

A TIGHT BOUND FOR SHORTEST AUGMENTING PATHS ON TREES*

BARTŁOMIEJ BOSEK, DARIUSZ LENIOWSKI, PIOTR SANKOWSKI, AND ANNA ZYCH-PAWLEWICZ

ABSTRACT. The shortest augmenting path technique is one of the fundamental ideas used in maximum matching and maximum flow algorithms. Since being introduced by Edmonds and Karp in 1972, it has been widely applied in many different settings. Surprisingly, despite this extensive usage, it is still not well understood even in the simplest case: online bipartite matching problem on trees. In this problem a bipartite tree $T = (W \uplus B, E)$ is being revealed online, i.e., in each round one vertex from B with its incident edges arrives. It was conjectured by Chaudhuri et. al. [CDKL09] that the total length of all shortest augmenting paths found is $O(n \log n)$. In this paper we prove a tight $O(n \log n)$ upper bound for the total length of shortest augmenting paths for trees improving over $O(n \log^2 n)$ bound [BLSZ15].

1. INTRODUCTION

One of the most fundamental techniques used to solve maximum matchings or flow problems is the augmenting path technique. It augments the solution along residual paths until the maximum size matching/flow is found. Intuitively, the work needed for that should be minimized if shortest paths are chosen each time. In particular, this was the key concept that allowed Edmonds and Karp in 1972 to show the first strongly polynomial time algorithm for the maximum flow problem [EK72]. Since then it has been widely applied. Surprisingly, despite this effort, it is still not well understood even in the simplest case — online bipartite matching problem on trees. This may be due to the fact that shortest augmenting paths do not seem to have strong enough structure admitting exact analysis. Other methods for choosing augmenting paths are easier to analyze [BLSZ14, CDKL09]. Our work is meant as a step forward towards understanding the shortest augmenting path method for computing the matching on bipartite graphs.

To be able to analyze this approach we adopt the following model. Let W and B be the bipartition of vertices over which the tree will be formed. The set W (called white vertices) is given up front to the algorithm, whereas the vertices in B (black vertices) arrive online. We denote by $F_t = \langle W \uplus B_t, E_t \rangle$ the forest after the t 'th black vertex has arrived where $X \uplus Y$ is a disjoint sum of X and Y . The graphs F_t for $t \in [n] = \{1, \dots, n\}$ are constructed online in the following manner. We start with $F_0 = \langle W \uplus B_0, E_0 \rangle = \langle W \uplus \emptyset, \emptyset \rangle$. In turn $t \in [n]$ a new vertex $b_t \in B$ together with all its incident edges $E(b_t)$ is revealed and F_t is defined as: $E_t = E_{t-1} \cup E(b_t)$ and $B_t = B_{t-1} \cup \{b_t\}$. In the model we consider, none of the newly added edges is allowed to close the cycle. For simplicity we assume that we add in total $n = |W|$ black vertices. The final graph is a tree denoted as $F_n = (W \uplus B_n, E_n)$.

The goal of the online algorithm is to compute for each F_t the maximum size matching M_t , possibly making use of M_{t-1} . In this paper we study one specific algorithm, referred to as the Shortest Augmenting Path algorithm. When b_t arrives, the Shortest Augmenting Path algorithm always chooses the shortest among all available augmenting paths. A natural question that we ask is what is the total length of all paths applied by the Shortest Augmenting Path algorithm. In this paper the unmatched vertices are referred to as *free*. For a vertex v we denote its neighborhood in F_t as $N_t(v)$. By $F[X] = \langle X, E(X) \rangle$ we denote a subgraph of F induced by $X \subseteq W \cup B$, where $E(X) = \{e \in E : e \subseteq X\}$.

*The work of all authors was supported by Polish National Science Center grant 2013/11/D/ST6/03100. Additionally, the work of P. Sankowski was partially supported by the project TOTAL (No 677651) that has received funding from ERC.

2. MOTIVATION AND RELATED WORK

The online bipartite matching problem with augmentations has recently received increasing attention [BHR17, BLSZ14, BLSZ15, CDKL09, GKKV95, GKS14]. The model we study has been introduced in [GKKV95]. As mentioned before, the key point of this model is to focus on bounding the total length of augmenting paths and not the running time of the algorithm. This is motivated as follows. Imagine that the white vertices are servers and black vertices are clients. The clients arrive online. A typical client may be a portable computing box, trying to connect to a huge network of services with some specific request. The edges of the graph reflect eligibility of the servers to answer clients request. The classical online model (as in [BM08, DJK13, KVV90]) does not allow preemption, i.e., the client cannot change the server. In such setting one must accept some clients not being served while they could possibly be served with preemption. In that model a famous ranking algorithm gives an optimal $(1 - 1/e)$ -approximation [KVV90]. The authors in [GKKV95] wonder if preemption makes sense. It may be beneficial to reallocate clients provided that only a limited number of reallocations is needed. This leads to the question of how many reallocations are needed if one insists on serving every client. In [GKKV95] a special case is studied when each client can connect to at most two servers. In such scenario the authors prove that the Shortest Augmenting Path algorithm performs $\mathcal{O}(n \log n)$ reallocations and that no algorithm can do better than that. Chaudhuri et al. [CDKL09] show that the Shortest Augmenting Path algorithm makes a total of $\mathcal{O}(n \log n)$ reallocations in the case of general bipartite graph, provided that the clients arrive in a random order. They conjecture, however, that this should be the case also for the worst case arriving order of clients. Until this paper, this conjecture remained open even for trees. In [BLSZ15] the authors prove a bound of $\mathcal{O}(n \log^2 n)$ for Shortest Augmenting Path algorithm given that the underlying graph is a tree. In this paper we take a different approach and prove the conjecture of Chaudhuri et al. for trees. In this restricted case, the authors of [CDKL09] proposed a different augmenting path algorithm that achieves total paths' length of $\mathcal{O}(n \log n)$. Their algorithm, however, is only applicable to trees. The Shortest Augmenting Path algorithm, on the other hand, applies to any bipartite graph and also is very simple. This is the reason why we feel it is important to study this algorithm. Our ultimate goal is to show the bound of $\mathcal{O}(n \log n)$ for general bipartite graphs. We believe that the techniques proposed in this paper are an important step forward on the path to achieve this goal. In parallel work to ours [BHR17] the authors provide a bound of $\mathcal{O}(n \log^2 n)$ total number of reallocations for the Shortest Augmenting Path algorithm on general bipartite graphs. This recent result has been accepted to *SODA 2018* and it nearly closes the conjecture of Chaudhuri et al. We note, however, that their techniques alone do not lead to $\mathcal{O}(n \log n)$ even for trees. Before this result, for general graphs, nothing interesting was known for Shortest Augmenting Path algorithm. A different algorithm was proposed achieving much worse $\mathcal{O}(n\sqrt{n})$ bound on the total length of augmenting paths [BLSZ14].

Our model is strongly related to dynamic algorithms. There, we are not only interested in constructing short augmenting paths. An efficient way of finding them is the most important aspect. Most papers in this area consider edge updates in a general fully-dynamic model which allows for insertions and deletions of edges intermixed with each other. This is a much more difficult scenario in which one cannot do much when constrained by our model. In particular, if edges are added to a bipartite graph, one can show an instance for which any algorithm maintaining a maximum matching performs $\Omega(n^2)$ reallocations. Hence, it is reasonable to stop insisting on matching every client and accept approximate solutions. Here we want to approximate the maximum matching size and not the number of reallocations. One also needs to keep in mind that a trivial greedy algorithm maintaining a maximal matching gives a $1/2$ -approximation and performs no reallocations at all. A $2/3$ -approximation algorithm by [NS13] achieves $\mathcal{O}(\sqrt{m})$ update time. Gupta and Peng give a $(1 - \varepsilon)$ -approximation in $\mathcal{O}(\sqrt{m}\varepsilon^{-2})$ time per update [GP13]. The $\mathcal{O}(\sqrt{m})$ barrier was broken by Bernstein and Stein who gave a $(\frac{2}{3} - \varepsilon)$ -approximation algorithm that achieves $\mathcal{O}(m^{1/4}\varepsilon^{-2.5})$ update time [BS15]. Finally, $(1 - \varepsilon)$ approximation in $\mathcal{O}(m\varepsilon^{-1})$ total time and with $\mathcal{O}(n\varepsilon^{-1})$ total length of paths was shown in [BLSZ14] in a model most related to ours, i.e., when vertices are added on one side of the

bipartition. There are also randomized algorithms in the dynamic model [San07] maintaining the exact size of a maximum matching with $\mathcal{O}(n^{1.495})$ update time. They do not imply any bound on the number of changes to the matching as they use algebraic techniques that are not based on augmenting paths.

3. THE MINI-MAX GAME

Our goal in this paper is to prove that the total length of all augmenting paths applied by Shortest Augmenting Path algorithm on a tree is $\mathcal{O}(n \log n)$. More formally, we want to prove the following, where by $\|\pi\|$ we denote the number of edges on a path π .

Theorem 1. *Let π_t be the path applied by Shortest Augmenting Path algorithm in turn t . Then $\sum_{t=1}^n \|\pi_t\| \in \mathcal{O}(n \log n)$.*

The idea is not to study directly the paths applied by Shortest Augmenting Path algorithm, but a collection of other paths that are possibly longer. To be more precise, we model a scenario where in each turn Shortest Augmenting Path algorithm gets the worst possible matching on F_t (i.e., the one maximizing the shortest augmenting path). We then study the worst case augmenting paths rather than the ones given by the matching produced by Shortest Augmenting Path algorithm. Interestingly, these paths can be defined without mentioning any matching. In this section we provide the appropriate definitions and show that they work as expected.

Let us consider what worst possible matching could there be. Think of a game, where the algorithm chooses a shortest augmenting path, and the adversary chooses a matching where such path is the longest. We are given graph F_t and the newly presented vertex b_t . We are interested in a matching where b_t is not matched, so that we model the worst case matching before b_t is matched. The game starts in vertex b_t , where the algorithm may choose which edge to follow among the unmatched edges incident to b_t . Then the algorithm stumbles upon a white vertex where it has to follow the matching edge chosen by the adversary. The game continues until a leaf is reached (either black or white, black meaning that the algorithm did not find a path). It is not hard to see that the algorithm, when it has a choice, wants to minimize the distance to a free white vertex, while the adversary tries to maximize it. This way we obtain a two-person game, where the outcome of the game is the length of the shortest augmenting path. If the path does not exist, we let the outcome be infinite. Throughout the paper we let $\infty + 1 = \infty$ and we write $x < \infty$ to indicate that x is simply an integer.

We move on to stating formal definitions. We start by introducing our game on any rooted tree T whose vertices are either black or white. We then define the outcome of the algorithm player in time t for a specific T closely related to F_t . For a rooted tree T we denote the list of children of a vertex v in T as $\text{Ch}_T(v)$ and a parent of v as $\text{parent}_T(v)$.

Definition 1. Let T be a rooted tree whose vertices are partitioned into two sets: $V(T) \subseteq B \uplus W$.

(1) For each $b \in B$ we define its revenue as

$$\text{mini-max}_T(b) = \begin{cases} \min_{w \in \text{Ch}_T(b)} \text{mini-max}_T(w) + 1 & \text{if } \text{Ch}_T(b) \neq \emptyset \\ \infty & \text{otherwise} \end{cases}$$

(2) and for each $w \in W$ we define its revenue as

$$\text{mini-max}_T(w) = \begin{cases} \max_{b \in \text{Ch}_T(w)} \text{mini-max}_T(b) + 1 & \text{if } \text{Ch}_T(w) \neq \emptyset \\ 0 & \text{otherwise.} \end{cases}$$

(3) We let $\text{mini-max-next}_T(v)$ be the child of v whose revenue determines the minimum or the maximum respectively.¹ If v has no children, $\text{mini-max-next}_T(v)$ is undefined.

