On Friedmann's Subexponential Lower Bound for Zadeh's Pivot Rule

Yann Disser[1,2] and Alexander V. Hopp[1,2(✉)]

[1] Graduate School of Computational Engineering, TU Darmstadt, Darmstadt, Germany
hopp@gsc.tu-darmstadt.de
[2] Department of Mathematics, TU Darmstadt, Darmstadt, Germany
disser@mathematik.tu-darmstadt.de

Abstract. The question whether the Simplex method admits a polynomial time pivot rule remains one of the most important open questions in discrete optimization. Zadeh's pivot rule had long been a promising candidate, before Friedmann (IPCO, 2011) presented a subexponential instance, based on a close relation to policy iteration algorithms for Markov decision processes (MDPs). We investigate Friedmann's lower bound construction and exhibit three flaws in his analysis: We show that (a) the initial policy for the policy iteration does not produce the required occurrence records and improving switches, (b) the specification of occurrence records is not entirely accurate, and (c) the sequence of improving switches described by Friedmann does not consistently follow Zadeh's pivot rule. In this paper, we resolve each of these issues. While the first two issues require only minor changes to the specifications of the initial policy and the occurrence records, the third issue requires a significantly more sophisticated ordering and associated tie-breaking rule that are in accordance with the LEAST-ENTERED pivot rule. Most importantly, our changes do not affect the macroscopic structure of Friedmann's MDP, and thus we are able to retain his original result.

1 Introduction

The Simplex method, originally proposed by Dantzig in 1947 [2], is one of the most important algorithms to solve linear programs in practice. At its core, it operates by maintaining a subset of basis variables while restricting non-basis variables to trivial values, and repeatedly replacing a basis variable according to a fixed *pivot rule* until the objective function value can no longer be improved. Exponential worst-case instances have been devised for many natural pivot rules (see, for example, [1,5,7,8]), and the question whether a polynomial time pivot rule exists remains one of the most important open problems in optimization.

This work is supported by the 'Excellence Initiative' of the German Federal and State Governments and the Graduate School CE at TU Darmstadt.
Full version digitally published at the University and State Library Darmstadt, available at http://tuprints.ulb.tu-darmstadt.de/id/eprint/7557.

A. Lodi and V. Nagarajan (Eds.): IPCO 2019, LNCS 11480, pp. 168–180, 2019.
https://doi.org/10.1007/978-3-030-17953-3_13

Zadeh's Least-Entered pivot rule [11] was designed to avoid the exponential behavior on known worst-case instances for other pivot rules. It is *memorizing* in that it selects a variable to enter the basis that improves the objective function and has previously been selected least often among all improving variables. For more than thirty years, Zadeh's rule defied all attempts to construct superpolynomial instances, and seemed like a promising candidate for a polynomial pivot rule.

It was a breakthrough when Friedmann eventually presented the first superpolynomial lower bound for Zadeh's pivot rule [4]. His construction uses a connection between the Simplex Algorithm and Howard's Policy Iteration Algorithm [6] for computing optimal policies in Markov decision processes (MDPs). For a given $n \in \mathbb{N}$, Friedmann's construction consists of an MDP of size $\mathcal{O}(n^2)$, an initial policy, and an ordering of the improving switches obeying the Least-Entered pivot rule. This ordering results in an exponential number of iterations when beginning with the initial policy and repeatedly making improving switches in the specified order. The construction translates into a linear program of the same asymptotic size for which the Simplex Algorithm with Zadeh's pivot rule needs $\Omega(2^n)$ steps. Since the input size is $\mathcal{O}(n^2)$, this in turn results in a superpolynomial lower bound. Recently, an exponential lower bound for Zadeh's pivot rule was found for AUSOs [10], but it is not clear whether this construction can be realized as a linear program. However, the construction is simpler than Friedmann's construction with more natural tie-breaking, and thus presents an alternative approach to devising lower bounds for memorizing pivot rules.

Our Contribution. In this paper, we expose different flaws in Friedmann's construction and present adaptations to eliminate them. We first show that the chosen initial policy does not produce the claimed set of improving switches, and propose a modified initial policy that leads to the desired behavior. Second, we observe that the given formula describing the occurrence records (that count the number of times an improving switch was made) is inaccurate, and provide a (small) correction that does not disturb the overall argument.

Most importantly, we exhibit a significant problem with the order in which the improving switches are applied in [4]. More precisely, we show that this order does not consistently obey Zadeh's pivot rule, and, in fact, that no consistent ordering exists that updates the MDP "level by level" in each phase according to a fixed order. This not only rules out Friedmann's ordering, but shows that a fundamentally different approach to ordering improving switches is needed. To amend this issue, we show the existence of an ordering and a tie-breaking rule compatible with the Least-Entered rule, such that applying improving switches according to the ordering still proceeds along the same macroscopic phases as intended by Friedmann. In this way, we are able to quantitatively retain Friedmann's superpolynomial lower bound.

Outline. Throughout this paper, we assume some basic familiarity with the construction given in [4] and Markov decision processes in general. We review the most important aspects and notation of [4] in Sect. 2. Section 3 treats issues with the initial policy and the description of the occurrence records and our adaptations to address them. The main part of this paper is Sect. 4, where we show that the sequence of improving switches can be reordered such that the Least-Entered rule is obeyed.

