers did not change and therefore $b_s^{i+\ell} > b_s^i$. ◀

Recall the proof of Lemma 3.8. We considered a tight path of length r to L_{2k}^{i+a} at iteration $i+a$. We proved by induction that after t iterations, the last t vertices on this path belong to L_{2k+1} (i.e., each iteration guarantees that another vertex on this path "stabilized" in L_{2k+1}). In the case of Random-Edge we cannot give this deterministic guarantee since only one random edge is switched at a time. However, since each improving switch might be chosen with probability $\Omega(\frac{1}{m})$, we can say that after $\Theta(tm \log n)$ iterations, with high probability, the last t vertices on the path do belong to L_{2k+1}. This observation is summarized in following lemma, whose proof, due to space constraints, is deferred to Appendix B.

▶ **Lemma 5.7.** *Assume $S'(i+j) = k$ and $L_k^{i+j} = L_k^i$, for every $0 < j \leq \ell$. If k is even then $\ell \leq 3mn \ln n$ with probability at least $\frac{1}{2}$.*

▶ **Corollary 5.8.** *Assume $S'(i+j) = k$ and $L_k^{i+j} = L_k^i$, for every $0 < j \leq \ell$. If k is even then $\mathbb{E}[\ell] = O(mn \log n)$.*

▶ **Lemma 5.9.** *Assume $S'(i+j) = k$, for every $0 < j \leq \ell$. If k is odd then $\mathbb{E}[\ell] = O(mn \log n)$.*

Proof. Same proof as for Lemma 5.7. After $\Omega(mn \log n)$ iterations the first k layers should be strongly static and by Lemma 3.7', after another iteration $S' \geq k+1$. ◀

We are ready to prove Theorem 5.1.

Proof of Theorem 5.1. We show that every $\Omega(mn^2 \log n)$ iterations, b_s is expected to increase and become higher than all of its previous values at least once.

Consider $\Omega(mn^2 \log n)$ consecutive iterations. As in the proof of Theorem 4.1 and by Lemma 5.6, whenever b_s increases, it becomes higher than all of its previous values. Therefore if S' increases during these iterations then we are done. By Corollary 5.8 and Lemma 5.9, b_s remains unchanged for at most $O(mn \log n)$ iterations in expectation. As stated in the Remark before Lemma 5.6, the only way for b_s to decrease is when S' decreases to an odd value and the last layer did not change. Therefore b_s can decrease at most n times without increasing, resulting in the $O(mn^2 \log n \cdot 2^n)$ bound. ◀

6 Concluding remarks and open problems

We have presented an $O(n2^n)$ bound on the number of iterations Switch-All performs on energy games and an $O(mn^2 \log n \cdot 2^n)$ expected bound for Random-Edge. To the best of our knowledge, this is the first exponential bound of the form 2^n, rather than the total number of strategies which is n^n, for non-binary EGs or DPGs.

Since the *layering technique* of Dorfman, Kaplan and Zwick [10] also applies for the result of Kozachinskiy [31] for DPGs, a natural question is whether Switch-All and Random-Edge have exponential bounds for DPGs also.

Another interesting direction for future research is to incorporate the binary-counter progress measure, which can be computed in polynomial time, into the design of a faster algorithm for solving Energy Games, or more generally, any class of games that admits a similar layered structure.

References

1 Marianne Akian, Stéphane Gaubert, and Alexander E. Guterman. Tropical polyhedra are equivalent to mean payoff games. *Int. J. Algebra Comput.*, 22(1), 2012. doi:10.1142/S0218196711006674.

2 Peter Austin and Daniele Dell'Erba. Errata to: "faster deterministic exponential time algorithm for energy games and mean payoff games". *CoRR*, abs/2310.04130, 2023. doi:10.48550/arXiv.2310.04130.

3 Henrik Björklund and Sergei G. Vorobyov. A combinatorial strongly subexponential strategy improvement algorithm for mean payoff games. *Discrete Applied Mathematics*, 155(2):210–229, 2007. doi:10.1016/J.DAM.2006.04.029.

4 Patricia Bouyer, Uli Fahrenberg, Kim G Larsen, Nicolas Markey, and Jiří Srba. Infinite runs in weighted timed automata with energy constraints. In *International Conference on Formal Modeling and Analysis of Timed Systems*, pages 33–47. Springer, 2008.