(4) We define the mini-max path starting in a vertex v as

$$\text{mini-max-path}_T(v) = \begin{cases} v \cdot \text{mini-max-path}_T(\text{mini-max-next}_T(v)) & \text{if } \text{mini-max-next}_T(v) \text{ is defined,} \\ v & \text{otherwise.} \end{cases}$$

¹If there are more such vertices we choose the first one according to some predefined order on $B \cup W$.

²Symbol \cdot denotes concatenation of paths.

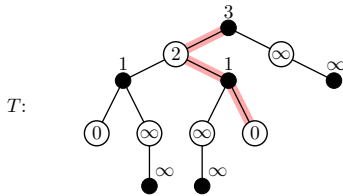


FIGURE 1. Example of a rooted mini-max tree with vertex revenues.

Definition 1 is illustrated by example in Figure 1. Based on this definition we can define the first and the second mini-max distance from a given vertex to a white leaf in a specific time moment t . In addition to that we define the first and second direction, i.e., the vertex one needs to follow to find the first and second mini-max distance.

Definition 2. Let $t \in [n]$ and $v \in B_t \cup W$. Let T be the connected component of v in F_t rooted in v . Let

$$\begin{aligned} \text{dist}_t(v) &= \text{mini-max}_T(v), \\ \text{dir}_t(v) &= \text{mini-max-next}_T(v). \end{aligned}$$

Let now S be a rooted tree, where from T we remove $\text{mini-max-next}_T(v)$ and all its descendants. Let

$$\begin{aligned} \text{sec-dist}_t(v) &= \text{mini-max}_S(v), \\ \text{sec-dir}_t(v) &= \text{mini-max-next}_S(v). \end{aligned}$$

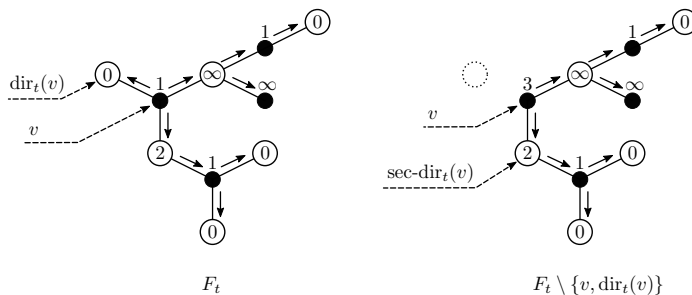


FIGURE 2. Example of mini-max distances for a vertex v in turn t .

Definition 2 is illustrated by example in Figure 2. We next observe the monotonicity of the mini-max distance functions we defined.

Observation 2. Fix a vertex $u \in B \cup W$. The functions $\text{dist}_t(u)$ and $\text{sec-dist}_t(u)$ are non-decreasing with respect to t for the whole range of t where $u \in V(F_t)$.

Proof. Observe that if T is a component of v in F_{t-1} rooted in v , then adding b_t causes that b_t possibly becomes a child of some white vertex $w \in V(T)$ (see Figure 10 to the left in Appendix B). Such change causes that either we take maximum over a larger set than before, or we take maximum/minimum over a set whose values do not decrease. In all these cases we cannot decrease the mini-max values. For the rigorous proof see Appendix A. \square

We intuitively explained how the mini-max distances correspond to the augmenting paths of the worst case matching, so the hope is that they bound from above the augmenting paths applied by Shortest Augmenting Path algorithm. The next lemma shows that this intuition is reflected in reality. It states that no matter what matching is given on F_t for some $t \in [n]$, the path chosen by Shortest Augmenting Path algorithm to match b_t is bounded by $\text{dist}_t(b_t)$.

Lemma 3. *Let $1 \leq t_0 \leq t \leq n$ and let ρ_t be the shortest augmenting path from b_{t_0} to a free white vertex according to any given matching M in F_t where b_{t_0} is free. It holds that if $\text{dist}_t(b_{t_0}) < \infty$ then ρ_t exists and $\|\rho_t\| \leq \text{dist}_t(b_{t_0})$.*

Proof. For the proof we refer to Appendix A. □

4. DEAD VERTICES

In this section we introduce another concept crucial for our result. We define here *dead* vertices and give some intuition why this makes sense. In fact dead vertices reflect the infinity of some mini-max distance functions. For completeness, in addition to defining dead vertices, we describe the situations when the mini-max distance functions are infinite. We start with the statements.

Definition 3. A vertex $b_{t_0} \in B_t$ breaks Hall's condition in time $t \geq t_0$ iff there exists $X \subseteq B_t$ such that

- (1) $b_{t_0} \in X$,
- (2) $|\mathcal{N}_t(X)| < |X|$, and
- (3) X is minimal under inclusion set satisfying (2).

Lemma 4. *Let $1 \leq t_0 \leq t \leq n$. Then $\text{dist}_t(b_{t_0}) = \infty$ iff b_{t_0} breaks Hall's condition in time t .*

Proof. " \Leftarrow ": Assume that $\text{dist}_t(b_{t_0}) < \infty$. Then, by Lemma 3, an augmenting path exists from b_{t_0} for a maximum matching M where b_{t_0} is free. By Hall's theorem this implies that b_{t_0} does not break Hall's condition. " \Rightarrow ": Deferred to Appendix B □

Lemma 4 reveals the following corollary.

Corollary 5. If $\text{dist}_t(b_t)$ is infinite, and we are given some maximum matching M_{t-1} for F_{t-1} , then there is no augmenting (with respect to M_{t-1}) path for b_t in time t .

Proof. Follows directly from Hall's theorem. □

We now move on to defining dead vertices. The definitions may not seem very intuitive, but we provide some intuition shortly after introducing them.

Definition 4. We say that a vertex $b_{t_0} \in B$ is *dead* in turn $t \geq t_0$ iff $\text{sec-dist}_t(b_{t_0}) = \infty$.

Definition 4 combined with Lemma 4 implies, that any black vertex that breaks Hall's condition in time t is dead in time t , but not necessarily the other way around.

Definition 5. A white vertex $w \in W$ is *dead* in time t iff $\text{dist}_t(w) = \infty$.

We say that a vertex is alive iff it is not dead. We denote as A_t the set of vertices of $B_t \cup W$ that are alive in turn t and as D_t the set of vertices of $B_t \cup W$ that are dead in turn t . If $v \in A_{t-1} \cap D_t$, we say that v dies in turn t . Note that due to monotonicity (Observation 2), once a vertex dies, it never comes back alive. The following observations bring some intuition into the picture of dead versus alive vertices. Observation 6 given below follows from Definitions 2, 4 and 5.

Observation 6.

- (1) A black leaf is dead from the moment it arrives.
- (2) A black vertex is dead iff it has at most one alive neighbour.
- (3) A white vertex is dead iff it has at least one dead neighbour.

The intuition behind dead vertices is that they determine regions of F_t where Hall's condition is either broken or tight. The mini-max paths in turn t that correspond to finite mini-max distances do not visit vertices that were dead in turn $t - 1$. Moreover, if a mini-max path in F_t (whose corresponding mini-max distance is finite) enters a vertex that is alive in turn t , it does not visit anymore vertices dead in turn t . We state this formally as Lemma 7. This reflects the behavior of augmenting paths. If a maximum matching is maintained, then the augmenting path from turn t does not enter the regions where Hall's condition is tight in F_{t-1} .

Lemma 7. *Let $t \in [n]$ and $v \in A_t$. Pick any vertex as a root of the connected component of v in F_t and let T be the corresponding rooted tree. Then $V(\text{mini-max-path}_T(v)) \subseteq A_t$.*

Proof. See Appendix B. □

In the remainder of this section we specify precisely which vertices die in turn t . The first lemma does not describe who dies or stays alive, but it is an important complement of the subsequent two lemmas, which cover all the situations when vertices die.

Lemma 8. *If b_t does not break Hall's condition in turn t , then b_t has at least one neighbour in F_t which was alive in turn $t - 1$.*

Proof. See Appendix B. □

So if a black vertex added in turn t does not break Hall's condition, then it has at least one neighbour who was alive in turn $t - 1$. The next two lemmas cover two cases. The first lemma states that if the new black vertex has at least two such neighbours, then no vertices die in turn t . If, however, it has exactly one such neighbour, then some vertices die in turn t and the second lemma describes precisely which ones.

Lemma 9. *If b_t has at least two neighbours which are alive in turn $t - 1$ then b_t is alive in turn t and no vertex dies in turn t .*

Proof. See Appendix B. □

The last lemma covers the only case when vertices die in turn t . It shows that there is a certain region around b_t where vertices die, and a barrier for that region are special vertices called life portals, defined below. The picture illustrating which region dies in turn t is given in Figure 3.

Definition 6. A black vertex b is a life portal in turn t iff $|N_t(b) \cap A_{t-1}| \geq 3$. The set of life portals in turn t is denoted as \mathcal{LP}_t .

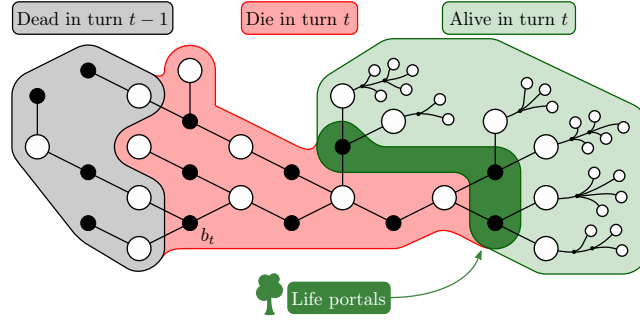


FIGURE 3. The vertices that die in turn t .

Lemma 10. *If b_t has exactly one neighbour in F_t which was alive in turn $t - 1$ and there is a path π from b_t to $v \in B_t \cup W$ such that*

- (1) *all vertices of π were alive in turn $t - 1$ and*
- (2) *there are no life portals from \mathcal{LP}_t on π ,*

then v dies in turn t . Vertices of $B_t \cup W$ that cannot be reached from b_t via such path do not die in turn t .

Proof. See Appendix B. □

Corollary 11. Lemma 10 shows, that for all t such that b_t does not break Hall's condition and has exactly one neighbour alive in turn $t - 1$, statement $|N_t(b) \cap A_{t-1}| \geq 3$ in Definition 6 is equivalent to $|N_t(b) \cap A_t| \geq 2$.

5. THE PROOF

In the remainder of the paper we present the proof of Theorem 1, which states that if π_t is the path applied by Shortest Augmenting Path algorithm in turn t , then $\sum_{t=1}^n \|\pi_t\| \in \mathcal{O}(n \log n)$. As we mentioned in Section 3, we do not study $\|\pi_t\|$ directly. Instead, we want to study distance functions $\text{dist}_t(b_t)$ introduced in Section 3. By Lemma 3 given in Section 3 we know that $\text{dist}_t(b_t)$ bounds $\|\pi_t\|$ from above. Recall that $\text{dist}_t(b_t)$ is the mini-max distance from b_t to a white leaf in F_t . By definition if $\text{dist}_t(b_t) < \infty$, then there is a path from b_t to a white leaf which certifies it. We introduce the formal definition of such path below.

Definition 7. Let $t \in [n]$ and $v \in B_t \cup W$. Let T be a connected component of v in F_t rooted at v . Then $\text{path}_t(v) = \text{mini-max-path}_T(v)$.

Note that by Definitions 1 and 2, if $\text{dist}_t(b_t) < \infty$, then $\|\text{path}_t(b_t)\| = \text{dist}_t(b_t)$. In addition to that, we define a path that certifies that $\text{sec-dist}_t(b_t)$ is finite.

