Competitive Maintenance of Minimum Spanning Trees under Stochastic Adversaries

Jarosław Kutyłowski

International Graduate School of Dynamic Intelligent Systems,
Heinz Nixdorf Institute, University of Paderborn, Germany

jarekk@upb.de

Abstract. We consider the problem of maintaining a minimum spanning tree within a graph with dynamically changing edge weights. An online algorithm is given an input sequence consisting of edge weight increases. It has to choose a minimum spanning tree after each such change in the graph. The task of the algorithm is to construct in each round a minimum spanning tree as similar to that from the previous round as possible. This problem occurs naturally in a dynamic network, where a stable communication backbone should be maintained.

We compare the number of changes in the minimum spanning trees produced by an online algorithm and that produced by an optimal offline algorithm. The number of changes is counted in the number of different edges between two minimum spanning trees in consecutive rounds.

We analyze the performance of the algorithm SimpleMST for this problem. It is known that the competitive ratio of such an algorithm on a graph with n vertices is at least $\Omega(\log n)$ assuming that the input sequence is constructed by an oblivious worst-case adversary. We show that for certain types of input sequences generated by two families of stochastic processes the competitive ratio of SimpleMST is constant with a high probability, substantially improving the performance achievable for worst-case adversaries.

1 Introduction

The problem of maintaining a minimum spanning tree in a dynamically changing graph is prominent in many applications. A communication topology based on a minimum spanning tree minimizes the total length of communication paths which must be held up. This property makes the MST a best-choice for scenarios where the length of the communication path is the most cost intensive factor for a communication backbone.

Within graphs with edges with equal weight, the minimum spanning tree may not be unique. Given a graph changing dynamically in terms of edge weights it is hard to find a minimum spanning tree which will retain its minimum property for a long time. We model the changes of the graph as an online sequence and compute costs of algorithms as the number of changes in the minimum spanning tree maintained by them.

1.1 Motivation

The problem of competitive maintenance of a minimum spanning tree occurs naturally in a robotic scenario, with several teams of robots deployed on an terrain. These teams operate on a large area and are, due to a restricted transmission range of communication devices, not able to communicate with each other on larger distances directly. On the other hand

the tasks that the robots have to perform require a communication backbone between the separated teams.

We propose to use mobile relay stations which form communication paths between teams of robots. We model robot teams as nodes of a communication graph. A possible path where relay stations can create a link is modeled as an edge in this communication graph – the length of the path is mapped to edge weight. Since the number of relay stations needed to support a communication path is proportional to its length, it is important to have a system of paths which connects the whole graph and on the other hand minimizes the total length of communication paths used – a minimum spanning tree is an optimal structure for this task. Furthermore it is advantageous when the number of changes in the minimum spanning tree is as small as possible, since then mobile relay stations have to travel least (to implement the changes occuring in the communication structure defined by the minimum spanning tree).

1.2 Related work

Research on minimum spanning trees dates back to textbook algorithms due to Kruskal [13] and Prim [15]. With regard to classical results, improved solutions have been considered e.g in [6]. This work assumes that the graph remains static and considers the classical runtime complexity of algorithms. Research in this classic area is still vivid, see e.g. recent results by Chazelle [5] and Pettie [14].

Large effort has been put into constructing data structures which also allow minimum spanning trees to be computed efficiently when changes in the structure of the graph occur. These changes can either concern edge weights as assumed in our work (see e.g. [11]) or might encompass adding and deleting vertices ([7, 10, 12]).

Furthermore kinetic spanning trees have been considered in [1] to model the changes of edge lengths in a more predictable way. This allows planning of changes in the minimum spanning tree in advance, in comparison to a strictly online problem where the future cannot be foreseen.

An online problem similar to our has been recently presented in [16] and analyzes the competitiveness of a minimum spanning tree algorithm which receives the weights of edges of a random graph as a sequence and has immediately to decide whether to include an edge into the MST or not.

For a survey of different types of subproblems in the area of minimum spanning trees, for applications and results see [9].