5 Lubos Brim and Jakub Chaloupka. Using strategy improvement to stay alive. *Int. J. Found. Comput. Sci.*, 23(3):585–608, 2012. doi:10.1142/S0129054112400291.

6 Lubos Brim, Jakub Chaloupka, Laurent Doyen, Raffaella Gentilini, and Jean-François Raskin. Faster algorithms for mean-payoff games. *Formal methods in system design*, 38(2):97–118, 2011. doi:10.1007/S10703-010-0105-X.

7 Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasi-polynomial time. *SIAM J. Comput.*, 51(2):17–152, 2022. doi:10.1137/17M1145288.

8 Arindam Chakrabarti, Luca De Alfaro, Thomas A Henzinger, and Mariëlle Stoelinga. Resource interfaces. In *International Workshop on Embedded Software*, pages 117–133. Springer, 2003. doi:10.1007/978-3-540-45212-6_9.

9 Anne Condon. The complexity of stochastic games. *Inf. Comput.*, 96(2):203–224, 1992. doi:10.1016/0890-5401(92)90048-K.

10 Dani Dorfman, Haim Kaplan, and Uri Zwick. A Faster Deterministic Exponential Time Algorithm for Energy Games and Mean Payoff Games. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 114:1–114:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.ICALP.2019.114.

11 Dani Dorfman, Haim Kaplan, and Uri Zwick. A faster deterministic exponential time algorithm for Energy games and mean payoff games. https://danidorfman.com/publication/energy-games/energy-games.pdf, 2019.

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

13 E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*, pages 368–377. IEEE Computer Society, 1991. doi:10.1109/SFCS.1991.185392.

14 John Fearnley, Sanjay Jain, Bart de Keijzer, Sven Schewe, Frank Stephan, and Dominik Wojtczak. An ordered approach to solving parity games in quasi-polynomial time and quasi-linear space. *Int. J. Softw. Tools Technol. Transf.*, 21(3):325–349, 2019. doi:10.1007/S10009-019-00509-3.

15 Nathanaël Fijalkow, Nathalie Bertrand, Patricia Bouyer-Decitre, Romain Brenguier, Arnaud Carayol, John Fearnley, Hugo Gimbert, Florian Horn, Rasmus Ibsen-Jensen, Nicolas Markey, Benjamin Monmege, Petr Novotný, Mickael Randour, Ocan Sankur, Sylvain Schmitz, Olivier Serre, and Mateusz Skomra. Games on graphs. *CoRR*, abs/2305.10546, 2023. doi:10.48550/arXiv.2305.10546.

16 Nathanaël Fijalkow, Pawel Gawrychowski, and Pierre Ohlmann. Value iteration using universal graphs and the complexity of mean payoff games. In Javier Esparza and Daniel Král', editors, *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic*, volume 170 of *LIPIcs*, pages 34:1–34:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020. doi:10.4230/LIPIcs.MFCS.2020.34.

17 Oliver Friedmann. An exponential lower bound for the latest deterministic strategy iteration algorithms. *Log. Methods Comput. Sci.*, 7(3), 2011. doi:10.2168/LMCS-7(3:23)2011.

18 B. Gärtner. The Random-Facet simplex algorithm on combinatorial cubes. *Random Structures & Algorithms*, 20(3):353–381, 2002. doi:10.1002/rsa.10034.

19 Vladimir A. Gurvich, Alexander V. Karzanov, and L.G. Khachivan. Cyclic games and an algorithm to find minimax cycle means in directed graphs. *USSR Computational Mathematics and Mathematical Physics*, 28:85–91, 1988.

20 Nir Halman. Simple stochastic games, parity games, mean payoff games and discounted payoff games are all LP-type problems. *Algorithmica*, 49(1):37–50, 2007. doi:10.1007/S00453-007-0175-3.

21 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. *J. ACM*, 60(1):1:1–1:16, 2013. doi:10.1145/2432622.2432623.

22 Thomas Dueholm Hansen, Mike Paterson, and Uri Zwick. Improved upper bounds for random-edge and random-jump on abstract cubes. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 874–881. SIAM, 2014. doi:10.1137/1.9781611973402.65.

23 Thomas Dueholm Hansen and Uri Zwick. An improved version of the random-facet pivoting rule for the simplex algorithm. In *Proc. of 47th STOC*, pages 209–218, 2015. doi:10.1145/2746539.2746557.

24 Thomas Dueholm Hansen and Uri Zwick. Random-Edge Is Slower Than Random-Facet on Abstract Cubes. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 51:1–51:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.51.