2 Friedmann's Lower Bound Construction

In [4], Friedmann uses the connection between the Simplex Algorithm and the Policy Iteration Algorithm for obtaining optimal policies in MDPs. Similarly, we also restrict our discussion to policy iteration for MDPs, with the understanding that results carry over to the Simplex Algorithm. We assume knowledge of MDPs and the connection to the Simplex Algorithm and refer to [9] for more information. For convenience, we refer to improving switches simply as switches.

Let $n \in \mathbb{N}$. Friedmann's construction emulates an n-bit binary counter by a Markov decision process G_n. For every n-digit binary number b, there is a policy σ_b for G_n representing b. We denote the i-th bit of b by b_i, so $b = (b_n, \dots, b_1)$. The MDP G_n is constructed such that applying the Policy Iteration Algorithm using the Least-Entered rule enumerates the policies σ_0 to σ_{2^n-1}. According to the pivot rule, the algorithm always chooses a switch chosen least often in the past. More specifically, an *occurrence record* ϕ is maintained, and, in every step, a switch minimizing ϕ is chosen. The rule does however not determine *which* switch minimizing ϕ should be chosen, so a tie-breaking rule is needed. For an edge e and a policy σ, we denote the occurrence record of e once σ is reached by $\phi^\sigma(e)$. We denote the set of improving switches with respect to σ by I_σ.

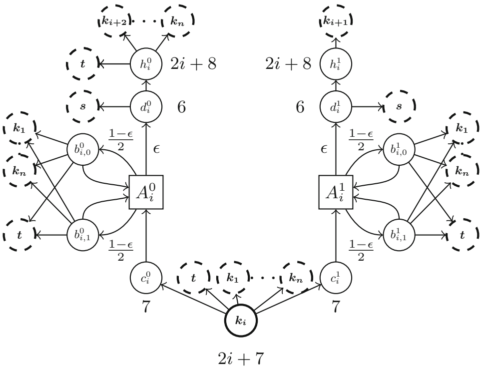
We fix the following notation. The set of n-digit binary numbers is denoted by $\mathbb{B}_n$. For $b \in \mathbb{B}_n, b \neq 0$, $\ell(b)$ denotes the least significant bit of b equal to 1. The unique policy representing $b \in \mathbb{B}_n$ constructed in [4] is denoted by σ_b.

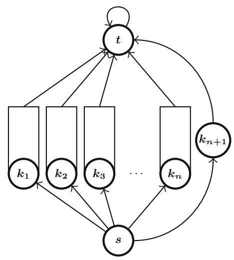
The process G_n can be interpreted as a "fair alternating binary counter" as follows. Usually, when counting from 0 to $2^n - 1$ in binary, less significant bits are switched more often than more significant bits. As the Least-Entered pivot rule forces the algorithm to switch all bits equally often, the construction must ensure to operate correctly when all bits are switched equally often. This is achieved by representing every bit by *two* gadgets where only one *actively* represents the bit. The gadgets alternate in actively representing the bit.

The construction consists of n structurally identical levels, where level i represents the i-th bit. A large number $N \in \mathbb{N}$ is used for defining the rewards and a small number $\epsilon \geq 0$ is used for defining the probabilities. The i-th level is shown in Fig. 1(a), the coarse structure of the whole MDP in Fig. 1(b).

A number n_v below or next to a vertex v in Fig. 1(a) denotes a reward of $(-N)^{n_v}$ associated with every edge leaving v. Other edges have a reward of 0.

Each level i contains two gadgets attached to the *entry vertex* k_i, called *lanes*. We refer to the left lane as *lane 0* and to the right lane as *lane 1*. Lane j of level i contains a randomization vertex A_i^j and two attached cycles with vertices $b_{i,0}^j$ and $b_{i,1}^j$. These gadgets are called *bicycles*, and we identify the bicycle containing A_i^j with that vertex. For a bicycle A_i^j, the edges $(b_{i,0}^j, A_i^j), (b_{i,1}^j, A_i^j)$ are called *edges of the bicycle*. For a policy σ, the bicycle A_i^j is *closed (w.r.t. σ)* if and only if $\sigma(b_{i,0}^j) = \sigma(b_{i,1}^j) = A_i^j$. A bicycle that is not closed is called *open*.





(a) Circular vertices are player-controlled, squares are randomization vertices. Bold vertices can be reached from other levels, dashed vertices do not belong to level i.

(b) The entry vertices are all connected to s and t. Connections between levels and from the levels to s are not shown here. The vertex k_{n+1} is needed for technical reasons.

Fig. 1. Level i of G_n (left) and coarse structure of G_n (right)

The i-th level of G_n corresponds to the i-th bit of the counter. Which of the bicycles of level i is actively representing the i-th bit depends on b_{i+1}. When this bit is equal to 1, A_i^1 is considered active. Otherwise, A_i^0 is considered active. The i-th bit is equal to 1 if and only if the active bicycle in level i is closed.

As initial policy, the MDP is provided the policy $\sigma^\star$ representing 0. Then, a sequence of policies $\sigma_1, \ldots, \sigma_{2^n-1}$ is enumerated by the Policy Iteration Algorithm using the Least-Entered pivot rule and an (implicit) tie-breaking rule. For $b \in \mathbb{B}_n, b \neq 0$, the policy σ_b should fulfill the following invariants.