The problem of competitively maintaining a minimum spanning tree has been considered already in [8]. The same cost measure has been applied as in our work – the authors have presented lower and upper bounds of $\Theta(n^2)$ for the competitive ratio of deterministic algorithms solving the problem of competitive maintenance of minimum spanning trees. They have shown a lower bound of $\Omega(\log n)$ for randomized algorithms and provided a randomized algorithm with competitive ratio of $\mathcal{O}(n \log n)$ for general graphs. For planar graphs, their randomized algorithm achieves an optimal expected competitive ratio of $\mathcal{O}(\log n)$.

1.3 Our contribution

We show that the lower bound of $\Omega(\log n)$ for the problem of competitive maintenance of minimum spanning trees can be substantially improved if we assume a weaker version of the

adversary. The assumption of having a worst-case adversary as considered in [8] is not very realistic, since in practice input sequences won't be that hard. We therefore try to impose limitations on the adversary. This leads to the definition of a stochastic adversary, which chooses input sequences by means of a stochastic process.

We examine two families of stochastic processes in Section 3 and 4. Section 2 introduces the SimpleMST algorithm and shows that it has a constant competitive ratio on input sequences having a certain property. In Section 3 and 4 we prove that both stochastic process families produce input sequences preserving this property with a high probability. In both sections we will be able to conclude that the competitive ratio of SimpleMST is constant with a high probability.

1.4 Model

We consider a connected graph $G = (V, E)$ with edges weighted initially by the function $w^0\colon\ E \mapsto \mathbb{N}_+$. We divide time into discrete time steps called rounds. The value of $\sigma(i) \in E$ defines the edge whose weight is increased in round i. The sequence σ is the input sequence.

Basing on the original input sequence we denote for convenience of notation by $\delta(i, e) \in \{0, 1\}$ the change of weight of edge $e \in E$ in the i-th round. Formally we have

$$\delta(i, e) = \begin{cases} 1, & \text{if } \sigma(i) = e \\ 0, & \text{otherwise .} \end{cases}$$

Furthermore we introduce the function $w\colon \mathbb{N}_+ \times E \mapsto \mathbb{N}_+$ which maps a round number r and edge e to the edge weight at the beginning at round r, thus

$$w^r(e) = w^0(e) + \sum_{i=1}^{r-1} \delta(i, e) \ .$$

An algorithm ALG solving the Online Dynamic Minimum Spanning Tree (ODMST) problem works on the graph $G = (V, E)$ and reads the input sequence σ. After obtaining $\sigma(r)$ it outputs a minimum spanning tree, denoted by $\mathcal{M}^r_{\text{ALG}}$. Since the algorithm does not know the future, it has no access to the values of $\sigma(i)$ for $i > r$ in round r. The cost of an algorithm in round r is defined as the number of edges in which $\mathcal{M}^{r-1}_{\text{ALG}}$ and $\mathcal{M}^r_{\text{ALG}}$ differ, formally

$$\mathcal{C}^r_{\text{ALG}} := \left|\{\, e \in E \mid e \notin \mathcal{M}^{r-1}_{\text{ALG}} \wedge e \in \mathcal{M}^r_{\text{ALG}} \,\}\right| \ .$$

Additionally $\mathcal{C}_{\text{ALG}}(\sigma)$ is the cost of the algorithm on the whole sequence, thus $\mathcal{C}_{\text{ALG}}(\sigma) = \sum_{i=1}^{|\sigma|} \mathcal{C}^i_{\text{ALG}}$.

We analyze the performance of algorithms for the ODMST problem applying competitive analysis, as defined in [3]. Appropriately, we define an optimal algorithm OPT computing a sequence of minimum spanning trees $\mathcal{M}^i_{\text{OPT}}$ for $i = 1, \ldots, |\sigma|$ minimizing the value of $\mathcal{C}_{\text{OPT}}(\sigma)$, with $\mathcal{C}_{\text{OPT}}(\sigma) = \sum_{i=1}^{|\sigma|} \mathcal{C}^i_{\text{OPT}}$, where

$$\mathcal{C}^r_{\text{OPT}} := \left|\{\, e \in E \mid e \notin \mathcal{M}^{r-1}_{\text{OPT}} \wedge e \in \mathcal{M}^r_{\text{OPT}} \,\}\right| \ .$$

The optimal algorithm has access to the whole sequence σ in advance.