Definition 8. Let $t \in [n]$ and $v \in B_t \cup W$. Let S be a connected component of v in $F_t - \{v, \text{dir}_t(v)\}$ rooted at v , where $F_t - \{v, \text{dir}_t(v)\}$ denotes F_t with edge $\{v, \text{dir}_t(v)\}$ removed. Then $\text{sec-path}_t(v) = \text{mini-max-path}_S(v)$.

Again by Definitions 1 and 2, if $\text{sec-dist}_t(b_t) < \infty$, then $\|\text{sec-path}_t(b_t)\| = \text{sec-dist}_t(b_t)$. Instead of proving Theorem 1, in the remainder of this paper we prove that $\sum_{t: \text{dist}_t(b_t) < \infty} \|\text{path}_t(b_t)\| \in \mathcal{O}(n \log n)$. This is in fact a stronger statement. We claim that even if the adversary picks the worst possible maximum matching in each turn, the Shortest Augmenting Path algorithm still applies paths of total length $\mathcal{O}(n \log n)$. Note that if $\text{dist}_t(b_t) = \infty$, then due to Corollary 5 Shortest Augmenting Path algorithm cannot match the new vertex b_t if the maximum matching is given on the remaining vertices. Our proof of such simple statement is unfortunately rather complex. Before we move on to it, we give some intuitions on where the actual problem hides. It is enlightening to discover, that with the additional assumption that the black vertices are of degree two or more, the statement above has a very simple proof.

Lemma 12. *If each black vertex b_t has degree at least 2, then $\sum_{t=1}^n \|\text{path}_t(b_t)\| \leq n \log_2 n$.*

Proof. We start by observing that no vertex ever dies. In turn $t = 0$ the only presented vertices are W , so by definition all vertices are alive in turn $t = 0$. Let $t > 0$ and assume that no vertices died up until turn $t - 1$. Due to Lemma 9 no vertex dies in turn t and b_t is alive in turn t . This implies that $\text{dist}_t(b_t) < \infty$ and $\text{sec-dist}_t(b_t) < \infty$. Hence, $\text{path}_t(b_t)$ and $\text{sec-path}_t(b_t)$ are two separate paths, contained in two different components of F_{t-1} connected in turn t by b_t . Also, $\|\text{path}_t(b_t)\| \leq \|\text{sec-path}_t(b_t)\|$. Thus, $\text{path}_t(b_t)$ is at most as long as the size of the smaller of the two components. We pay for $\text{path}_t(b_t)$ by charging 1 token to each vertex in every component but the largest one among the components of F_{t-1} connected by b_t in turn t . A vertex v is charged when v 's component size increases at least twice, so v cannot be charged more than $\log_2 n$ times. This gives a total charge of at most $n \log_2 n$. \square

The essence of this proof is that every time a black vertex is added, it connects at least two trees into one. As a consequence there are at least two alternative mini-max paths starting from the added vertex, each in a separate tree. The length of the shorter of the two can be charged to the vertices of the smaller tree. If we allow black vertices of degree 1, the situation becomes more complicated, because: (1) there is no alternative path, i.e., the path needs to follow the only edge adjacent to the newly added black vertex, and (2) no trees are merged. Nevertheless the proof of Theorem 1 is a generalization of the proof of Lemma 12. The majority of the remainder of this paper is devoted to addressing issue (2). We define trees which are merged in each turn and allow introducing the charging scheme that generalizes the scheme of Lemma 12. We start, though, by addressing issue (1). To that end we introduce a concept of a dispatching vertex. Even though b_t does not necessarily fork into two alternative mini-max paths, there is another vertex which does. It is the first life portal on $\text{path}_t(b_t)$. We refer to it as dispatching vertex in turn t . To be more formal, we introduce the following definition.

Definition 9. The dispatching vertex at time $t \in [n]$ is the first black vertex on $\text{path}_t(b_t)$ such that $|\mathcal{N}_t(b) \cap A_t| \geq 2$. We denote it as \check{b}_t .

First observe that \check{b}_t has two alive neighbours in turn t , so there are two alternative mini-max paths branching from \check{b}_t . Our next observation is that if $\check{b}_t \neq b_t$, then $\check{b}_t \in \mathcal{LP}_t$: if b_t has two neighbours alive in $t-1$, then due to Lemma 9 no vertex dies in turn t so b_t has two neighbours alive in turn t and hence $b_t = \check{b}_t$; otherwise, if b_t has one neighbour alive in turn $t-1$, then due to Lemma 10 and Corollary 11 it holds that \check{b}_t is the first life portal of \mathcal{LP}_t on $\text{path}_t(b_t)$. Then also \check{b}_t is the first vertex on $\text{path}_t(b_t)$ that remains alive. All vertices that follow \check{b}_t on $\text{path}_t(b_t)$ remain alive as well. It may happen that $\text{path}_t(b_t)$ contains no life portals, in which case there is no dispatching vertex defined in turn t . This case however is not of concern, since the whole $\text{path}_t(b_t)$ dies in turn t due to Lemma 10. In general, we do not have to worry about vertices that die, and we state this observation as Observation 13 preceded by Definition 10.

Definition 10. Let $t \in [n]$ be such that $\text{dist}_t(b_t) < \infty$. We let $\text{path}_t(b_t) = \text{path}_t^p(b_t) \cdot \text{path}_t^s(b_t)$, where $\text{path}_t^p(b_t)$ is the prefix of $\text{path}_t(b_t)$ that dies (possibly empty) and $\text{path}_t^s(b_t)$ is the corresponding suffix (also possibly empty).

Note that if \check{b}_t is defined then $\text{path}_t^s(b_t)$ begins with \check{b}_t . Since the final forest has $2n$ vertices and each can die only once, we have the following:

Observation 13. $\sum_{t:\text{dist}_t(b_t) < \infty} \|\text{path}_t^p(b_t)\| \leq 2n$.

Thus, to bound $\sum_{t:\text{dist}_t(b_t) < \infty} \|\text{path}_t(b_t)\|$ it suffices to bound $\sum_{t:\text{dist}_t(b_t) < \infty} \|\text{path}_t^s(b_t)\|$. We conclude the list of properties of the dispatching vertex with the following observation.

Observation 14. Let $t \in [n]$ and $\text{dist}_t(b_t) < \infty$ and \check{b}_t is defined. Then $\|\text{path}_t^s(b_t)\| = \text{dist}_t(\check{b}_t)$.

Proof. See Appendix C. \square

To proceed further, we introduce the crucial notion in our proof: the notion of a level. The levels are some numbers assigned to vertices: each vertex is assigned its level. The intuitive meaning of the level of a vertex is the following. Consider $\text{path}_t(b_t)$, which is the worst case shortest augmenting path for a black vertex b_t added in turn t . For a vertex v , if $\text{path}_t(b_t)$ crosses v , level in F_t returns the value representing the length of the suffix of $\text{path}_t(b_t)$ starting in v . Formally, the level function is defined in the following way.

Definition 11. For $v \in W \cup B$ and $t \in \{0, \dots, n\}$ let

$$\text{level}_t(v) = \begin{cases} \text{sec-dist}_t(v) & v \in W, \\ \text{dist}_t(v) & v \in B_t, \\ 0 & \text{otherwise.} \end{cases}$$

It may at first seem confusing that the level of a white vertex is the second maximum distance to a leaf. It is defined this way because, surprisingly, in every turn t the path $\text{path}_t(b_t)$ enters its white vertices through the edge determining the maximum distance from the white vertex to a leaf. We illustrate the introduced definitions in Figure 4. We present there an example run of an online scenario together with changing levels of vertices. We mark the dispatching vertices in each turn. An important property of the level function is that the levels of vertices drop by at most one along both $\text{path}_t(b_t)$ and $\text{sec-path}_t(b_t)$.

Lemma 15. For $v \in W \cup B_t$ and $u \in \{\text{dir}_t(v), \text{sec-dir}_t(v)\}$ it holds that $|\text{level}_t(v) - \text{level}_t(u)| \leq 1$.

Proof. See Appendix C. \square

We are ready to move on to the proof of Theorem 1. We consider two cases:

- (1) the level of a dispatching vertex in turn t grows by at most a factor of β
- (2) the level of a dispatching vertex in turn t grows by more than a factor of β

where β is some constant value greater than 1 which we reveal later on. The total length of paths $\text{path}_t^s(b_t)$ satisfying case (1) is bounded by Lemma 16 while the total length of paths $\text{path}_t^s(b_t)$ satisfying case (2) is bounded by Lemma 17.

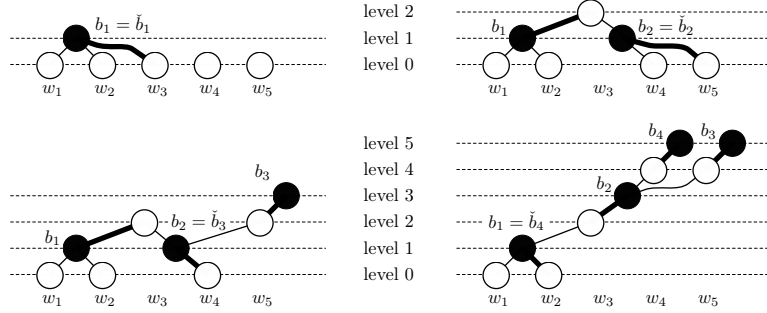


FIGURE 4. Levels

Lemma 16. *For cases when $\text{dist}_t(b_t) < \infty$, \check{b}_t is defined and $\text{level}_t(\check{b}_t) < \beta \text{level}_{t-1}(\check{b}_t)$ the total length of paths $\text{path}_t^s(b_t)$ is bounded by $2\beta n + \beta n \log_2 n$.*

Proof. Let t be such that it satisfies the assumptions of the lemma. First observe that $b_t \neq \check{b}_t$, otherwise $\text{level}_{t-1}(\check{b}_t) = 0 < \text{level}_t(\check{b}_t)/\beta$. Due to Lemma 8 and Lemma 9, b_t has precisely one neighbour alive in turn $t - 1$.