1. Exactly the bicycles $A_i^{b_{i+1}}$ corresponding to bits $b_i = 1$ are closed.
2. For all other bicycles A_i^j, it holds that $\sigma_b(b_{i,0}^j) = \sigma_b(b_{i,1}^j) = k_{\ell(b)}$.
3. All entry vertices point to the lane containing the active bicycle if $b_i = 1$ and to $k_{\ell(b)}$ otherwise.
4. The vertex s points to the entry vertex of the least significant set bit.
5. All vertices h_i^0 point to the entry vertex of the first level after level $i+1$ corresponding to a set bit, or to t if no such level exists.
6. The vertex d_i^j points to h_i^j if and only if $b_{i+1} = j$ and to s otherwise.

The Policy Iteration Algorithm is only allowed to switch one edge per iteration. However, the policy σ_{b+1} cannot be reached from σ_b by performing a single switch. Therefore, intermediate policies need to be introduced for the transition from σ_b to σ_{b+1}. These intermediate policies are divided into six *phases*. In each phase, a different "task" is performed within the construction. Let $\ell' := \ell(b+1)$.

1. In phase 1, switches inside of some bicycles are performed to keep the occurrence records of the bicycle edges as balanced as possible. For every open bicycle A_i^j, at least one of the edges $(b_{i,0}^j, A_i^j), (b_{i,1}^j, A_i^j)$ is switched. Some

bicycles are allowed to switch both edges such that their occurrence record can "catch up" with the other edges. In the active bicycle of level ℓ', we also switch both edges, as this bicycle needs to be closed with respect to σ_{b+1}.
2. In phase 2, the new least significant set bit is "made accessible". Thus, $k_{\ell'}$ is switched to $c^j_{\ell'}$, where j is the lane containing the active bicycle.
3. In phase 3, we perform the "resetting process". The entry vertices of all levels i corresponding to bits with $(b+1)_i = 0$ switch to $k_{\ell'}$. The same is done for all vertices $b^j_{i,l}$ contained in inactive bicycles and all vertices $b^j_{i,l}$ corresponding to levels i with $(b+1)_i = 0$.
4. In phase 4, the vertices h^0_i are updated according to $\ell(b+1)$.
5. In phase 5, we switch s to the entry vertex corresponding to $\ell(b+1)$.
6. In phase 6, we update the vertices d^j_i such that h^0_i is the target of d^0_i if and only if $(b+1)_{i+1} = 0$ and h^1_i is the target of d^1_i if and only if $(b+1)_{i+1} = 1$.

Before discussing our findings, we need to introduce notation related to binary counting. We further briefly describe the tables contained in [4].

Let $b \in \mathbb{B}_n$. By binary counting, we refer to the process of enumerating the binary representations of all numbers $\tilde{b} \in \{0, 1, \ldots, b\}$. These numbers are used to determine how often and when edges of G_n are improving and will be applied.

Intuitively, we are interested in *schemes* that we observe when counting from 0 to b, or, more precisely, in the set of numbers that *match a scheme*. A *scheme* is a set $S \subseteq \mathbb{N} \times \{0, 1\}$ and b *matches* S if $b_i = q$ for all $(i, q) \in S$. Since the occurrence records of the edges depend on how often a specific scheme occurred when counting from 0 to b, we introduce the following terms.

Definition 1 ([4]). *Let $b \in \mathbb{B}_n, i \in \{1, \ldots, n\}$ and let S be a scheme. The* flip set *$F(b, i, S)$ is the set of all numbers between 0 and b matching S whose least significant bit is the i-th bit. The* flip number *is defined as $f(b, i, S) := |F(b, i, S)|$ and we set $f(b, i) := f(b, i, \emptyset)$.*

We now briefly describe the tables of [4]. For $p \in \{1, \ldots, 6\}$, [4, Table 2] defines the term phase p policy. As we prove later, there is an issue concerning the side conditions of phase 3. For a phase p policy σ, [4, Table 3] contains subsets L^p_σ and supersets U^p_σ of the set of improving switches. The last table is [4, Table 4]. For $b \in \mathbb{B}_n$, it contains the occurrence records ϕ^{σ_b} of the edges with respect to the unique policy representing b. We discuss an issue regarding this table in Sect. 3. Other than correcting these issues, we rely on [4, Tables 2, 3, 4].

3 Initial Policy and Occurrence Records

In this section, we discuss the initial policy $\sigma^\star$ used in [4] and the description of the occurrence records given in [4, Table 4]. We show that $\sigma^\star$ contradicts several aspects of [4], and provide an alternative initial policy resolving these issues. Then, we discuss why the description of the occurrence records given in [4, Table 4] is not entirely accurate and provide a correction of this inaccuracy.

On [4, page 11], the following is stated regarding $\sigma^\star$: "*As designated initial policy* $\sigma^\star$, *we use* $\sigma^\star(d_i^j) = h_i^j$ *and* $\sigma^\star(\_) = t$ *for all other player 0 nodes with non-singular out-degree.*" This initial policy, however, is inconsistent with the sub- and supersets of improving switches given in [4, Table 3] and [4, Lemma 4].[1]

Issue 1. The initial policy $\sigma^\star$ described in [4, page 10] contradicts [4, Table 3] since $I_{\sigma^\star} \neq \{(b_{i,r}^j, A_i^j) : \sigma^\star(b_{i,r}^j) \neq A_i^j\}$. In addition, when the Policy Iteration Algorithm is started with $\sigma^\star$, at least one of [4, Tables 3, 4] is incorrect for $b = 1$.