Following [4, 2] we introduce the notion of a stochastic adversary which creates the input sequence by means of a stochastic process. The stochastic process chooses the input sequence

according to some probability distribution π. We denote $m = |E|$ and adapt classical competitive analysis to match the definition of stochastic adversary. We say that a deterministic algorithm ALG solving the ODMST problem has a competitive ratio of $\mathcal{R}_{\mathrm{ALG}}$ with a probability of $1/m^{\alpha}$ if

$$\mathbf{Pr}_{\sigma}\left[\mathcal{C}_{\mathrm{ALG}}(\sigma) \leq \mathcal{R}_{\mathrm{ALG}} \cdot \mathcal{C}_{\mathrm{OPT}}(\sigma) + k\right] \geq 1 - \frac{1}{m^{\alpha}} \ .$$

The described probability is calculated with respect to the probability distribution π over all input sequences. The constant k may depend on the distribution π, the parameter α and the input graph G.

1.5 Stochastic adversaries

We analyze two notions of stochastic adversaries, described by families of stochastic processes. In the independent stochastic process defined in Section 3 the weight of an edge $e \in E$ is increased in each round independently from other edges with some probability p_e. Then a transformation is applied to the constructed sequence, so that only one edge weight is increased per round. This is necessary to match the definition of the ODMST problem.

In the second stochastic process we increase the weight of exactly one edge e in each round. Edge e is chosen with a probability distribution π from the set E in each round. This process resembles the worst case scenarios from [8], where the worst-case adversary is restricted to increase only one edge in each round.

Both stochastic adversaries are in fact families of stochastic processes since they are parametrized and can be tailored to specific needs. Due to this versatility, we motivate that most practically occuring input sequences can be modeled by these processes.

2 SimpleMST algorithm

In this section we introduce the algorithm SimpleMST (Algorithm 1) and show that it achieves a constant competitive ratio on a subset of all input sequences. The SimpleMST algorithm given here is a simplification of the MSTMark algorithm given in [8]. Before explaining the way in which SimpleMST works, we have to introduce some notation.

Basic notation. The set of alternative edges $\mathcal{A}(e, r)$ is defined for a graph $G = (V, E)$, a round r, an algorithm ALG and an edge $e \in \mathcal{M}^r_{\mathrm{ALG}}$. Removing e from $\mathcal{M}^r_{\mathrm{ALG}}$ splits the tree $\mathcal{M}^r_{\mathrm{ALG}}$ into two parts. Consider the vertex sets V_1 and V_2 of both parts. Then the set of edges on the cut between V_1 and V_2 is denoted by

$$\mathcal{A}(e, r) = \{\, (u, v) \in E \mid u \in V_1 \wedge v \in V_2 \,\} \ .$$

We also define a set of alternatives which have a certain weight $\mathcal{A}_{\mathrm{W}}(e, r, w) = \{\, e' \in \mathcal{A}(e, r) \mid w^r(e') = w \,\}$. We extend $\mathcal{A}_{\mathrm{W}}(\cdot)$ by defing it not only for one edge, but also for a set of edges E' as

$$\mathcal{A}_{\mathrm{W}}(E', r, w) = \bigcup_{e \in E'} \mathcal{A}_{\mathrm{W}}(e, r, w) \ .$$

SimpleMST works in the following way. Consider a round r in which the weight of an edge $e \in \mathcal{M}_{\mathrm{ALG}}$ is increased. If there exist alternative edges for e with weight $w^{r-1}(e)$,