In order to show an appropriate charging scheme, consider the final forest $F = F_n$. We study the connected components of a subforest $F[A_t]$ of F induced on vertices alive in turn t . Recall that vertices not yet presented are considered alive. In turn t some vertices, in particular b_t , die. Due to Lemma 10 vertices that die in turn t form a connected component D of $F[A_{t-1}]$. Then the connected component C of \check{b}_t in $F[A_{t-1}]$ splits into D and components C_1, \dots, C_k in $F[A_t]$. Let C_1 be the largest component among C_1, \dots, C_k . Due to Lemma 10, $\text{path}_t^s(b_t)$ is contained entirely in one of the components C_1, \dots, C_k , say $\text{path}_t^s(b_t)$ is contained in C_i . If $i \neq 1$, we can charge the length of $\text{path}_t^s(b_t)$ by charging 1 token to the vertices of C_i . A particular vertex can be charged at most $\log_2 n$ tokens this way, as each time it is charged its component halves the size. It remains to deal with the case when $\text{path}_t^s(b_t)$ is contained in the largest component C_1 . For the reference see Figure 5. Let w^p be the predecessor of \check{b}_t on $\text{path}_t(b_t)$. Let T be the

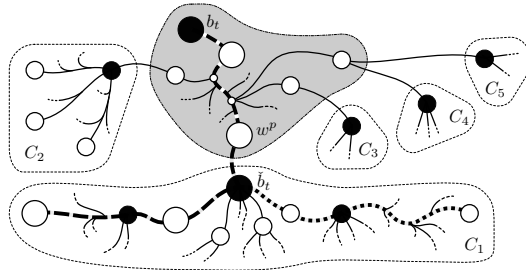


FIGURE 5. Splitting the component C into C_1, \dots, C_5 and D .

connected component of \check{b}_t in F_{t-1} rooted at \check{b}_t and let T_\curvearrowright be the connected component of \check{b}_t in F_{t-1} rooted at w^p . For the reference see Figure 6. By Observation 14 and our assumptions it

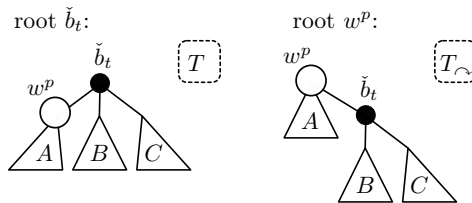


FIGURE 6. The connected component of F_{t-1} rooted in \check{b}_t and w^p .

holds that

$$\begin{aligned} \|\text{path}_t^s(b_t)\| &= \text{dist}_t(\check{b}_t) < \beta \text{dist}_{t-1}(\check{b}_t) = \\ &= \beta \min_{w \in \text{Ch}_T(\check{b}_t)} \text{mini-max}_T(w) + \beta \leq \beta \text{mini-max}_T(w^p) + \beta. \end{aligned}$$

The constant cost of β gives a total cost of βn over all turns. What remains to show is how to charge the cost of $\beta \text{mini-max}_T(w^p)$. Since $w^p \in A_{t-1}$ it holds that

$$\begin{aligned} \text{mini-max}_T(w^p) &= \max_{b \in \text{Ch}_{T_\curvearrowright}(w^p) \setminus \{\check{b}_t\}} \text{mini-max}_{T_\curvearrowright}(b) + 1 \leq \\ &\leq \max_{b \in \text{Ch}_{T_\curvearrowright}(w^p)} \text{mini-max}_{T_\curvearrowright}(b) + 1 = \text{dist}_{t-1}(w^p) < \infty. \end{aligned}$$

Thus $\text{mini-max}_T(w^p) = \|\text{mini-max-path}_T(w^p)\|$. We charge the vertices of $\text{mini-max-path}_T(w^p)$ to pay for the cost given by its length. Due to Lemma 7 it holds that $\text{mini-max-path}_T(w^p)$ visits only vertices that are alive in turn $t-1$. By definition $V(\text{mini-max-path}_T(w^p)) \cap C_1 = \emptyset$, so each vertex of $\text{mini-max-path}_T(w^p)$ either dies in turn t or is contained in C_i for $i > 1$. To pay for that, we charge β tokens to every vertex that dies in turn t and we charge β tokens to each vertex of components $C_2 \dots C_k$. The total charge for this case sums up to $\beta n + \beta n \log_2 n$. If we add the charge we needed for other cases, we obtain a total of $2\beta n + \beta n \log_2 n$. \square

Lemma 17. *For cases when $\text{dist}_t(b_t) < \infty$, \check{b}_t is defined and $\text{level}_t(\check{b}_t) \geq \beta \text{level}_{t-1}(\check{b}_t)$ the sum of the lengths of paths $\text{path}_t^s(b_t)$ is bounded by $\frac{\beta(\beta+1)}{(\beta-1)^2} n(2 \ln n + 3.4) + n$.*

Proof. Given the level function, we want to consider the vertices of $F = F_n$ whose level in turn t is above a certain value l . To be more precise, we need to consider the subforest of F induced by such vertices. This forest changes dynamically as the turns pass by. We describe it more formally below.

Definition 12. For $t \in [n]$ and $l \in \mathbb{N}$ we define $F_t^l = F[\{v \in W \cup B : l \leq \text{level}_t(v)\}]$.

Recall that if $b \in B \setminus B_t$ then $\text{level}_t(b) = 0$. For a subforest $F' = \langle V', E' \rangle$ of $F = \langle W \cup B, E \rangle$, we denote a connected component of vertex $v \in V'$ as $\text{comp}(v, F')$. The family of all connected components is denoted as $\mathcal{C}(F') = \{\text{comp}(v, F') : v \in V'\}$. For a fixed l we observe how F_t^l changes from turn $t-1$ to t . Since the level function is monotonic, i.e., it satisfies $\text{level}_{t-1}(v) \leq \text{level}_t(v)$ (see Observation 2), the following hold:

Observation 18. F_{t-1}^l is a subforest of F_t^l . Also,

$$V(F_t^l) = V(F_{t-1}^l) \cup \{v \in V(F) : \text{level}_{t-1}(v) < l \leq \text{level}_t(v)\}.$$

We fix a turn t for which $\text{dist}_t(b_t) < \infty$, \check{b}_t is defined, and $\text{level}_t(\check{b}_t) \geq \beta \text{level}_{t-1}(\check{b}_t)$. The idea is the following. We let $\underline{l} = \text{level}_{t-1}(\check{b}_t)$ and $\bar{l} = \text{level}_t(\check{b}_t)$. For the purpose of the proof we need a function that describes some intermediate level between $\text{level}_{t-1}()$ and $\text{level}_t()$. We thus extend the level function to rational indices:

$$\text{level}_{t-1/2}(v) = \begin{cases} \text{level}_{t-1}(v) & \text{if } v = \check{b}_t \\ \text{level}_t(v) & \text{otherwise.} \end{cases}$$

Observe that $\text{level}_t(v)$ function is still monotonic in t after the extension. We illustrate these definitions by example in Figure 7. We observe that on levels l from \underline{l} to $\bar{l} - 1$ the separate components of $F_{t-1/2}^l$ are merged in F_t^l . It is this merging that allows us to provide the charging scheme. The level defined for fractional indices may be interpreted as an additional fractional turn between $t-1$ and t .

Let us fix a level l . In every turn $t = 0, \frac{1}{2}, 1, 1\frac{1}{2}, \dots, n$ a number of δ tokens is assigned to every connected component C in F_t^l such that $|C| \geq \rho l$, where δ and ρ are constants that we compute later. Smaller components are not assigned any tokens. Note that if l is large, only large components are assigned tokens. We plan to use these tokens to pay for the mini-max paths in each turn. First, however, we describe how we maintain such an assignment on level l .

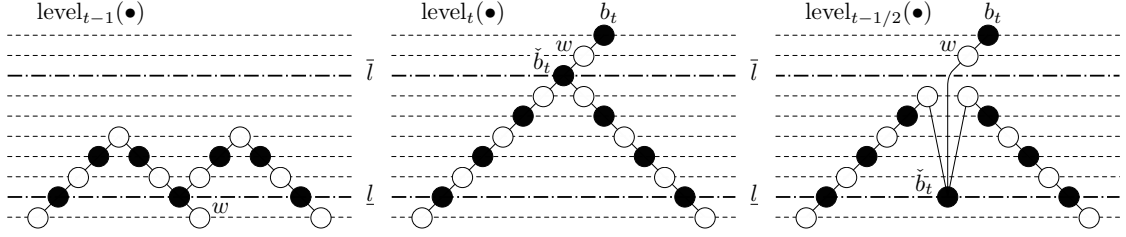


FIGURE 7. Fractional levels

First we consider moving from turn $t-1$ to turn $t-1/2$. The forest $F_{t-1/2}^l$ is obtained by adding the set of vertices $\Delta = V(F_{t-1/2}^l) \setminus V(F_{t-1}^l) = \{v \in B_t \cup W : \text{level}_{t-1}(v) < l \leq \text{level}_{t-1/2}(v)\}$ to F_{t-1}^l (see Observation 18). We want to add some structure to this process. We divide transformation from F_{t-1}^l to $F_{t-1/2}^l$ into two sub-phases. In the first sub-phase, the vertices of Δ form new singleton components: $\mathcal{C}' = \bigcup_{v \in \Delta} \{\langle \{v\}, \emptyset \rangle\}$, where $\langle \{v\}, \emptyset \rangle$ is a graph with only one vertex v and without edges. Together with the set of connected components of F_{t-1}^l (referred to as $\mathcal{C}(F_{t-1}^l)$) they form a family $\mathcal{I} = \mathcal{C}(F_{t-1}^l) \cup \mathcal{C}'$. In the second sub-phase, components in \mathcal{I} merge whenever there is an edge of F connecting them, finally becoming the connected components of $F_{t-1/2}^l$. Every component $C \in \mathcal{C}(F_{t-1/2}^l)$ can be assigned a set of components $\mathcal{I}_C = \{C_1 \dots C_k\} \subseteq \mathcal{I}$ that merged into C . There are two possible options:

- (a) there is the component $C_i \in \mathcal{I}$ with size $|C_i| \geq \rho l$, so C_i is already assigned δ tokens
- (b) every $C_i \in \mathcal{I}$ satisfies $|C_i| < \rho l$, so none of them is assigned any tokens.

In case (a), δ assigned to C_i , which ceases to exist, is now transferred to C . In case (b), if $|C| \geq \rho l$, every vertex $v \in C$ chips in with a payment of $\frac{\delta}{\rho l}$, so the vertices of C pay in total at least δ . We count the total amount that is paid at the end of the proof.

We now consider the transition from turn $t-1/2$ to turn t . There is only one vertex, mainly \check{b}_t , which changes its level. Its level increases from \check{l} to \bar{l} . Level l is only affected by this transition if $\check{l} < l \leq \bar{l}$. So, the forest F_t^l is formed from $F_{t-1/2}^l$ by adding \check{b}_t . The only difference between $\mathcal{C}(F_{t-1/2}^l)$ and $\mathcal{C}(F_t^l)$ is that some family of separate components of $\mathcal{C}(F_{t-1/2}^l)$ becomes connected by \check{b}_t and forms a new connected component $\text{comp}(\check{b}_t, F_{t-1/2}^l)$. The set of components that merge into $\text{comp}(\check{b}_t, F_{t-1/2}^l)$ is precisely $\mathcal{I}' = \mathcal{C}(\text{comp}(\check{b}_t, F_{t-1/2}^l) \setminus \check{b}_t) \cup \{\langle \check{b}_t, \emptyset \rangle\}$. The way of assigning δ to $\text{comp}(\check{b}_t, F_{t-1/2}^l)$ if $|\text{comp}(\check{b}_t, F_{t-1/2}^l)| \geq \rho l$ is the same as in the transition from $t-1$ to $t-1/2$. The difference is that now we want to utilize some of the assigned tokens to pay for the mini-max path in turn t . Thus, we distinguish three cases now:

- (i) exactly one of the components $C' \in \mathcal{I}'$ satisfies $|C'| \geq \rho l$, so C' is assigned δ tokens,
- (ii) every $C \in \mathcal{I}'$ satisfies $|C| < \rho l$ so none of them is assigned tokens,
- (iii) two or more components $C', C'' \in \mathcal{I}'$ satisfy $|C'| \geq \rho l$ and $|C''| \geq \rho l$, so C' and C'' are both already assigned δ tokens.

Cases (i) and (ii) are handled in the exactly same manner as in the transition from $t-1$ to $t-1/2$. The difference is that in case (iii) we utilize δ tokens assigned to C' while δ tokens assigned to C'' transfer to $\text{comp}(\check{b}_t, F_{t-1/2}^l)$.

It remains to prove that the tokens utilized in turn t suffice to pay for $\|\text{path}_t^s(b_t)\|$. Observation 14 shows that $\text{level}_t(\check{b}_t) = \|\text{path}_t^s(b_t)\|$. So we need to pay $\text{level}_t(\check{b}_t)$ tokens when moving from turn $t-1/2$ to t . Let $\rho := (\beta - 1)/(\beta + 1)$. By Claim 19, proved later on, case (iii) occurs on at least $(\bar{l} - \check{l})/2$ levels. Since $\check{l} \leq \bar{l}/\beta$, we utilize at least $\delta(\bar{l} - \bar{l}/\beta)/2 = \frac{\delta(1-1/\beta)}{2} \text{level}_t(\check{b}_t)$ tokens. Setting $\delta := 2/(1 - 1/\beta)$ allows paying the desired amount.