25 Romain Hollanders, Balázs Gerencsér, Jean-Charles Delvenne, and Raphaël M Jungers. About upper bounds on the complexity of policy iteration, 2014.

26 R.A. Howard. *Dynamic programming and Markov processes.* MIT Press, 1960.

27 Marcin Jurdziński. Deciding the winner in parity games is in up ∩ co-up. *Information Processing Letters*, 68(3):119–124, 1998.

28 Marcin Jurdziński and Ranko Lazić. Succinct progress measures for solving parity games. In *Proc. of 32nd LICS*, pages 1–9, 2017. `doi:10.1109/LICS.2017.8005092`.

29 Gil Kalai. A subexponential randomized simplex algorithm (extended abstract). In *Proc. of 24th STOC*, pages 475–482, 1992. `doi:10.1145/129712.129759`.

30 Gil Kalai. Linear programming, the simplex algorithm and simple polytopes. *Math. Program.*, 79:217–233, 1997. `doi:10.1007/BF02614318`.

31 Alexander Kozachinskiy. Polyhedral value iteration for discounted games and energy games. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 600–616. SIAM, 2021. `doi:10.1137/1.9781611976465.37`.

32 Karoliina Lehtinen, Pawel Parys, Sven Schewe, and Dominik Wojtczak. A recursive approach to solving parity games in quasipolynomial time. *Log. Methods Comput. Sci.*, 18(1), 2022. `doi:10.46298/LMCS-18(1:8)2022`.

33 László Lovász. *Combinatorial problems and exercises*, volume 361. American Mathematical Soc., 2007.

34 Walter Ludwig. A subexponential randomized algorithm for the simple stochastic game problem. *Inf. Comput.*, 117(1):151–155, 1995. `doi:10.1006/inco.1995.1035`.

35 Yishay Mansour and Satinder Singh. On the complexity of policy iteration. *arXiv preprint arXiv:1301.6718*, 2013. `arXiv:1301.6718`.

36 Jiří Matoušek, Micha Sharir, and Emo Welzl. A subexponential bound for linear programming. *Algorithmica*, 16(4/5):498–516, 1996. `doi:10.1007/BF01940877`.

37 Jiří Matoušek and Tibor Szabó. Random edge can be exponential on abstract cubes. *Advances in Mathematics*, 204(1):262–277, 2006.

38 Jirí Matoušek. Lower bounds for a subexponential optimization algorithm. *Random Struct. Algorithms*, 5(4):591–608, 1994. `doi:10.1002/RSA.3240050408`.

39 Robert McNaughton. Infinite games played on finite graphs. *Ann. Pure Appl. Logic*, 65(2):149–184, 1993. `doi:10.1016/0168-0072(93)90036-D`.

40 Anuj Puri. *Theory of hybrid systems and discrete event systems.* University of California, Berkeley, 1995.

41 Tibor Szabó and Emo Welzl. Unique sink orientations of cubes. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 547–555. IEEE Computer Society, 2001. `doi:10.1109/SFCS.2001.959931`.

42 Jens Vöge and Marcin Jurdzinski. A discrete strategy improvement algorithm for solving parity games. In E. Allen Emerson and A. Prasad Sistla, editors, *Computer Aided Verification, 12th International Conference, CAV 2000, Chicago, IL, USA, July 15-19, 2000, Proceedings*, volume 1855 of *Lecture Notes in Computer Science*, pages 202–215. Springer, 2000. `doi:10.1007/1072216718`.

43 Yinyu Ye. The simplex and policy-iteration methods are strongly polynomial for the markov decision problem with a fixed discount rate. *Math. Oper. Res.*, 36(4):593–603, 2011. `doi:10.1287/MOOR.1110.0516`.

44 Uri Zwick and Mike Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158(1-2):343–359, 1996. `doi:10.1016/0304-3975(95)00188-3`.

A Correctness of Strategy Improvement

The next two lemmas give structural insights on the costs of edges in $play^{\sigma,\pi}$ and $play^{\sigma',\pi'}$.

▶ **Lemma A.1.** *Let $\sigma, \sigma', \pi, \pi'$ be as in Theorem 2.11. Let $u \in V$ with $val^{\sigma,\pi}(u) > 0$.*

(i) *If $u \in V_0$, then $val^{\sigma,\pi}(u) \leq c(u,v) + val^{\sigma,\pi}(v)$ where $(u,v) = \sigma'(u)$.*

(ii) *If $u \in V_1$, then $val^{\sigma,\pi}(u) \leq c(u,v) + val^{\sigma,\pi}(v)$ where $(u,v) = \pi'(u)$. Moreover, if $\pi'(u) \neq \pi(u)$ then the inequality is strict.*

Proof.