Thus, the initial policy needs to be changed. We propose the following policy that resolves both issues. Note that it also fulfills [4, Lemma 1] and can thus indeed be used as initial policy for G_n.

Theorem 1. *Define the policy* σ^* *via* $\sigma^*(d_i^0) := h_i^0$ *and* $\sigma^*(d_i^1) := s$ *for all* $i \in \{1, \dots, n\}$ *and* $\sigma^*(\cdot) := t$ *for all other player-controlled vertices with non-singular out-degree. Then* $I_{\sigma^*} = \{(b_{i,r}^j, A_i^j) : \sigma^*(b_{i,r}^j) \neq A_i^j\}$ *and starting the Policy Iteration Algorithm with* σ^* *does not contradict [4, Tables 3, 4] for* $b = 1$.

We next prove an issue related to the occurrence records of the bicycles as specified in [4, Table 4].

Issue 2. Let $b < 2^{n-k-1} - 1$ for some $k \in \mathbb{N}$. Assume that the occurrence records of the edges are given by [4, Table 4]. Then, there is a pair $(b_{i,0}^j, A_i^j), (b_{i,1}^j, A_i^j)$ such that at least one of them has a negative occurrence record.

The problem is that the given description does not properly distinguish between inactive bicycles that need to catch up with the counter and inactive bicycles that do not need to do so. Informally, for $b \in \mathbb{B}_n$ the occurrence records once the policy σ_b is reached can be described as follows: (a) Every closed and active bicycle has an occurrence record corresponding to the last time it was closed, (b) every open and active bicycle has an occurrence record of b and (c) inactive bicycles are either "catching up" with other bicycles and thus have an occurrence record less than b or already finished catching up and have an occurrence record of b.

To resolve Issue 2, we formulate an additional condition. It is used to distinguish inactive bicycles that might need to catch up with the counter because they have already been closed once (if $b \geq 2^{i-1} + j \cdot 2^i$), and inactive bicycle that do not need to catch up because they have not been closed before.

To formulate this condition, we need more notation. Let $b \in \mathbb{B}_n$ and A_i^j be a fixed bicycle. We define g as the largest number smaller than b such that the least significant set bit of g has index i and the $(i+1)$-th bit is equal to j. In addition, we define $z := b - g - 2^{i-1}$ and $\phi^{\sigma_b}(A_i^j) := \phi^{\sigma_b}(b_{i,0}^j, A_i^j) + \phi^{\sigma_b}(b_{i,1}^j, A_i^j)$.

According to the proof of [4, Lemma 5], the switches inside a cycle center A_i^j should then be applied according to the following rules.

[1] Proofs for all statements can be found in the full paper [3].

1. If A_i^j is open and active, we switch one edge of the bicycle.
2. Let $j := b_{\ell(b+1)+1}$. In addition to 1., the second edge of $A_{\ell(b+1)}^j$ is switched.
3. If A_i^j is inactive and $b < 2^{i-1} + j \cdot 2^i$, one edge of the bicycle is switched.
4. If A_i^j is inactive, $b \geq 2^{i-1} + j \cdot 2^i$ and $z < \frac{1}{2}(b-1-g)$, both edges of A_i^j are switched. If $z \geq \frac{1}{2}(b-1-g)$, only one edge is switched.

The following theorem gives a correct description of the occurrence records.

Theorem 2. *Suppose that improving switches within bicycles are applied as described by rules 1 to 4. Let $b \in \mathbb{B}_n$ and A_i^j be a bicycle. Then, the occurrence records of $(b_{i,0}^j, A_i^j)$ and $(b_{i,1}^j, A_i^j)$ are correctly specified by the system*

$$|\phi^{\sigma_b}(b_{i,0}^j, A_i^j) - \phi^{\sigma_b}(b_{i,1}^j, A_i^j)| \leq 1 \tag{3.1}$$

$$\phi^{\sigma_b}(A_i^j) = \begin{cases} g+1, & A_i^j \text{ is closed and active} \\ b, & A_i^j \text{ is open and active} \\ b, & A_i^j \text{ is inactive and} b < 2^{i-1} + j \cdot 2^i \\ g+1+2z, & A_i^j \text{ is inactive and} b \geq 2^{i-1} + j \cdot 2^i \end{cases} \tag{3.2}$$

4 Improving Switches of Phase 3

We next discuss the application of improving switches in phase 3. There are two contradictory descriptions in [4] how to apply these. We prove that neither of the given orderings obeys the Least-Entered rule. We additionally show that a natural adaptation of Friedmann's scheme still does not obey the Least-Entered rule. We then go on to prove the existence of an ordering and an associated tie-breaking rule that obey the Least-Entered rule while still producing the intended behavior of Friedmann's construction.

Throughout this section, for a fixed $b \in \mathbb{B}_n$, we use $\ell := \ell(b)$ and $\ell' := \ell(b+1)$.