Now we count the sum of lengths of $\text{path}_t^s(b_t)$. Every vertex pays $\frac{\delta}{\rho l}$ at most once per level and the highest level is not greater than $2n$, so the total amount paid by a vertex over all the turns is bounded by $\frac{\delta}{\rho} \sum_{l=1}^{2n} \frac{1}{l} \leq \frac{\delta}{\rho} (\ln(2n) + 1)$. Hence, the total amount paid by all vertices is

at most $\frac{\delta}{\rho}n(\ln n + 1.7)$. If we plug in the constants $\rho = (\beta - 1)/(\beta + 1)$ and $\delta = 2/(1 - 1/\beta)$ into above bound, we obtain that $\sum_{t \in \mathcal{T}} \|\text{path}_t^s(b_t)\| \leq \frac{\beta(\beta+1)}{(\beta-1)^2}n(2 \ln n + 3.4)$. \square

To complete the proof of Lemma 17 we move on to proving the following Claim.

Claim 19. *For a fixed $t \in [n]$ let $l_0 = \text{level}_{t-1}(\check{b}_t) + 1$ and $l_1 = \lfloor (\text{level}_{t-1}(\check{b}_t) + \text{level}_t(\check{b}_t))/2 \rfloor$. For $\rho = \frac{\beta-1}{\beta+1}$ and $l \in \{l_0, \dots, l_1\}$ there exist two different vertices $w_1, w_2 \in N_t(\check{b}_t) \cap A_t$ such that $\text{comp}(w_1, F_{t-1/2}^l)$ and $\text{comp}(w_2, F_{t-1/2}^l)$ are two separate components of $F_{t-1/2}^l$ and $|\text{comp}(w_i, F_{t-1/2}^l)| \geq \rho l$ for $i \in \{1, 2\}$.*

Proof. Fix $l \in \{l_0, \dots, l_1\}$. By definition $N_t(\check{b}_t) \cap A_t \geq 2$. Thus, $\text{dist}_t(\check{b}_t) < \infty$ and $\text{sec-dist}_t(\check{b}_t) < \infty$. We show that $w_1 = \text{dir}_t(\check{b}_t)$ and $w_2 = \text{sec-dir}_t(\check{b}_t)$ satisfy the desired conditions.

First note that $w_1, w_2 \in V(F_{t-1/2}^l)$, because $\text{level}_t(w_i) \geq \text{level}_t(\check{b}_t) - 1 \geq l_1$ for $i \in \{1, 2\}$ due to Lemma 15. Note also that $\text{comp}(w_1, F_{t-1/2}^l)$ and $\text{comp}(w_2, F_{t-1/2}^l)$ are two separate components of $F_{t-1/2}^l$ because the only path connecting w_1 and w_2 in F is through \check{b}_t and $\check{b}_t \notin V(F_t^l)$ because $\text{level}_{t-1/2}(\check{b}_t) = \text{level}_{t-1}(\check{b}_t) < l_0 \leq l$.

Due to Lemma 15 the levels of vertices drop by at most one along $\text{path}_t(\check{b}_t)$ and $\text{sec-path}_t(\check{b}_t)$. Let π_1 be the prefix of $\text{path}_t(\check{b}_t)$ of length $\bar{l} - l$ and π_2 be the prefix of $\text{sec-path}_t(\check{b}_t)$ of length $\bar{l} - l$, where $\bar{l} = \text{level}_t(\check{b}_t)$. It holds that if $v \in V(\pi_i)$ then $\text{level}_{t-1/2}(v) = \text{level}_t(v) \geq l$. Because $l_0 - 1 = \text{level}_{t-1}(\check{b}_t) \leq \text{level}_t(\check{b}_t)/\beta = \bar{l}/\beta$ and $l \leq l_1 = \lfloor (l_0 - 1 + \bar{l})/2 \rfloor$ we have $l \leq (\bar{l}/\beta + \bar{l})/2 = (1/\beta + 1) \cdot \bar{l}/2$. This implies $|\text{comp}(w, F_t^l)| \geq \bar{l} - l = \frac{2\beta}{\beta+1} \cdot \frac{1+1/\beta}{2} \cdot \bar{l} - l \geq \frac{2\beta}{\beta+1} \cdot l - l = \frac{\beta-1}{\beta+1} \cdot l$. Setting $\rho = (\beta - 1)/(\beta + 1)$ completes the proof of the claim. \square

We can now put all the pieces together to prove our main result.

Theorem 20. $\sum_{t \in [n]: \text{dist}_t(b_t) < \infty} \text{dist}_t(b_t) \in \mathcal{O}(n \log n)$.

Proof. By definition 10 we have $\text{path}_t(b_t) = \text{path}_t^p(b_t) \cdot \text{path}_t^s(b_t)$. By Observation 13 it holds that $\sum_{t \in [n]: \text{dist}_t(b_t) < \infty} \|\text{path}_t^p(b_t)\| \leq 2n$. If \check{b}_t is undefined, then $\text{path}_t^s(b_t)$ is empty so its length is 0. For the cases when \check{b}_t is defined and $\text{level}_t(\check{b}_t) < \beta \text{level}_{t-1}(\check{b}_t)$ we have $\sum_{t \in [n]: \text{dist}_t(b_t) < \infty} \|\text{path}_t^s(b_t)\| \leq 2\beta n + \beta n \log n$. For the cases when \check{b}_t is defined and $\text{level}_t(\check{b}_t) \geq \beta \text{level}_{t-1}(\check{b}_t)$ we have $\sum_{t \in [n]: \text{dist}_t(b_t) < \infty} \|\text{path}_t^s(b_t)\| \leq \frac{\beta(\beta+1)}{(\beta-1)^2}n(2 \ln n + 3.4) + n$. This gives the theorem statement for any $\beta > 1$. \square

REFERENCES

- [BHR17] Aaron Bernstein, Jacob Holm, and Eva Rotenberg. Online bipartite matching with amortized $O(\log^2 n)$ replacements. *CoRR*, abs/1707.06063, 2017.
- [BLSZ14] Bartłomiej Bosek, Dariusz Leniowski, Piotr Sankowski, and Anna Zych. Online bipartite matching in offline time. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 384–393, 2014.
- [BLSZ15] Bartłomiej Bosek, Dariusz Leniowski, Piotr Sankowski, and Anna Zych. Shortest augmenting paths for online matchings on trees. In *Approximation and Online Algorithms - 13th International Workshop, WAOA 2015, Patras, Greece, September 17-18, 2015. Revised Selected Papers*, pages 59–71, 2015.
- [BM08] Benjamin E. Birnbaum and Claire Mathieu. On-line bipartite matching made simple. *SIGACT News*, 39(1):80–87, 2008.
- [BS15] Aaron Bernstein and Cliff Stein. Fully dynamic matching in bipartite graphs. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part I*, pages 167–179, 2015.
- [CDKL09] Kamalika Chaudhuri, Constantinos Daskalakis, Robert D. Kleinberg, and Henry Lin. Online bipartite perfect matching with augmentations. In *INFOCOM 2009. 28th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 19-25 April 2009, Rio de Janeiro, Brazil*, pages 1044–1052, 2009.
- [DJK13] Nikhil R. Devanur, Kamal Jain, and Robert D. Kleinberg. Randomized primal-dual analysis of RANKING for online bipartite matching. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 101–107, 2013.

- [EK72] Jack Edmonds and Richard M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, 1972.
- [GKKV95] Edward F. Grove, Ming-Yang Kao, P. Krishnan, and Jeffrey Scott Vitter. Online perfect matching and mobile computing. In *Algorithms and Data Structures, 4th International Workshop, WADS '95, Kingston, Ontario, Canada, August 16-18, 1995, Proceedings*, pages 194–205, 1995.
- [GKS14] Anupam Gupta, Amit Kumar, and Cliff Stein. Maintaining assignments online: Matching, scheduling, and flows. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 468–479, 2014.
- [GP13] Manoj Gupta and Richard Peng. Fully dynamic $(1 + \epsilon)$ -approximate matchings. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 548–557, 2013.
- [KVV90] Richard M. Karp, Umesh V. Vazirani, and Vijay V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, May 13-17, 1990, Baltimore, Maryland, USA*, pages 352–358, 1990.
- [NS13] Ofer Neiman and Shay Solomon. Simple deterministic algorithms for fully dynamic maximal matching. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 745–754, 2013.
- [San07] Piotr Sankowski. Faster dynamic matchings and vertex connectivity. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 118–126, 2007.

APPENDIX A. THE MINI-MAX GAME

Observation 2. Fix a vertex $u \in B \cup W$. The functions $\text{dist}_t(u)$ and $\text{sec-dist}_t(u)$ are non-decreasing with respect to t for the whole range of t where $u \in V(F_t)$.

Proof. We only prove monotonicity of $\text{dist}_t(v)$ here, the proof of monotonicity of $\text{sec-dist}_t(v)$ is analogous. Fix $t \in [n]$. We claim that $\text{dist}_{t-1}(v) \leq \text{dist}_t(v)$ provided that $v \in V(F_{t-1})$. Let T be a component of v in F_{t-1} rooted at v and S be a component of v in F_t rooted at v . Note that S is a subtree of T . By definition it holds that $\text{dist}_{t-1}(v) = \text{mini-max}_T(v)$ and $\text{dist}_t(v) = \text{mini-max}_S(v)$. We prove our claim by a bottom-up induction on T . The inductive hypothesis we prove is that $\text{mini-max}_T(u) \leq \text{mini-max}_S(u)$ for $u \in V(T)$.

First assume that u is black. The neighborhood of u does not change from $t-1$ to t , since after u is presented, it never changes its neighborhood. Therefore $\text{Ch}_T(u) = \text{Ch}_S(u)$. If u is a leaf of T , then $\text{mini-max}_T(u) = \infty = \text{mini-max}_S(u)$. If u has children then by induction hypothesis on children it follows that

$$\text{mini-max}_T(u) = \min_{w \in \text{Ch}_T(u)} \text{mini-max}_T(w) + 1 \leq^{\text{ind}} \min_{w \in \text{Ch}_S(u)} \text{mini-max}_S(w) + 1 = \text{mini-max}_S(u).$$

Now assume that u is white. The neighborhood of u can change from $t-1$ to t , but it can only increase, meaning that $\text{Ch}_T(u) \subseteq \text{Ch}_S(u)$. If u is a leaf in T , then by the fact that $\text{mini-max}_S(u)$ is non-negative it holds that $\text{mini-max}_T(u) = 0 \leq \text{mini-max}_S(u)$. If u has children in T , then by induction hypothesis we infer that

$$\begin{aligned} \text{mini-max}_T(u) &= \max_{b \in \text{Ch}_T(u)} \text{mini-max}_T(b) + 1 \leq^{\text{ind}} \max_{b \in \text{Ch}_T(u)} \text{mini-max}_S(b) + 1 \leq \\ &\leq \max_{b \in \text{Ch}_S(u)} \text{mini-max}_S(b) + 1 = \text{mini-max}_S(u). \end{aligned}$$

□

Lemma 3. Let $1 \leq t_0 \leq t \leq n$ and let ρ_t be the shortest augmenting path from b_{t_0} to a free white vertex according to any given matching M in F_t where b_{t_0} is free. It holds that if $\text{dist}_t(b_{t_0}) < \infty$ then ρ_t exists and $\|\rho_t\| \leq \text{dist}_t(b_{t_0})$.

Proof. Fix $t_0 \in [n]$ and $t \in \{t_0, \dots, n\}$. Let T be the connected component of b_{t_0} in F_t rooted at b_{t_0} . We prove our lemma by a bottom-up induction on T .