(i) Immediate by Corollary 2.9.

(ii) If $\pi'(u) = \pi(u)$ then $val^{\sigma,\pi}(u) = c(u,v) + val^{\sigma,\pi}(v)$. Otherwise, since $\pi' \in SI(\pi)$ we get $val^{\sigma,\pi}(u) < c(u,v) + val^{\sigma,\pi}(v)$. ◀

▶ **Lemma A.2.** *Let $\sigma, \sigma', \pi, \pi'$ be as in Theorem 2.11. Let $u_1, ..., u_k = u_1$ be a cycle in $G(\sigma', \pi')$. If $val^{\sigma,\pi}(u_i) > 0$ for every $1 \leq i < k$, then the cycle is positive.*

Proof. By Lemma A.1 the modified costs, with respect to $f = val^{\sigma,\pi}$, of all edges on the cycle are nonnegative. Thus, the modified cost of the cycle, and hence its original cost, is nonnegative. Assume, for the sake of contradiction, that the cost of the cycle is 0 (so all its edges are tight). By Lemma A.1(ii) we have $\pi'(u_i) = \pi(u_i)$ for every i such that $u_i \in V_1$. Thus $u_1, ..., u_k$ is also a cycle in $G(\sigma', \pi)$ and since it has zero cost there exists $1 \leq i < k$ such that $val^{\sigma',\pi}(u_i) = 0 < val^{\sigma,\pi}(u_i)$, a contradiction to σ being a best-response to π. ◀

We are now ready to prove Theorem 2.11.

Proof of Theorem 2.11. Let $v \in V$, we prove $val^{\sigma,\pi}(v) \leq val^{\sigma',\pi'}(v)$. If $val^{\sigma,\pi}(v) = 0$ then this holds trivially. Assume $val^{\sigma,\pi}(v) > 0$ and let $P = v_1, v_2, ...$ be the infinite walk in $G(\sigma', \pi')$ starting at $v_1 = v$. If $val^{\sigma,\pi}(v_i) > 0$ for all i then by Lemma A.2, P loops over a positive cycle and thus $val^{\sigma',\pi'}(v) = \infty \geq val^{\sigma,\pi}(v)$. Otherwise, let j be the minimal index such that $val^{\sigma,\pi}(v_j) = 0$. By Lemma A.1 it holds that $c(v_i, v_{i+1}) + val^{\sigma,\pi}(v_{i+1}) - val^{\sigma,\pi}(v_i) \geq 0$ for all $i < j$. Summing these terms up we get

$$0 \leq \sum_{i=1}^{j-1} c(v_i, v_{i+1}) + val^{\sigma,\pi}(v_j) - val^{\sigma,\pi}(v) = \sum_{i=1}^{j-1} c(v_i, v_{i+1}) - val^{\sigma,\pi}(v).$$

Thus, P has a prefix with cost at least $val^{\sigma,\pi}(v)$ yielding $val^{\sigma',\pi'}(v) \geq val^{\sigma,\pi}(v)$.

We are left to prove that if $\pi \neq \pi'$, then there exists a vertex $u \in V$ such that $val^{\sigma,\pi}(u) < val^{\sigma',\pi'}(u)$. Let $u \in V_1$ be such that $(u,v) = \pi'(u) \neq \pi(u)$, therefore $val^{\sigma,\pi}(u) < \infty$. By Lemma A.1(ii), it holds that $c(u,v) + val^{\sigma,\pi}(v) - val^{\sigma,\pi}(u) > 0$. Let $P = v_1, v_2, ...$ be the infinite walk in $G(\sigma', \pi')$ starting at $v_1 = u$. As in the previous argument if $val^{\sigma,\pi}(v_i) > 0$ for all i then $val^{\sigma',\pi'}(u) = \infty > val^{\sigma,\pi}(u)$. Otherwise, as above, we get that

$$0 < \sum_{i=1}^{j-1} c(v_i, v_{i+1}) - val^{\sigma,\pi}(u),$$

thereby proving the theorem. ◀

B Missing Proofs

Proof of Lemma 3.5'. By induction on $0 \le j \le \min\{k, \ell\}$. The case $j = 0$ is trivial. As in the proof of Lemma 3.5, we prove:

- If j is even and $v \in L_j^i$, then $f_{i+1}(v) = f_i(v)$ and $v \in L_j^{i+1}$.
- If j is odd and $v \in L_j^{i+1}$, then $f_{i+1}(v) = f_i(v)$ and $v \in L_j^i$.