Algorithm 1 SimpleMST

1: **if** $e \in \mathcal{M}^{r-1}_{\text{ALG}}$ increased in round r **and** $\mathcal{A}_{\text{W}}(e, r-1, w^{r-1}(e)) \neq \emptyset$ **then**
2: $\quad e' \leftarrow$ choose one edge out of $\mathcal{A}_{\text{W}}(e, r-1, w^{r-1}(e))$
3: $\quad$ remove e from $\mathcal{M}^{r}_{\text{ALG}}$ and substitute it with e'
4: **end if**

then SimpleMST selects one of these edges uses it instead of e in $\mathcal{M}^{r}_{\text{ALG}}$. In other cases SimpleMST ignores the edge weight increase. This technique guarantees that each $\mathcal{M}^{r}_{\text{ALG}}$ is always a minimum spanning tree, as shown in [11] and used previously in [8]. We say that SimpleMST incurs a cost on an edge e if the weight of edge e is increased and then e must be substituted with another edge e' so that the weight of the MST is maintained.

We define the history of some edge $e \in \mathcal{M}^{r}_{\text{ALG}}$ as the sequence $h_e = [(e_1, r_1), (e_2, r_2), \ldots]$ such that e substituted e_1 in $\mathcal{M}^{r_1}_{\text{ALG}}$ in some round r_1. For the rest of the sequence it holds that e_i substituted e_{i+1} in $\mathcal{M}^{r_{i+1}}_{\text{ALG}}$ in some round r_{i+1}. Naturally, $h_e(i)$ denotes the i-th element of the sequence. If the history for e does not contain any entries, we say $h_e = \emptyset$. Informally, this sequence represents the "movement sequence" of one edge from $\mathcal{M}_{\text{ALG}}$ over the graph G.

We do not make any assumptions on the initial tree $\mathcal{M}^{0}_{\text{ALG}}$ used by SimpleMST. The following argumentation works for any choice of $\mathcal{M}^{0}_{\text{ALG}}$. If not instructed to do otherwise, SimpleMST chooses one of the possible minimum spanning trees for $\mathcal{M}^{0}_{\text{ALG}}$.

In the remaining part of this section, we will analyze the performance of SimpleMST under the assumption that in each round there are at most c edges with equal weight. This assumption will be than proven to be right for most input sequences for both types of stochastic adversaries.

Hits and number of edges. A hit occurs when the weight of an edge e is increased in round r, and $e \in \mathcal{M}^{r-1}_{\text{ALG}}$, and the weight of the minimum spanning tree does not increase in this round. Additionally either $h_e = \emptyset$ or $h_e(1) = (e', r')$ exists and it holds $w^{r'}(e') < w^{r}(e)$. In other words, SimpleMST is forced to perform a change in its tree (the **if** in line 1 of SimpleMST evaluates to true) in round r and the predecessor of e in its history had a weight smaller than w. We will often say that a hit occurred on edge e if increasing the weight of edge e caused the hit.

We denote the set of edges with weight w in round r by $E(w, r) \subseteq E$. The cardinality of the set $E(w, r)$ is denoted by $\#E(w, r)$. As already mentioned, we will assume that $\#E(w, r) \leq c$ for all w and r for this analysis.

Cost assignment scheme. We assign edges to hits. If a hit H occurs on an edge with weight w in round r, all edges in $E(w, r)$ are assigned to H. An edge remains assigned to a hit until its weight has been increased or it is assigned to another hit. Every time SimpleMST incurs cost on an edge e assigned to H (the weight of edge e is increased and SimpleMST must remove e from its spanning tree), this cost is assigned to H. This includes the cost created on the hit.

Since we have $\#E(w, r) \leq c$ for all w, r the number of edges assigned to one hit is at most c. So, the maximum cost which can be assigned to one hit is c. The following lemma shows that all cost of SimpleMST is assigned to hits. Its technical proof can be found in the appendix.

Lemma 1. *Assume the weight of an edge e is increased in round r, such that $e \in \mathcal{M}^{r-1}_{\mathrm{ALG}}$ and $e \notin \mathcal{M}^{r}_{\mathrm{ALG}}$. Then either this edge is assigned to some hit, or a hit occurs at this moment.*

Cost of optimal algorithm. In the previous considerations we have defined a scheme which assigns all cost of SimpleMST to hits, such that each hit is assigned a cost of at most c. What is still left is some mapping of hits to OPT's costs.