4.1 Issues with Friedmann's Switching Order

In Sect. 2, we stated that during phase 3, improving switches need to be applied for every entry vertex k_i contained in a level i with $(b+1)_i = 0$. In addition, several bicycles need to be opened. However, according to the description given in [4, pages 9–10], both of these updates should not be performed for all levels but only those with an index smaller than ℓ'. To be precise, the following is stated:[2] "*In the third phase, we perform the major part of the resetting process. By resetting, we mean to unset lower bits again, which corresponds to reopening the respective bicycles. Also, we want to update all other inactive or active but not set bicycles again to move to the entry point $k_{\ell'}$. In other words, we need to update the lower entry points k_z with $z < \ell'$ to move to $k_{\ell'}$, and the bicycle nodes $b_{z,l}^j$ to move to $k_{\ell'}$. We apply these switches by first switching the entry node k_z for some $z < \ell'$ and then the respective bicycle nodes $b_{z,r}^j$.*"

However, there is an issue regarding this informal description.

[2] The notation in the quote was adapted from [4] to be in line with our paper.

Issue 3. For every $b \in \{1, \ldots, 2^{n-2} - 1\}$, the informal description of phase 3 contradicts [4, Tables 2, 4]. It additionally violates the LEAST-ENTERED pivot rule during the transition from σ_b to σ_{b+1} for every $b \in \{3, \ldots, 2^{n-2} - 2\}$.

In other parts of the construction, Friedmann seems to apply the switches differently, by not only applying them for levels with a lower index than the least significant set bit but for all levels. Especially, the side conditions of [4, Table 2] for defining a phase p policy rely on the fact that these switches are applied for all levels i with $(b+1)_i = 0$. According to the proof of [4, Lemma 5], the switches need to be applied as follows: (See Footnote 2) "*In order to fulfill all side conditions for phase 3, we need to perform all switches from higher indices to smaller indices, and* k_i to $k_{\ell'}$ *before* $b_{i,r}^j$ *with* $(b+1)_{i+1} \neq j$ *or* $(b+1)_i = 0$ *to* $k_{\ell'}$". However, applying improving switches in this way results in another issue.[3]

Issue 4. Applying the improving switches as described in [4, Lemma 5] does not obey the LEAST-ENTERED pivot rule.

We can show an even stronger statement. (See Footnote 3) Friedmann applies improving switches of phase 3 as follows: During the transition from σ_b to σ_{b+1}, switches are applied "one level after another" where the order of the levels depends on $\ell(b+1)$. That is, depending on $\ell(b+1)$, an ordering $S^{\ell(b+1)}$ of the levels is considered and when i_1 appears before i_2 in $S^{\ell(b+1)}$, all switches in level i_1 need to be applied before any switch of level i_2. We prove that applying improving switches in this way violates the LEAST-ENTERED pivot rule at least once, independently of how $S^{\ell(b+1)}$ is chosen.

Issue 5. Consider some $b \in \mathbb{B}_n$ and the transition from σ_b to σ_{b+1}. Suppose that the switches of phase 3 are applied "level by level" according to any fixed ordering of the levels as described above. Further suppose that this ordering only depends on $\ell(b+1)$. Then, the LEAST-ENTERED pivot rule is violated.

Observe that Issue 5 rules out a broader class of orderings. In some sense, this shows that Friedmann's ordering needs to be changed fundamentally, and cannot be fixed by slight adaptation.

4.2 Fixing the Ordering of the Improving Switches

We now prove the existence of an ordering and an associated tie-breaking rule for the application of the switches of phase 3 that obey the LEAST-ENTERED rule. We then show that these can be used to prove the existence of an ordering and an associated tie-breaking rule that obey the LEAST-ENTERED rule for all phases that produces the intended behavior.

For every phase p policy σ, [4, Table 3] gives a subset L_σ^p and a superset U_σ^p of the improving switches I_σ for σ, see [4, Lemma 4]. The improving switch that

[3] The proof can be found in Appendix A.

is then applied in σ is always contained in L_σ^p, and U_σ^p is analyzed instead of I_σ to show that the intended switch can indeed be applied. Now, let σ be a phase 3 policy. We need to compare L_σ^3 and U_σ^3 since all switches that can possibly be applied during phase 3 are contained in U_σ^3. This is done via partitioning U_σ^3. The comparison then enables us to show that there is always a switch contained in L_σ^3 minimizing the occurrence record. This justifies that "*we will only use switches from L_σ^p*" [4, page 12] (at least for phase $p = 3$). We then show the following: All improving switches that should be applied during phase 3 according to [4] can be applied (in a different order) during phase 3, without violating the LEAST-ENTERED pivot rule.

As outlined in Sect. 2, the transition from σ_b to σ_{b+1} is partitioned into six phases. During the third phase, the MDP is reset, that is, some bicycles are opened and the targets of some entry vertices are changed. Therefore, a phase 3 policy σ is always associated with such a transition and we always implicitly consider the underlying transition from σ_b to σ_{b+1}.

We begin by further investigating the occurrence records of switches that should be applied during phase 3, i.e., we analyze the set L_σ^3. First, the occurrence record of these switches is bounded from above by the flip number $f(b, \ell')$.

Lemma 1. *Let σ be a phase 3 policy. Then* $\max_{e\in L_\sigma^3} \phi^\sigma(e) \leq f(b, \ell')$.

The following lemma gives a matching lower bound of $f(b, \ell')$ on all improving switches that should be applied *after* phase 3. It will also be used to estimate the occurrence records of possible improving switches contained in U_σ^3.

Lemma 2. *Let σ be a phase 3 policy. Assume that the Policy Iteration Algorithm is started with the policy σ^*. Then* $\min_{e\in L_\sigma^4 \cup L_\sigma^5 \cup L_\sigma^6} \phi^\sigma(e) \geq f(b, \ell')$.