Before we state inductive hypothesis, we need to introduce a few more definitions. We denote as $\rho_T(v)$ the shortest alternating path from v to a free white vertex in the subtree of T rooted at v . Here, an augmenting path is taken with respect to M and a white vertex is considered free if it is free in the subtree. If there is no such alternating path we write $\rho_T(v) = \perp$ for which length $\|\perp\| = \infty$. It is not hard to see that such a path is either a single free white vertex, or it starts with an unmatched edge if v is black and with a matched edge if v is white. When we say that a vertex is unmatched from below, we mean that all its incident edges leading to its children in T are free from matching. Similarly, we say that a vertex is unmatched from above, if the edge leading to its parent in T is free from matching.

Our inductive hypothesis states the following. If v is

- (a) black and unmatched from below or
- (b) white and unmatched from above

then $\|\rho_T(v)\| \leq \text{mini-max}_T(v)$.

Let us first assume that v is black. If v is a leaf in T , then $\rho_T(v) = \perp$, so $\|\rho_T(v)\| = \infty = \text{mini-max}_T(v)$, so the hypothesis is fulfilled. Let us consider the case when v is not a leaf. We consider two sub-cases of this case. First assume that $\rho_T(v) = \perp$. This implies that $\rho_T(w) = \perp$ for all $w \in \text{Ch}_T(v)$. By induction it holds that

$$\text{mini-max}_T(v) = \min_{w \in \text{Ch}_T(v)} \text{mini-max}_T(w) + 1 \stackrel{\text{ind}}{=} \infty + 1 = \infty.$$

Now assume that $\rho_T(v) \neq \perp$. Let w be a successor of v on $\rho_T(v)$ and let $w' = \text{mini-max-next}_T(v)$. Then by the power of induction we infer that

$$\text{mini-max}_T(v) = \text{mini-max}_T(w') + 1 \geq^{\text{ind}} \|\rho_T(w')\| + 1 \geq \|\rho_T(w)\| + 1 = \|\rho_T(v)\|.$$

Let us now assume that v is white. If v is a leaf, then $\|\rho_T(v)\| = 0 = \text{mini-max}_T(v)$ so the hypothesis is fulfilled. If v is not a leaf, we again consider two cases. First assume that $\rho_T(v) = \perp$. In such case there has to be a vertex $b' \in \text{Ch}_T(v)$ matched to v , otherwise v is free and the path $\rho_T(v)$ exists. In such case $\rho_T(b') = \perp$ otherwise there would exist augmenting path also from v . Then by inductive assumption $\text{mini-max}_T(b') = \infty$. From that it follows that

$$\text{mini-max}_T(v) = \max_{b \in \text{Ch}_T(v)} \text{mini-max}_T(b) + 1 \geq \text{mini-max}_T(b') + 1 \stackrel{\text{ind}}{=} \infty + 1 = \infty.$$

Now assume that $\rho_T(v) \neq \perp$. In this case either v is free in its subtree and $\|\rho_T(v)\| = 0 = \text{mini-max}_T(v)$, or there is a vertex $b' \in \text{Ch}_T(v)$ matched to v . But then by the power of induction we infer that

$$\text{mini-max}_T(v) = \max_{b \in \text{Ch}_T(v)} \text{mini-max}_T(b) + 1 \geq \text{mini-max}_T(b') + 1 \geq \stackrel{\text{ind}}{\|\rho_T(b')\|} + 1 = \|\rho_T(v)\|.$$

This proves the hypothesis, in particular this proves that $\|\rho_T(b_{t_0})\| \leq \text{mini-max}_T(b_{t_0}) = \text{dist}_t(b_{t_0})$. Note that if $\text{dist}_t(b_{t_0}) < \infty$, then $\|\rho_T(b_{t_0})\| < \infty$, and that by definition implies that $\rho_t = \rho_T(b_{t_0})$ exists and $\|\rho_t\| = \|\rho_T(b_{t_0})\| \leq \text{dist}_t(b_{t_0})$. \square

APPENDIX B. DEAD VERTICES

Lemma 4. *Let $1 \leq t_0 \leq t \leq n$. Then $\text{dist}_t(b_{t_0}) = \infty$ iff b_{t_0} breaks Hall's condition in time t .*

Proof.

" \Leftarrow ": Assume that $\text{dist}_t(b_{t_0}) < \infty$. Then, by Lemma 3, an augmenting path exists from b_{t_0} for a maximum matching M where b_{t_0} is free. By Hall's theorem this implies that b_{t_0} does not break Hall's condition.

" \Rightarrow ": Let T be the connected component of b_{t_0} in F_t rooted at b_{t_0} . Assume $\text{dist}_t(b_{t_0}) = \infty$. We are going to define a subtree S of T such that $V(S) \cap B_t$ is precisely the set X that certifies that b_{t_0} breaks Hall's condition in time t .

For that we first define a forest F that is a subgraph of T and pick one component of F to be S . Let $F = (W_F \cup B_F, E_F)$, where $W_F = W \cap V(T)$ are the white vertices of T , $B_F = B_t \cap V(T)$ are the black vertices of T and $E_F = E_1 \cup E_2$, where

$$E_1 = \{(\text{parent}_T(w), w) : w \in W_F\},$$

$$E_2 = \{(w, \text{mini-max-next}_T(w)) : w \in W_F \text{ and } \text{mini-max-next}_T(w) \text{ exists}\}.$$

Next we let S be the connected component of b_{t_0} in F rooted at b_{t_0} . The construction of S is shown in Figure 8. Before we proceed, we prove a helpful observation that $\text{mini-max}_S(u) = \infty$

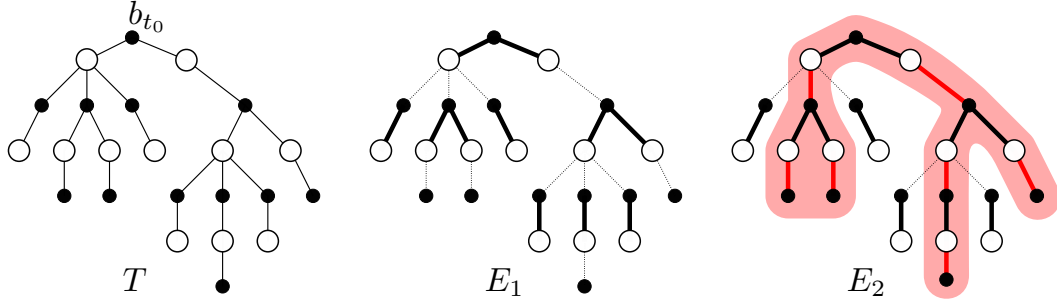


FIGURE 8. The construction of rooted tree S in the proof of Lemma 4.

holds for all $u \in V(S)$. To that end we assume for the sake of contradiction that there is $u \in V(S)$ such that $\text{mini-max}_S(u) < \infty$ and we pick the closest to the root such vertex. Then $\text{mini-max}_S(\text{parent}_S(u)) = \infty$. We consider two cases. If u is white then $v = \text{parent}_S(u)$ is black and thus

$$\infty = \text{mini-max}_S(v) = \min_{w \in \text{Ch}_S(v)} \text{mini-max}_S(w) + 1 \leq \text{mini-max}_S(u) + 1 < \infty.$$

This gives a contradiction. If, on the other hand, u is black then $v = \text{parent}_S(u)$ is white and hence, due to the way S is constructed, v has only one child u in S . Thus

$$\infty = \text{mini-max}_S(v) = \max_{b \in \text{Ch}_S(v)} \text{mini-max}_S(b) + 1 = \text{mini-max}_S(u) + 1 < \infty.$$

This also gives a contradiction.

Let us denote white vertices of S as $W_S = V(S) \cap W$ and black vertices of S as $B_S = V(S) \cap B_t$. Now let $X = B_S$. We prove that conditions (1)-(3) of Definition 3 hold for X . Condition (1) trivially holds. Let us consider condition (2). Since $\text{mini-max}_S(w) = \infty$ for all $w \in W_S$, every $w \in W_S$ has a child in S and by the way S is constructed this child is unique. On the other hand each black vertex $b \in B_S \setminus \{b_{t_0}\}$ has a unique parent. This gives a one-to-one map $f : B_S \setminus \{b_{t_0}\} \rightarrow W_S$ (such that $f(b) := \text{parent}_S(b)$) and implies $|W_S| = |B_S \setminus \{b_{t_0}\}|$.

We move on to proving condition (3). Let $X' \subsetneq X$. If $b_{t_0} \notin X'$ then $|X'| = |f(X')| \leq |N_t(X')|$. So, we can assume that $b_{t_0} \in X'$. Then there exists $b' \in X \setminus X'$ such that $\text{parent}_S(\text{parent}_S(b')) \in X'$. Thus

$$|X'| = |X' \setminus \{b_{t_0}\}| + 1 = |f(X' \setminus \{b_{t_0}\})| + 1 = |f(X' \setminus \{b_{t_0}\}) \cup \{\text{parent}_S(b')\}| \leq |N_t(X')|.$$

This proves the minimality of X and ends the proof of the lemma. \square

Lemma 7. *Let $t \in [n]$ and $v \in A_t$. Pick any vertex as a root of the connected component of v in F_t and let T be the corresponding rooted tree. Then $V(\text{mini-max-path}_T(v)) \subseteq A_t$.*

Proof. Observe that

$$\text{mini-max}_T(b') \leq \text{sec-dist}_t(b') \quad \text{for } b' \in B_t. \quad (1)$$

Let $\text{mini-max-path}_T(v) = (v^{(1)}, v^{(2)}, \dots, v^{(k)})$. We prove that $v^{(i)}$ is alive by induction on $i = 1, 2, \dots, k$. Vertex $v^{(1)} = v$ is alive by the assumptions of the lemma. We now assume that $v^{(i)}$ is alive and aim to prove that $v^{(i+1)}$ is alive too. If $v^{(i)} \in W$ then by Observation 6 we know that all neighbours of $v^{(i)}$ are alive. In particular $v^{(i+1)} \in A_t$. We are left with the case when $b := v^{(i)} \in B_t$ and $w := v^{(i+1)} = \text{mini-max-next}_T(b) \in W$. We aim to prove that w is alive, i.e. $\text{dist}_t(w) < \infty$. Let S be the connected component of v in F_t rooted in w . For the reference see Figure 9.