Consider the case when in a round r the weight of an edge $e \in \mathcal{M}^{r-1}_{\mathrm{ALG}}$ is increased and SimpleMST has no alternatives for e. Then, obviously SimpleMST leaves $e \in \mathcal{M}^{r}_{\mathrm{ALG}}$ and the weight of the minimum spanning is increased in r. Then the weight of OPT's minimum spanning tree must increase in r. Since only the weight of one edge is increased in r, it must hold that $e \in \mathcal{M}^{r-1}_{\mathrm{OPT}}$ and $e \in \mathcal{M}^{r}_{\mathrm{OPT}}$, otherwise the weight of its tree would not increase.

On the contrary, when in a round r the weight of an edge $e \in \mathcal{M}^{r-1}_{\mathrm{ALG}}$ and $e \in \mathcal{M}^{r-1}_{\mathrm{OPT}}$ is increased and SimpleMST has an alternative for e, then $e \notin \mathcal{M}^{r}_{\mathrm{OPT}}$. If OPT would leave $e \in \mathcal{M}^{r}_{\mathrm{OPT}}$ then the weight of $\mathcal{M}_{\mathrm{OPT}}$ would increase in round r where the weight of $\mathcal{M}_{\mathrm{ALG}}$ does not increase.

Consider a hit H on edge e in round r. Assume that the entry $h_e(1) = (e', r')$ exists. Then we have $w^{r'}(e) < w^{r}(e)$ and there exists at least one round r'' with $r > r'' > r'$ such that the the weight of e is increased in r''. Obviously we have $e \in \mathcal{M}^{r''-1}_{\mathrm{ALG}}$ and $e \in \mathcal{M}^{r''}_{\mathrm{ALG}}$. Thus, the weight of the minimum spanning tree has increased in r'' and by the argumentation from the previous paragraph it must hold $e \in \mathcal{M}^{r''}_{\mathrm{OPT}}$. So, it either still holds that $e \in \mathcal{M}^{r-1}_{\mathrm{OPT}}$ and OPT has to remove e from $\mathcal{M}_{\mathrm{OPT}}$ in r or OPT must have removed e from $\mathcal{M}_{\mathrm{OPT}}$ between rounds r'' and $r-1$. This movement caused a unit cost to OPT and we assign this cost to the hit H.

On the other hand, if we have $h_e = \emptyset$ we have no OPT's cost to assign to hit H. Fortunately we can bound the number of hits having this property. To accomplish this, assign unique identifiers to all edges of $\mathcal{M}^{0}_{\mathrm{ALG}}$. Every time an edge e is substituted with another edge e' in $\mathcal{M}_{\mathrm{ALG}}$ we move the identifier of e to e'. This way, we can track a specific edge of $\mathcal{M}_{\mathrm{ALG}}$ with its identifier. Considering all hits occurring on edges having assigned an identifier u, there is at most one hit such that $h_e = \emptyset$ and e is assigned u. This is the first hit on u in the input sequence. Every next hit on edge e, such that e has the identifier u assigned will already have $h_e(1)$ existing, because of the previous hit.

With this assignment, we can map every hit to a unit cost of OPT and this mapping is injective but for $n = |V|$ hits occurring at the beginning. We can conclude this argument with the following corollary.

Corollary 1. *Assume an input sequence σ such that $\#E(w, r) \leq c$ for all r, w. Then for algorithm* SimpleMST *it holds*

$$\mathcal{C}_{\mathrm{ALG}}(\sigma) \leq c \cdot \mathcal{C}_{\mathrm{OPT}}(\sigma) + n \ .$$

3 Independent stochastic process

Consider all sequences with a length r_{max} generated by the independent stochastic process. We will show that only few of these sequences have $\max_{w \in W \wedge r \in R} \#E(w, r) > c$, with $R = \{r_s, \ldots, r_{max}\}$ and $W \subseteq R$ for an appropriate r_s. We assume the following stochastic process. We start with any input graph with all edge weights equal 0. In meta-round r, edge $e \in E$ executes a bernoulli random trial with success probability p_e, independently from other edges.