The case in which j is even is identical to the original proof of Lemma 3.5.

Assume j is odd and let $v \in L_j^{i+1}$. Thus $v \in V_1$. By the definition of the layers, v has an f_{i+1}-tight path $P = v_0(= v), v_1, ..., v_r$, containing vertices of L_j^{i+1}, to $v_r \in L_{j-1}^{i+1}$. The following claim (specifically, the case $s = 0$) finishes the proof.

▷ Claim B.1. For every $s < r$, it holds that $v_s \in L_j^{i-1} \cap L_j^i \cap L_j^{i+1}$ and $f_{i+1}(v_s) = f_{i-1}(v_s)$.

Proof. The proof is by a backward induction on $s = r-1, r-2, \ldots, 1$.

We begin by showing that $f_{i+1}(v_{r-1}) = f_i(v_{r-1})$. Consider the edge $e = (v_{r-1}, v_r) \in E$ which is f_{i+1}1-tight. Since $v_r \in L_{j-1}^{i+1}$, then by the induction hypothesis of the main induction on j, it holds that $f_{i+1}(v_r) = f_i(v_r)$. Moreover, since $j - 1 < \min(k, \ell)$ we have that $f_i(v_r) = f_{i-1}(v_r)$. By strong stability[11] at iteration i, e must be f_{i-1}-valid, hence

$$f_{i-1}(v_{r-1}) \ge c(e) + f_{i-1}(v_r) = c(e) + f_{i+1}(v_r) = f_{i+1}(v_{r-1}),$$

where the last equality holds since P is f_{i+1}-tight. Hence $f_{i+1}(v_{r-1}) = f_{i-1}(v_{r-1})$.

Thus, e is also f_i-tight, and therefore $v_{r-1} \in L_j^i$. By stability, $L_j^i \subseteq L_j^{i-1}$ and we are done. ◁

The proof of the induction step is similar. We use the equality $f_{i+1}(v_{s+1}) = f_{i-1}(v_{s+1})$ and the inductive assumption that $v_{s+1} \in L_j^i$ to deduce that $v_s \in L_j^i$ and that $f_{i+1}(v_s) = f_{i-1}(v_s)$. ◀

Proof of Lemma 3.7'. The proof that the first $2k$ layers are stable at iteration $i + 1$ is identical to the original proof (but this time uses Lemma 3.5' instead of Lemma 3.5). The first $2k - 1$ layers are strongly static by the assumption. Hence, by the definition of strong stability, the first $2k$ layers are strongly stable at iteration $i + 1$ ◀

Proof of Lemma 5.7. The assumption implies that the first k layers are strongly static during those iterations. We bound the probability that $\ell > 3mn \ln n$.

Let $v \in L_{k+1}^{i+\ell}$ and let $P = v_0 \ldots v_r$ be an $f_{i+\ell}$-tight path in $L_{k+1}^{i+\ell}$ from $v = v_0$ to $v_r \in L_k^{i+\ell}(= L_k^i)$. We prove by backward induction on $s < r$ that with probability of at least $1 - (r - s) \cdot \frac{1}{n^2}$, $v_s \in L_{k+1}^{i+j}$ for every $j \ge 2(r - s) \cdot m \ln n$. i.e., the tth last vertex on P stayed in layer $k + 1$ since iteration $i + t \cdot m \ln n$. Indeed, assume $\{v_r, v_{r-1}, \ldots, v_s\}$ are in L_{k+1} since iteration $T = i + 2(r - s) \cdot mn \ln n$. If $e = (v_{s-1}, v_s)$ is f_T-valid, then we are done (since $f_T(v_s) = f_{i+\ell}(v_s)$ and thus e is tight and will remain so). Assume that e is an improving switch with respect to f_T. Hence, in each iteration (until e becomes valid) there is a probability of at least $\frac{1}{m}$ that e will be chosen by Random-Edge. Therefore with probability of at least $1 - \frac{1}{n^2}$, after $2m \ln n$ iterations e will become valid and therefore v_{s-1} belongs to layer $k + 1$ since iteration $T + 2m \ln n = i + 2(r - s + 1) \cdot m \ln n$ until iteration ℓ. The inductive claim is achieved by applying union bound with the inductive step.