We now partition U_σ^3 as follows (note that $U_\sigma^4 \subseteq U_\sigma^3$), cf. [4, Table 3]:

$$
\begin{aligned}
U_\sigma^{3,1} &\coloneqq \{(k_i, k_z) : \sigma(k_i) \notin \{k_z, k_{\ell'}\}, z \leq \ell' \wedge (b+1)_i = 0\} \\
U_\sigma^{3,2} &\coloneqq \{(b_{i,r}^j, k_z) : \sigma(b_{i,r}^j) \notin \{k_z, k_{\ell'}\}, z \leq \ell' \wedge (b+1)_i = 0\}\} \\
U_\sigma^{3,3} &\coloneqq \{(b_{i,r}^j, k_z) : \sigma(b_{i,r}^j) \notin \{k_z, k_{\ell'}\}, z \leq \ell' \wedge (b+1)_{i+1} \neq j\} \\
U_\sigma^{3,4} &\coloneqq U_\sigma^4
\end{aligned}
$$

Lemma 2 can be used to show that the occurrence records of edges contained in $U^{3,4}$ are too large. To be precise, no switch contained in one of this sets will be applied during phase 3 when following the LEAST-ENTERED rule.

Lemma 3. *Let σ be a phase 3 policy. Then, for all $e \in L_\sigma^3$ and $\tilde{e} \in I_\sigma \cap U_\sigma^{3,4}$, it holds that $\phi^\sigma(e) \leq \phi^\sigma(\tilde{e})$.*

It remains to analyze the sets $U_\sigma^{3,1}, U_\sigma^{3,2}$ and $U_\sigma^{3,3}$. We show that applying certain switches contained in L_σ^3 prevent other switches contained in these sets from being applied. To do so, we introduce subsets of $U_\sigma^{3,1}, U_\sigma^{3,2}$ and $U_\sigma^{3,3}$. The idea is to "slice" these sets such that for each slice, one improving switch prevents the whole slice from being applied. We thus define the following sets:

$$S^{3,1}_{i,\sigma} := \{(k_i, k_z) : \sigma(k_i) \notin \{k_z, k_{\ell'}\}, z \leq \ell' \wedge (b+1)_i = 0\} \subseteq U^{3,1}_\sigma$$
$$S^{3,2}_{i,j,r,\sigma} := \{(b^j_{i,r}, k_z) : \sigma(k_i) \notin \{k_z, k_{\ell'}\}, z \leq \ell' \wedge (b+1)_i = 0\} \subseteq U^{3,2}_\sigma$$
$$S^{3,3}_{i,j,r,\sigma} := \{(b^j_{i,r}, k_z) : \sigma(k_i) \notin \{k_z, k_{\ell'}\}, z \leq \ell' \wedge (b+1)_i \neq j\} \subseteq U^{3,3}_\sigma$$

The informal idea discussed previously is formalized by the following lemma.

Lemma 4. *The following statements hold.*

1. *Let σ be the phase 3 policy in which the switch $(k_i, k_{\ell'})$ is applied. Let σ' be a phase 3 policy of the same transition reached after σ. Then $I_{\sigma'} \cap S^{3,1}_{i,\sigma'} = \emptyset$.*
2. *Let σ be the phase 3 policy in which the improving switch $(b^j_{i,l}, k_{\ell'})$ with $\sigma(b^j_{i,l}) \neq k_{\ell'}$ and $(b+1)_i = 0$ is applied. Let σ' be a phase 3 policy of the same transition reached after σ. Then $I_{\sigma'} \cap S^{3,2}_{i,j,l,\sigma'} = \emptyset$.*
3. *Let σ be the phase 3 policy in which the improving switch $(b^j_{i,l}, k_{\ell'})$ with $\sigma(b^j_{i,l}) \neq k_{\ell'}$ and $(b+1)_{i+1} \neq j$ is applied. Let σ' be a phase 3 policy of the same transition reached after σ. Then $I_{\sigma'} \cap S^{3,3}_{i,j,l,\sigma'} = \emptyset$.*

This statement can then be used to prove the following lemma.

Lemma 5. *Let σ be a phase 3 policy. Then there is an edge $e \in L^3_\sigma$ minimizing the occurrence record among all improving switches.*

This lemma does not yet imply that all switches of phase 3 can be applied since it is not clear why it cannot happen that phase 4 is reached although not all switches of phase 3 were applied yet. However, the following theorem proves that this is impossible (See footnote 3).

Theorem 3. *There is an ordering of the improving switches and an associated tie-breaking rule compatible with the* Least-Entered *pivot rule such that all improving switches contained in $L^3_{\sigma_b}$ are applied and the* Least-Entered *pivot rule is obeyed during phase 3.*

Although Theorem 3 shows that the improving switches of phase 3 can be applied such that the Least-Entered rule is obeyed, it does not imply that the transition from σ_b to σ_{b+1} can be executed as intended in [4]. That is, it does not imply that the improving switches of the other phases can be applied as intended. This however can be shown using Theorem 3, yielding the following result.

Theorem 4. *Fix the transition from σ_b to σ_{b+1} for some $\sigma \in \mathbb{B}_n$. There is an order in which to apply improving switches during this transition such that the* Least-Entered *rule is obeyed, and the switches of phase p are applied before any switches of phase $p+1$, for every $p \in \{1, \ldots, 5\}$.*

Acknowledgments. The authors are very grateful to Oliver Friedmann for helpful comments and discussions, as well as support in using his implementation of the original construction to verify our findings.