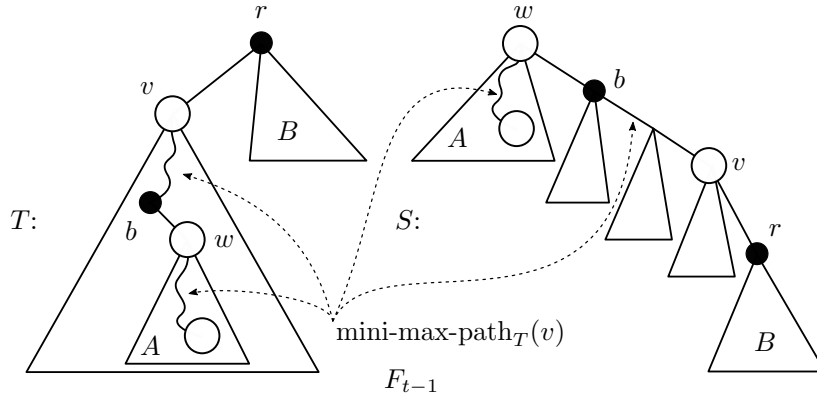


FIGURE 9. Aliveness of $\text{mini-max-path}_t(v)$

Then

$$\text{dist}_t(w) = \text{mini-max}_S(w) = \max_{b' \in \text{Ch}_S(w)} \text{mini-max}_S(b').$$

So it suffices to prove that $\text{mini-max}_S(b') < \infty$ for any child b' of w in S . If $b' = b$ then

$$\text{mini-max}_S(b) \leq^{\text{by(1)}} \text{sec-dist}_t(b) < \infty$$

where the last inequality holds by aliveness of b . If $b' \neq b$ then the subtree of b' is identical for T and S and as a consequence

$$\begin{aligned} \text{mini-max}_S(b') &= \text{mini-max}_T(b') \leq \max_{b'' \in \text{Ch}_T(w)} \text{mini-max}_T(b'') = \\ &= \text{mini-max}_T(w) - 1 = \text{mini-max}_T(b) - 2 \leq^{\text{by(1)}} \text{sec-dist}_t(b) - 2 < \infty. \end{aligned}$$

\square

Lemma 8. *If b_t does not break Hall's condition in turn t , then b_t has at least one neighbour in F_t which was alive in turn $t - 1$.*

Proof. Let T be the component of b_t in F_t rooted at b_t . By Lemma 4 we have that

$$\infty > \text{dist}_t(b_t) = \min_{w \in \text{Ch}_T(b_t)} \text{mini-max}_T(w) + 1 = \min_{w \in \text{Ch}_T(b_t)} \text{dist}_{t-1}(w) + 1.$$

Hence, the new vertex b_t has a white neighbour w' such that $\text{dist}_{t-1}(w') < \infty$. By definition w' is alive in turn $t - 1$. \square

Lemma 9. *If b_t has at least two neighbours which are alive in turn $t - 1$ then b_t is alive in turn t and no vertex dies in turn t .*

Proof. We first prove that no vertex dies in turn t . Let $u \in B_{t-1} \cup W$ for some $t \in [n]$ and assume that b_t is as stated in the lemma. Let T be the connected component of u in F_{t-1} rooted at u and let S be the component of u in F_t rooted at u . If u is not connected to b_t in F_t , then by definition $\text{sec-dist}_{t-1}(u) = \text{sec-dist}_t(u)$ since $T = S$. Hence, u does not die in turn t .

Otherwise $N_t(b_t)$ contains a vertex of T , let it be w . For this case, to prove that u does not die in turn t , we show the following:

$$\begin{aligned} \text{dist}_{t-1}(u) < \infty &\Rightarrow \text{dist}_t(u) < \infty \text{ and} \\ \text{sec-dist}_{t-1}(u) < \infty &\Rightarrow \text{sec-dist}_t(u) < \infty. \end{aligned}$$

Let $F_t - \{w, b_t\}$ be the graph obtained by removing edge $\{w, b_t\}$ from F_t . Let R be the connected component of b_t in $F_t - \{w, b_t\}$, rooted at b_t . The only difference between T and S is that b_t becomes a child of w in S and R is attached to T via edge (w, b_t) . This is illustrated in Figure 10 to the left. Note that the only vertices who change their mini-max revenue are on the path π_{uw} from u to w in T . Also, in the proof of Observation 2 we proved that $\text{mini-max}_T(v) \leq \text{mini-max}_S(v)$ for $v \in V(T)$. These two observations imply that it suffices to show that $\text{mini-max}_T(w) < \infty \Rightarrow \text{mini-max}_S(w) < \infty$. Due to the assumption of the lemma, there exists $w' \in \text{Ch}_S(b_t)$ such that $\text{mini-max}_S(w') = \text{dist}_{t-1}(w') < \infty$. Thus

$$\text{mini-max}_S(b_t) = \min_{w'' \in \text{Ch}_S(b_t)} \text{mini-max}_S(w'') + 1 \leq \text{mini-max}_S(w') + 1 < \infty.$$

For the reference see Figure 10 to the right.

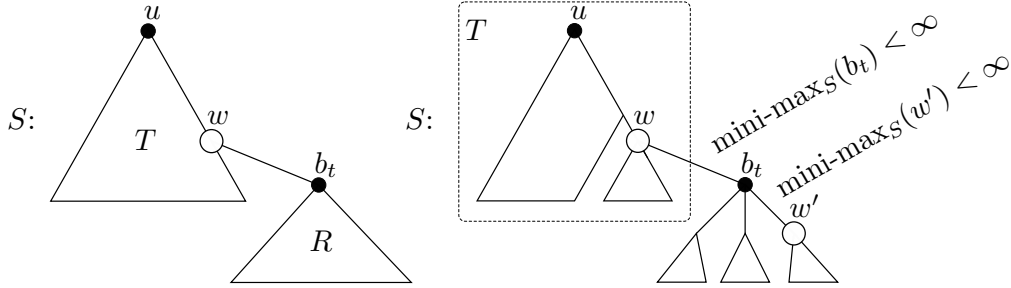


FIGURE 10. The case when b_t has at least two neighbours which are alive.

Hence

$$\text{mini-max}_S(w) = \max\{\text{mini-max}_T(w), \text{mini-max}_S(b_t) + 1\} \begin{cases} = \infty & \text{if } \text{mini-max}_T(w) = \infty, \\ < \infty & \text{otherwise.} \end{cases}$$

To complete the proof it remains to show that $b_t \in A_t$. Let T be a component of F_t rooted at b_t . Note that $\text{dist}_{t-1}(w) = \text{mini-max}_T(w)$ for $w \in \text{Ch}_T(b_t)$. Since there exists $w_1, w_2 \in \text{Ch}_T(b_t)$ such that $\text{dist}_{t-1}(w_i) = \text{mini-max}_T(w_i) < \infty$ for $i \in \{1, 2\}$, we have $\text{dist}_t(b_t) < \infty$ and $\text{sec-dist}_t(b_t) < \infty$. \square

Lemma 10. *If b_t has exactly one neighbour in F_t which was alive in turn $t - 1$ and there is a path π from b_t to $v \in B_t \cup W$ such that*

- (1) *all vertices of π were alive in turn $t - 1$ and*
- (2) *there are no life portals from \mathcal{LP}_t on π ,*

then v dies in turn t . Vertices of $B_t \cup W$ that cannot be reached from b_t via such path do not die in turn t .

Proof. The illustration for this lemma is provided in Figure 3. By monotonicity of $\text{dist}_t(v)$ and $\text{sec-dist}_t(v)$ functions with respect to t (see Observation 2), the vertices dead in turn $t - 1$ remain dead in turn t . By Definitions 2 and 4 vertex b_t dies in turn t . By Observation 6 all other vertices on π also die in turn t . What we need to prove is that the remaining vertices stay alive, i.e., do not die in turn t .

For the sake of this proof we introduce notation for a rooted subtree. For a rooted tree T and a vertex $u \in V(T)$ we let $\Delta_T(u)$ denote the subtree of T rooted at u containing u and all its descendants in T . We first observe that $\mathcal{LP}_t \subseteq A_{t-1}$ due to Observation 6. For $p \in \mathcal{LP}_t$ let $\text{WA}_{t-1}(p) = N_{t-1}(p) \cap A_{t-1}$. Due to Definition 6, $|\text{WA}_{t-1}(p)| \geq 3$. We first prove that life portals remain alive, i.e., $\mathcal{LP}_t \subseteq A_t$. Let T be the connected component of F_{t-1} rooted in p and let S be the connected component of F_t rooted in p . Observe that $\text{WA}_{t-1}(p) \subseteq \text{Ch}_T(p)$. There is only one vertex $w \in \text{Ch}_T(p)$, for which $\Delta_T(w) \neq \Delta_S(w)$. Hence, there exist two vertices $w_1, w_2 \in \text{WA}_{t-1}(p)$, for whom $\Delta_T(w_i) = \Delta_S(w_i)$ for $i \in \{1, 2\}$. It then holds that for $i \in \{1, 2\}$:

$$\infty > \text{dist}_{t-1}(w_i) \geq \text{mini-max}_T(w_i) = \text{mini-max}_S(w_i).$$

Also, $\text{Ch}_T(p) \setminus \{\text{mini-max-next}_T(p)\}$ contains either w_1 or w_2 , say it contains w_1 . Then

$$\text{sec-dist}_t(p) = \min_{w \in \text{Ch}_S(p) \setminus \{\text{mini-max-next}_T(p)\}} \text{mini-max}_S(w) + 1 \leq \text{mini-max}_S(w_1) + 1 < \infty.$$

Now let $u \in B_t \cup W \setminus \mathcal{LP}_t$ such that u was alive in turn $t-1$. It remains to prove that u does not die in turn t . If u is not connected to b_t in F_t , then u obviously stays alive in turn t . Otherwise there is a unique path from u to b_t in F_t and this path contains a life portal, let it be $p \in \mathcal{LP}_t$. Let now T be a component of u in F_{t-1} rooted in u and let S be a component of u in F_t rooted in u .

Let $F_t - \{w, b_t\}$ be the graph obtained by removing edge $\{w, b_t\}$ from F_t . Let R be the connected component of b_t in $F_t - \{w, b_t\}$, rooted at b_t . The only difference between T and S is that b_t becomes a child of some vertex $w \in V(\Delta_T(p))$ and R is attached to T via edge (w, b_t) . Also, in the proof of Observation 2 we proved that $\text{mini-max}_T(v) \leq \text{mini-max}_S(v)$ for $v \in V(T)$.

Based on these observations, it suffices to show that $\text{mini-max}_T(p) < \infty \Rightarrow \text{mini-max}_S(p) < \infty$. For that it suffices to show that $\text{mini-max}_S(p) < \infty$. Now observe that $\text{WA}_{t-1}(p) \cap \text{Ch}_S(p) \geq 2$ (one of the vertices in $\text{WA}_{t-1}(p)$ could be p 's parent). Let $w_1, w_2 \in \text{WA}_{t-1}(p) \cap \text{Ch}_S(p)$. Then there is $w_i \in \{w_1, w_2\}$ such that $\Delta_T(w_i) = \Delta_S(w_i)$. Since $w_i \in \text{WA}_{t-1}(p)$, it holds that

$$\infty > \text{dist}_{t-1}(w_i) \geq \text{mini-max}_T(w_i) = \text{mini-max}_S(w_i).$$

This implies that

$$\text{mini-max}_S(p) = \min_{w \in \text{Ch}_S(p)} \text{mini-max}_S(w) + 1 \leq \text{mini-max}_S(w_i) + 1 < \infty.$$

□

APPENDIX C. THE PROOF

Observation 14. *Let $t \in [n]$ and $\text{dist}_t(b_t) < \infty$ and \check{b}_t is defined. Then $\|\text{path}_t^s(b_t)\| = \text{dist}_t(\check{b}_t)$.*

Proof. The statement of the observation holds by definition if $\check{b}_t = b_t$. Otherwise the only possible scenario is the one covered by Lemma 10. Let w^p be the predecessor of \check{b}_t on $\text{path}_t(b_t)$. According to Lemma 10 it holds that $w^p \in D_t$. By Lemma 7 $\text{path}_t(\check{b}_t)$ cannot visit w^p . Therefore $\text{path}_t(\check{b}_t) = \text{path}_t^s(b_t)$, and this implies the desired claim. \square

This section is devoted to proving Lemma 15. The proof requires analyzing many trees rooted at the vertices of the considered mini-max paths. Such analysis becomes much simpler if we reformulate some of the definitions introduced in the paper. We start by reformulating the definition of the mini-max() function.