Edge e increases its weight by one in meta-round r if the experiment was successful and leaves its weight as it is otherwise. Since the ODMST problem assumes that at most one edge is increased in a round, we divide each meta-round into rounds, such that exactly one edge weight is increased in each round. This defines the input sequence σ. This division can be performed in any way, we do not pose any restrictions on it.

The success probability p_e can be different for each edge $e \in E$ and we denote the vector of probabilities $\pi = (p_e)_{e \in E}$. We denote $m = |E|$ for the input graph $G = (V, E)$. For a stochastic adversary defined by the vector π define β to be the smallest value such that

$$p_e(1 - p_e) \geq \frac{1}{m^\beta} \ .$$

We will show that for any input sequence length given, a constant competitive ratio can be achieved with a high probability. For this, we introduce the Proposition 1 which formally states the fact that the greatest part of most input sequences generated by the described stochastic process has less than c edges with equal weight in every meta-round. Basing on the fact that in a meta-round each edge increases its weight at most once, if there are never more than c edges with equal weight in a meta-round, there can be never more than $2c$ edges with equal weight in a round, no matter how the division of meta-round into rounds falls out. This is because in a worst case, in the division into rounds all edges with weight $w - 1$ are first increased to w and then the number of edges with weight w can reach $2c$.

Proposition 1. *For any r_{max} and a set of meta-rounds $R \subseteq \{m^{\phi(\alpha,c,\beta,\varepsilon)}, \ldots, r_{max}\}$, set of weights $W \subseteq R$, an appropriate constant c and β defined as above it holds*

$$\mathbf{Pr}\left[\max_{\substack{w \in W \\ r \in R}} \#E(w, r) > c\right] \leq \frac{1}{m^\alpha} \ .$$

To prove the property, we introduce the following lemma.

Lemma 2. *For a fixed meta-round $r \geq m^{\frac{c(1+\beta/2)+1+\alpha-\log_m \varepsilon}{c/2-2-\varepsilon}}$, fixed weight $w \leq r$, an appropriate $\varepsilon > 0$, and β defined as above it holds*

$$\mathbf{Pr}\left[\#E(w, r) > c\right] \leq \frac{1}{m^{\alpha - \log_m \varepsilon}\, r^{2+\varepsilon}} \ .$$

Proof. Let us look at one specific edge e. The probability that the weight of e is exactly w in meta-round r is

$$\mathbf{Pr}[w^r(e) = w] = \binom{r}{w} p_e^w \left(1 - p_e\right)^{r-w} \ .$$

Let us consider $\mathbf{Pr}[w^r(e) = w]$ as a function of w. It assumes its maximum value for $w = rp_e$. Substituting w with rp_e and applying Stirling's formula, we can upper bound the value of the following expression

$$\max_{w \in \{0,\ldots,r\}} \mathbf{Pr}\left[w^r(e) = w\right] \leq \frac{1}{\sqrt{rp_e(1 - p_e)}} \ .$$

Furthermore, by assumption $p_e(1 - p_e) \geq 1/m^\beta$ for all $e \in E$ we have

$$\max_{\substack{w \in \{0,\ldots,r\} \\ e \in E}} \mathbf{Pr}\left[w^r(e) = w\right] \leq \sqrt{m^\beta / r} \ . \tag{1}$$

Let us turn our attention to the probability $\mathbf{Pr}[\#E(w,r)=i]$ for a fixed meta-round r and a fixed weight w. This probability is defined and upper bounded as follows

$$\begin{aligned}
\mathbf{Pr}[\#E(w,r)=i] &= \sum_{\substack{E'\subseteq E\\|E'|=i}} \left[\prod_{e\in E'} \mathbf{Pr}[w^r(e)=w] \prod_{e\in E\setminus E'} (1-\mathbf{Pr}[w^r(e)=w])\right] \\
&\le \sum_{\substack{E'\subseteq E\\|E'|=i}} \left[\left(\max_{e\in E} \mathbf{Pr}[w^r(e)=w]\right)^i \left(1-\min_{e\in E} \mathbf{Pr}[w^r(e)=w]\right)^{m-i}\right] \\
&= \binom{m}{i} \left(\max_{e\in E} \mathbf{Pr}[w^r(e)=w]\right)^i \left(1-\min_{e\in E} \mathbf{Pr}[w^r(e)=w]\right)^{m-i} \\
&=: \widetilde{\mathbf{Pr}}[\#E(w,r)=i] \ .
\end{aligned}$$