A Proofs of Selected Statements

This section contains the proofs of the main statements. The proofs use the following two statements whose proofs can be found in [3].

Lemma A.1. *Let $i \in \{2, \dots, n-2\}$ and $l < i$. Then, there is a number $b \in \mathbb{B}_n$ with $\ell(b+1) = l$ such that for all $j \in \{i+2, \dots, n\}$, $\phi^{\sigma_b}(k_i, k_{\ell'}) < \phi^{\sigma_b}(k_j, k_{\ell'})$ and $(k_i, k_{\ell'}), (k_j, k_{\ell'}) \in L^3_{\sigma_b}$.*

Lemma A.2. *Assume that for any transition, the switches that should be applied during phase 3 were applied in some order. Let $i \in \{2, \dots, n-2\}$ and $l < i$. Then there is a $b \in \mathbb{B}_n$ with $\ell(b+1) = l$ such that $\phi^{\sigma_b}(k_{i+1}, k_{\ell'}) < \phi^{\sigma_b}(b^1_{i,r}, k_{\ell'})$, where $r \in \{0, 1\}$ is arbitrary and $(k_{i+1}, k_{\ell'}), (b^1_{i,r}, k_{\ell'}) \in L^3_{\sigma_b}$.*

We now prove the main statements of this paper.

Issue 4. Applying the improving switches as described in [4, Lemma 5] does not obey the Least-Entered pivot rule.

Proof. According to [4, Lemma 5], the switches of phase 3 should be applied as follows (See footnote 2): "*[...] we need to perform all switches from higher indices to smaller indices, and k_i to $k_{\ell'}$ before $b^j_{i,l}$ with $(b+1)_{i+1} \neq j$ or $(b+1)_i = 0$ to $k_{\ell'}$*".

Let $i \in \{2, \dots, n-2\}, l < i$ and $j \in \{i+2, \dots, n-2\}$. By Lemma A.1, there is a number $b \in \mathbb{B}_n$ such that $l = \ell(b+1)$ and $\phi^{\sigma_b}(k_i, k_{\ell'}) < \phi^{\sigma_b}(k_j, k_{\ell'})$. In addition, $(k_i, k_{\ell'}), (k_j, k_{\ell'}) \in L^3_{\sigma_b}$. Therefore, the switch $(k_j, k_{\ell'})$ should be applied before the switch $(k_i, k_{\ell'})$ during the transition from σ_b to σ_{b+1} when following the description of [4].

Consider the phase 3 policy σ of this transition in which the switch $(k_j, k_{\ell'})$ should be applied. Then, since $j > i$ and we "*perform all switches from higher indices to smaller indices*", the switch $(k_i, k_{\ell'})$ was not applied yet. However, it still is an improving switch for the policy σ. This implies $\phi^{\sigma_b}(k_j, k_{\ell'}) = \phi^{\sigma}(k_j, k_{\ell'})$ and $\phi^{\sigma_b}(k_i, k_{\ell'}) = \phi^{\sigma}(k_i, k_{\ell'})$. Consequently, $\phi^{\sigma_b}(k_i, k_{\ell'}) < \phi^{\sigma_b}(k_j, k_{\ell'})$ implies that $\phi^{\sigma}(k_i, k_{\ell'}) < \phi^{\sigma}(k_j, k_{\ell'})$. Thus, since the edge $(k_i, k_{\ell'})$ is an improving switch for σ having a lower occurrence record than $(k_j, k_{\ell'})$ and σ was chosen as the policy in which $(k_j, k_{\ell'})$ should be applied, the Least-Entered rule is violated. □

Issue 5. Consider some $b \in \mathbb{B}_n$ and the transition from σ_b to σ_{b+1}. Suppose that the switches of phase 3 are applied "level by level" according to any fixed ordering of the levels as described above. Further suppose that this ordering only depends on $\ell(b+1)$. Then, the Least-Entered pivot rule is violated.

Proof. To prove Issue 5, we show that applying the improving switches as discussed before violates Zadeh's Least-Entered rule several times by showing the following statement: Let S^i be an ordering of $\{1, \dots, n\}$ for $i \in \{1, \dots, n\}$. Suppose that the improving switches of phase 3 of the transition from σ_b to σ_{b+1} are applied in the order defined by $S^{\ell(b+1)}$ for all $b \in \mathbb{B}_n$. Then, for every possible least significant bit $l \in \{1, \dots, n-4\}$, assuming that the ordering S^l obeys the Least-Entered rule results in a contradiction.

Fix some $l \in \{1, \dots, n-4\}$. Consider the ordering $S^l = (s_1, \dots, s_n)$. For any $k \in \{1, \dots, n\}$, we denote the position of k within S^l by $k^\star$. Towards a contradiction, assume that applying the improving switches level by level according to the ordering S^l obeys the Least-Entered rule. We show that this assumption yields $(l+1)^\star < (n-1)^\star$ and $(n-1)^\star < (l+1)^\star$.