Definition 13. For $w \in W$, $b \in B_t$ such that $\{w, b\} \in E(W \cup B_t)$ the determined mini-max distance is defined as

$$\begin{aligned} \text{det-dist}_t(w, b) &= \begin{cases} \min_{w' \in N_t(b) \setminus \{w\}} \text{det-dist}_t(b, w') + 1 & \text{if } N_t(b) \setminus \{w\} \neq \emptyset \\ \infty & \text{otherwise,} \end{cases} \\ \text{det-dist}_t(b, w) &= \begin{cases} \max_{b' \in N_t(w) \setminus \{b\}} \text{det-dist}_t(w, b') + 1 & \text{if } N_t(w) \setminus \{b\} \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Vertex $b' \in N_t(w) \setminus \{b\}$ which determines the maximum is denoted as $\text{det-dir}_t(b, w)$. If there are more such vertices we choose the first one in some predefined order on the vertices $B \cup W$. If $N_t(w) \setminus \{b\}$ is empty then $\text{det-dir}_t(b, w)$ is not defined. Vertex $w' \in N_t(b) \setminus \{w\}$ which determines the minimum is denoted as $\text{det-dir}_t(w, b)$. If there are more such vertices we choose the first one in some predefined order on the vertices $B \cup W$. If $N_t(b) \setminus \{w\}$ is empty then $\text{det-dir}_t(w, b)$ is not defined.

It is easy to observe that if we take a connected component of a vertex $v \in W \cup B_t$ and we indicate v as a root then we obtain a rooted tree T and the following holds

$$\text{mini-max}_T(u) = \begin{cases} \text{det-dist}_t(\text{parent}_T(u), u) & \text{if } u \neq v, \\ \text{dist}_t(u) & \text{otherwise} \end{cases}$$

for all $u \in V(T)$. As a consequence we obtain two observations that follow.

Observation 21. *For $w \in W$, $b \in B_t$ the mini-max distance equals*

$$\begin{aligned} \text{dist}_t(b) &= \begin{cases} \min_{w' \in N_t(b)} \text{det-dist}_t(b, w') + 1 & \text{if } N_t(b) \neq \emptyset \\ \infty & \text{otherwise.} \end{cases} \\ \text{dist}_t(w) &= \begin{cases} \max_{b' \in N_t(w)} \text{det-dist}_t(w, b') + 1 & \text{if } N_t(w) \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Vertex $w' \in N_t(b)$ which determines the minimum equals $\text{dir}_t(b)$. If there are more such vertices this is the first one in some predefined order on the vertices $B \cup W$. If $N_t(b)$ is empty then $\text{dir}_t(b)$ is not defined. Vertex $b' \in N_t(w)$ which determines the maximum equals $\text{dir}_t(w)$. If there are more such vertices this is the first one in some predefined order on the vertices $B \cup W$. If $N_t(w)$ is empty then $\text{dir}_t(w)$ is not defined.

Observation 22. *For $w \in W$, $b \in B_t$ the mini-max distance equals*

$$\begin{aligned} \text{sec-dist}_t(b) &= \begin{cases} \min_{w' \in N_t(b) \setminus \{\text{dir}_t(b)\}} \text{det-dist}_t(b, w') + 1 & \text{if } N_t(b) \setminus \{\text{dir}_t(b)\} \neq \emptyset \\ \infty & \text{otherwise.} \end{cases} \\ \text{sec-dist}_t(w) &= \begin{cases} \max_{b' \in N_t(w) \setminus \{\text{dir}_t(w)\}} \text{det-dist}_t(w, b') + 1 & \text{if } N_t(w) \setminus \{\text{dir}_t(w)\} \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \end{aligned}$$

Vertex $w' \in N_t(b) \setminus \{\text{dir}_t(b)\}$ which determines the minimum equals $\text{sec-dir}_t(b)$. If there are more such vertices this is the first one in some predefined order on the vertices $B \cup W$. If $N_t(b) \setminus \{\text{dir}_t(b)\}$ is empty then $\text{sec-dir}_t(b)$ is not defined. Vertex $b' \in N_t(w) \setminus \{\text{dir}_t(w)\}$ which determines the maximum equals $\text{sec-dir}_t(w)$. If there are more such vertices this is the first one

in some predefined order on the vertices $B \cup W$. If $N_t(w) \setminus \{\text{dir}_t(b)\}$ is empty than $\text{sec-dir}_t(w)$ is not defined.

The meaning of $\text{dir}_t(\cdot)$ and $\text{sec-dir}_t(\cdot)$ functions is given by the following observation which follows from Definition 13 and Observations 21 and 22.

Observation 23. For each $v \in B_t \cup W$ it holds that

$$\begin{aligned} \text{dist}_t(v) &= \text{det-dist}_t(v, \text{dir}_t(v)) + 1, \\ \text{sec-dist}_t(v) &= \text{det-dist}_t(v, \text{sec-dir}_t(v)) + 1. \end{aligned}$$

Observation 23 provides a formula for $\text{dist}_t(v)$ and $\text{sec-dist}_t(v)$ in terms of $\text{det-dist}_t(v, \cdot)$. If we want to determine $\text{det-dist}_t(u, v)$ by $\text{dist}_t(v)$ and $\text{sec-dist}_t(v)$ it is worth noting that $\text{dist}_t(v)$ almost always equals $\text{sec-dist}_t(v)$. The only exception is when $\text{dist}_t(v)$ is determined by u and equals $\text{sec-dist}_t(v)$.

Observation 24. For any edge $\{v, u\} \in E_t$ the following holds

$$\text{det-dist}_t(u, v) = \begin{cases} \text{dist}_t(v) & \text{if } u \neq \text{dir}_t(v) \\ \text{sec-dist}_t(v) & \text{otherwise.} \end{cases}$$

Because $\text{dist}(\cdot)$ is a min (max) taken over a bigger set than $\text{sec-dist}(\cdot)$ we have the following.

Observation 25. For $w \in W$ and $b \in B_t$ such that $\{w, b\} \in E_t$ the following is holds

$$\begin{aligned} \text{dist}_t(b) &\leq \text{det-dist}_t(w, b) \leq \text{sec-dist}_t(b), \\ \text{sec-dist}_t(w) &\leq \text{det-dist}_t(b, w) \leq \text{dist}_t(w). \end{aligned}$$

The function $\text{det-dist}_t(v, u)$, given an entry edge (v, u) to a vertex u , returns one plus the distance of some neighbor $\text{det-dir}_t(v, u)$ of u other than v . This defines a determined mini-max path in a natural recursive way. An undetermined mini-max path can be defined using the notion of a determined mini-max path.

Definition 14. For any $t = 0, \dots, n$ and $\{v, u\} \in E_t$ we define

$$\text{det-path}_t(v, u) = \begin{cases} v \cdot \text{det-path}_t(u, \text{det-dir}_t(v, u)) & \text{if } \text{det-dir}_t(v, u) \text{ exists,} \\ v \cdot u & \text{otherwise.} \end{cases}$$

We observe the following.

Observation 26. Let $t = 0, \dots, n$ and $v \in B_t \cup W$. Then

$$\begin{aligned} \text{path}_t(v) &= \begin{cases} \text{det-path}_t(v, \text{dir}_t(v)) & \text{if } \text{dir}_t(v) \text{ exists,} \\ v & \text{otherwise.} \end{cases} \\ \text{sec-path}_t(v) &= \begin{cases} \text{det-path}_t(v, \text{sec-dir}_t(v)) & \text{if } \text{sec-dir}_t(v) \text{ exists,} \\ v & \text{otherwise.} \end{cases} \end{aligned}$$

The next observation shows a relation between mini-max functions and the lengths of the corresponding mini-max paths.

Observation 27. For any $t = 0, \dots, n$ and any edge $\{v, u\} \in E_t$ and $x \in W \cup B_t$ we have that

$$\begin{aligned} \|\text{det-path}_t(v, u)\| &= \text{det-dist}_t(v, u) && \text{if } \text{det-dist}_t(v, u) < \infty, \\ \|\text{path}_t(x)\| &= \text{dist}_t(x) && \text{if } \text{dist}_t(x) < \infty, \\ \|\text{sec-path}_t(x)\| &= \text{sec-dist}_t(x) && \text{if } \text{sec-dist}_t(x) < \infty. \end{aligned}$$

We are now ready to prove the following interesting property that says that the level of the $\text{dir}(\cdot)$ and $\text{sec-dir}(\cdot)$ can not change by more than ± 1 .

Lemma 15. For $v \in W \cup B_t$ and $u \in \{\text{dir}_t(v), \text{sec-dir}_t(v)\}$ it holds that $|\text{level}_t(v) - \text{level}_t(u)| \leq 1$.

Proof. We present the proof for $v = w \in W$ and then $u = b \in B_t$. The case when $v \in B_t$ is symmetric. We split this case to the following four subcases.

First subcase : We assume that $\text{dir}_t(w) = b$ and $\text{dir}_t(b) = w$. Then

$$\begin{aligned} \text{level}_t(b) &=^{\text{Def.11.}} \text{dist}_t(b) =^{\text{Obs.23.}} \det\text{-dist}_t(b, w) + 1 =^{\text{Obs.24.}} \\ &= \text{sec-dist}_t(w) + 1 =^{\text{Def.11.}} \text{level}_t(w) + 1. \end{aligned}$$

Second subcase : We assume that $\text{dir}_t(w) = b$ and $\text{dir}_t(b) \neq w$. Then

$$\begin{aligned} \text{level}_t(w) &=^{\text{Def.11.}} \text{sec-dist}_t(w) \leq^{\text{Obs.25.}} \text{dist}_t(w) =^{\text{Obs.23.}} \\ &= \det\text{-dist}_t(w, b) + 1 =^{\text{Obs.24.}} \text{dist}_t(b) + 1 =^{\text{Def.11.}} \text{level}_t(b) + 1. \end{aligned}$$

On the other side we have that

$$\begin{aligned} \text{level}_t(b) &=^{\text{Def.11.}} \text{dist}_t(b) =^{\text{Obs.21.}} \min_{w' \in N_t(b)} \det\text{-dist}_t(b, w') + 1 \leq \\ &= \det\text{-dist}_t(b, w) + 1 =^{\text{Obs.24.}} \text{sec-dist}_t(w) + 1 =^{\text{Def.11.}} \text{level}_t(w) + 1. \end{aligned}$$

Third subcase : We assume that $\text{sec-dir}_t(w) = b$ and $\text{dir}_t(b) = w$. Then

$$\begin{aligned} \text{dist}_t(b) + 2 &\leq^{\text{Obs.25.}} \det\text{-dist}_t(w, b) + 2 =^{\text{Obs.23.}} \text{sec-dist}_t(w) + 1 \leq^{\text{Obs.25.}} \\ &\leq \det\text{-dist}_t(b, w) + 1 =^{\text{Obs.23.}} \text{dist}_t(b) \end{aligned}$$

but it is not possible.

Fourth subcase : The last case describes situation when $\text{sec-dir}_t(w) = b$ and $\text{dir}_t(b) \neq w$. Then

$$\begin{aligned} \text{level}_t(w) &=^{\text{Def.11.}} \text{sec-dist}_t(w) =^{\text{Obs.23.}} \det\text{-dist}_t(w, b) + 1 =^{\text{Obs.24.}} \\ &= \text{dist}_t(b) + 1 =^{\text{Def.11.}} \text{level}_t(b) + 1. \end{aligned}$$

□

B. BOSEK: THEORETICAL COMPUTER SCIENCE DEPARTMENT, FACULTY OF MATHEMATICS AND COMPUTER SCIENCE, JAGIELLONIAN UNIVERSITY, KRAKÓW, POLAND

E-mail address: bosek@tcs.uj.edu.pl

D. LENIOWSKI, P. SANKOWSKI, AND A. ZYCH-PAWLEWICZ: INSTITUTE OF COMPUTER SCIENCE, UNIVERSITY OF WARSAW, POLAND

E-mail address: d.leniowski@mimuw.edu.pl

E-mail address: sank@mimuw.edu.pl

E-mail address: anka@mimuw.edu.pl