[11] This is the only deviation from the original proof.

Hence, with probability of at least $1 - n \cdot \frac{1}{n^2}$, all vertices in $L^{i+\ell}_{k+1}$ remained in layer $k+1$ since iteration $i + 2mn \ln n \leq i + \ell - mn \ln n$ (and their f values did not change). This means that the first $k+1$ layers are static since iteration $i + 2mn \ln n$ for at least $mn \ln n$ iterations. Since the first $k+1$ layers are not strongly stable, there is an edge $e \in E(L^{i+2mn\ln n}_{k+1}, L^{i+2mn\ln n}_{\leq k+1}) \cup E(V_1, L^{i+2mn\ln n}_{\leq k+1})$ which is an improving switch since iteration $i + 2mn \ln n$. The probability of e not being chosen until iteration $i + \ell$ is at most $\frac{1}{n^2}$. Therefore, by union bound, the probability that the assumption of the lemma holds is at least $1 - n \cdot \frac{1}{n^2} - \frac{1}{n^2} \geq \frac{1}{2}$. ◀

C Example

▶ **Example C.1.** In the following example we present a game graph in which, during greedy strategy improvement, it takes several iterations for an odd layer to stabilize. Let $G = (V_0 = \{s\}, V_1 = \{v_1, ..., v_m\}, c)$ be the game graph illustrated in Figure 1. The edge costs satisfy $c(v_i, v_{i-1}) = 0, c(v_i, s) = 1$ for every $2 \leq i \leq m$ and $c(s, s) = 0, c(v_1, s) = 2$. Note that V is winning for player-0, and moreover s is winning for player-0 with 0 initial energy.

Let the initial strategy for player-1 be $\pi_1(v_i) = s$ for all i. The strategy edges are full lines, while non strategy edges are dashed. It is easy to see that $L^1_0 = \{s\}, L^1_1 = V$ since we have $f_1(s) = 0, f_1(v_1) = 2, f_1(v_i) = 1$ for every $i \geq 2$. Note that (v_2, v_1) is the only improving switch with respect to f_1 and therefore player-1 performs an improving switch.

By induction, we get that

$$\pi_k(v_i) = \begin{cases} s & i = 1 \\ v_{i-1} & 1 < i \leq k \\ s & i > k \end{cases}, \quad f_k(v_i) = \begin{cases} 2 & i \leq k \\ 1 & i > k \end{cases}, \quad L^k_0 = \{s\}, L^k_1 = V$$

Thus, it takes L_1 at least $m - 1$ iterations to stabilize.

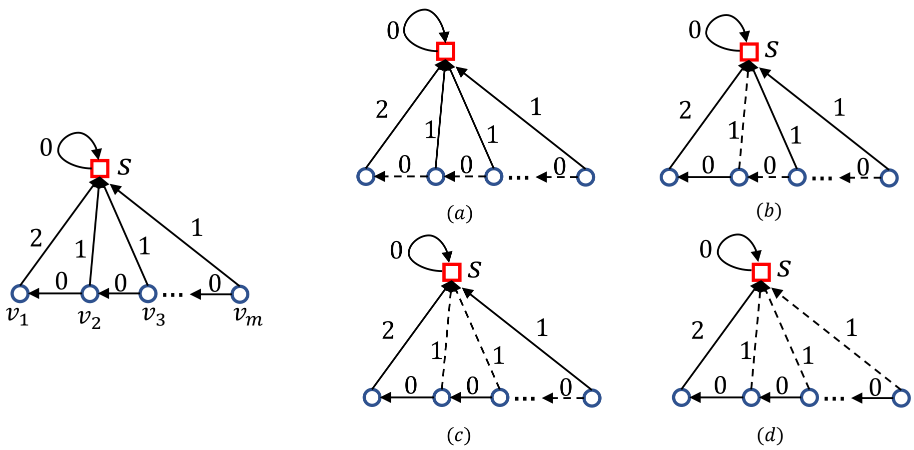

Figure 1 The energy game is given on the left. Red squares refer to player 0 and blue circles refer to player 1. The sequence (a)-(d) corresponds to greedy strategy improvements. Dashed edges are *not* used in the current strategy of player 1. Strategies $(a) - (c)$ are the first 3 strategies, while strategy (d) is the strategy at iteration m.