Let $i \in \{l+1, \dots, n-2\}$. Then, $i > l$ and therefore, by Lemma A.2, there is a number $b \in \mathbb{B}_n$ with $\ell(b+1) = \ell' = l$ and $\phi^{\sigma_b}(k_{i+1}, k_{\ell'}) < \phi^{\sigma_b}(b^1_{i,r}, k_{\ell'})$ such that $(k_{i+1}, k_{\ell'}), (b^0_{i,r}, k_{\ell'}) \in L^3_{\sigma_b}$. Thus, both switches need to be applied during the transition from σ_b to σ_{b+1}. Because of $\phi^{\sigma_b}(k_{i+1}, k_{\ell'}) < \phi^{\sigma_b}(b^1_{i,r}, k_{\ell'})$, level $i+1$ needs to appear before level i within the ordering S^l. Since this argument can be applied for all $i \in \{l+1, \dots, n-2\}$, the sequence

$$(n-1, n-2, \dots, l+1)$$

needs to be a (not necessarily consecutive) subsequence of S^l. In particular, $(n-1)^\star < (l+1)^\star$ since $l+1 \neq n-1$ by assumption.

Let $i = l+1$ and $j \in \{i+2, \dots, n\}$. Then, by Lemma A.1, there is a number $b \in \mathbb{B}_n$ with $\ell(b+1) = l$ such that $\phi^{\sigma_b}(k_i, k_{\ell'}) < \phi^{\sigma_b}(k_{i+2}, k_{\ell'})$ and $(k_i, k_{\ell'}), (k_{i+2}, k_{\ell'}) \in L^3_{\sigma_b}$. Now, both switches need to be applied during the transition from σ_b to σ_{b+1}. Therefore, for all $i \in \{l+1, \dots, n-2\}$, level i needs to appear before any of the levels level $j \in \{i+2, \dots, n\}$ within S^l. But this implies that the sequence

$$(l+1, l+3, l+4, \dots, n-1, n)$$

needs to be a (not necessarily consecutive) subsequence of S^l. In particular, $(l+1)^\star < (n-1)^\star$ since $n-1 \geq l+3$ as we have $l \leq n-4$ by assumption. This however contradicts $(n-1)^\star < (l+1)^\star$. □

Theorem 3. *There is an ordering of the improving switches and an associated tie-breaking rule compatible with the* Least-Entered *pivot rule such that all improving switches contained in $L^3_{\sigma_b}$ are applied and the* Least-Entered *pivot rule is obeyed during phase 3.*

Proof. Let σ denote the first phase 3 policy of the transition from σ_b to σ_{b+1}. Then, $L^3_\sigma = L^3_{\sigma_b}$. By Lemma 5, there is an edge $e_1 \in L^3_\sigma$ minimizing the occurrence record I_σ. Applying this switch results in a new phase 3 policy $\sigma[e_1]$ such that $L^3_{\sigma[e_1]} = L^3_\sigma \setminus \{e_1\}$. Now, again by Lemma 5, there is an edge $e_2 \in L^3_{\sigma[e_1]}$ minimizing the occurrence record $I_{\sigma[e_1]}$.

We can now apply the same argument iteratively until we reach a phase 3 policy $\hat{\sigma}$ such that $\left|L^3_{\hat{\sigma}}\right| = 1$ while only applying switches contained in $L^3_{\sigma_b}$. Then, by construction and by Lemma 5, $(e_1, e_2, \dots)$ defines an ordering of the edges of $L^3_{\sigma_b}$ and an associated tie-breaking rule that always follow the Least-Entered rule. When the policy $\hat{\sigma}$ with $\left|L^3_{\hat{\sigma}}\right| = 1$ is reached, applying the remaining improving switch results in a phase 4 policy. Then, all improving switches contained in $L^3_{\sigma_b}$ were applied and the Least-Entered pivot rule was obeyed. □

References

1. Avis, D., Chvátal, V.: Notes on Bland's pivoting rule. In: Balinski, M.L., Hoffman, A.J. (eds.) Polyhedral Combinatorics. Mathematical Programming Studies, vol. 8, pp. 24–34. (1978). https://doi.org/10.1007/BFb0121192
2. Dantzig, G.B.: Linear Programming and Extensions. Princeton University Press, Princeton (1963)
3. Disser, Y., Hopp, A.V.: On Friedmann's subexponential lower bound for Zadeh's pivot rule. Technical report, Technische Universität Darmstadt (2018). http://tuprints.ulb.tu-darmstadt.de/id/eprint/7557
4. Friedmann, O.: A subexponential lower bound for Zadeh's pivoting rule for solving linear programs and games. In: Proceedings of the 15th International Conference on Integer Programming and Combinatoral Optimization (IPCO), pp. 192–206 (2011)
5. Goldfarb, D., Sit, W.Y.: Worst case behavior of the steepest edge simplex method. Discrete Appl. Math. **1**(4), 277–285 (1979)
6. Howard, R.A.: Dynamic Programming and Markov Processes. MIT Press, Cambridge (1960)
7. Jeroslow, R.G.: The simplex algorithm with the pivot rule of maximizing criterion improvement. Discrete Math. **4**(4), 367–377 (1973)
8. Klee, V., Minty, G.J.: How good is the simplex algorithm? In: Inequalities III, pp. 159–175. Academic Press, New York (1972)
9. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York (2005)
10. Thomas, A.: Unique sink orientations: complexity, structure and algorithms. Ph.D. thesis, ETH Zurich (2017)
11. Zadeh, N.: What is the worst case behavior of the simplex algorithm? Technical report 27, Departments of Operations Research, Stanford (1980)