We will write $\widetilde{\mathbf{Pr}}[\#E(w,r)=i]$ to denote the upper bound on $\mathbf{Pr}[\#E(w,r)=i]$ and aim for upper bounding $\widetilde{\mathbf{Pr}}[\#E(w,r)=c]$ with $1/(r^{2+\varepsilon}\, m^{1+\alpha-\log_m \varepsilon})$. So,

$$\begin{aligned}
\widetilde{\mathbf{Pr}}[\#E(w,r)=c] &= \binom{m}{c} \left(\max_{e\in E} \mathbf{Pr}\left[w^r(e)=w\right]\right)^c \left(1-\min_{e\in E} \mathbf{Pr}\left[w^r(e)=w\right]\right)^{n-c} \\
&\le m^c \left(\max\nolimits_{\substack{w\in\{0,\ldots,r\}\\ e\in E}} \mathbf{Pr}\left[w^r(e)=w\right]\right)^c \\
&\le m^c \left(m^\beta/r\right)^{c/2} \ . \qquad (2)
\end{aligned}$$

To have $\widetilde{\mathbf{Pr}}[\#E(w,r)=c] \le 1/\left(r^{2+\varepsilon}\, m^{1+\alpha-\log_m \varepsilon}\right)$ we need

$$m^c \left(m^\beta/r\right)^{c/2} \le \frac{1}{r^{2+\varepsilon}\, m^{1+\alpha-\log_m \varepsilon}} \ .$$

Using the upper bound from Eq. (1) we obtain

$$\begin{aligned}
m^c \ \frac{m^{c\beta/2}}{r^{c/2}} &\le \frac{1}{r^{2+\varepsilon}\, m^{1+\alpha-\log_m \varepsilon}} \ , \\
m^{c+c\beta/2+1+\alpha-\log_m \varepsilon} &\le r^{c/2-2-\varepsilon} \ , \\
m^{\frac{c(1+\beta/2)+1+\alpha-\log_m \varepsilon}{c/2-2-\varepsilon}} &\le r \ . \qquad (3)
\end{aligned}$$

Now we want to upper bound the probability that the number of edges with weight w in a meta-round r is greater than c

$$\begin{aligned}
\mathbf{Pr}[\#E(w,r)\ge c] &= \sum_{i=c}^{m} \mathbf{Pr}[\#E(w,r)=i] \le m \cdot \max_{i=c,\ldots,m} \mathbf{Pr}[\#E(w,r)=i] \\
&\le m \cdot \max_{i=c,\ldots,m} \widetilde{\mathbf{Pr}}[\#E(w,r)=i] \qquad (4) \\
&\le m \cdot \max_{i=c,\ldots,m} m^i \left(m^\beta/r\right)^{i/2} \ . \qquad (5)
\end{aligned}$$

If $\sqrt{m^\beta/r} \le 1/m$ then $m^c \left(m^\beta/r\right)^{c/2}$ is largest for $i=c$. Thus, if $r \ge m^{\beta+2}$, we have

$$\max_{i=c,\ldots,m} m^i \left(m^\beta/r\right)^{i/2} = m^c \left(m^\beta/r\right)^{c/2} \ .$$

So, it holds

$$\mathbf{Pr}[\#E(w,r) \geq c] \leq \frac{1}{r^{2+\varepsilon}\, m^{\alpha - \log_m \varepsilon}} \ ,$$

if r fulfills Eq. (3) and $r \geq m^{\beta+2}$. The last condition holds if r fulfills Eq. (3) and $\beta \geq \frac{\alpha-5-2\varepsilon}{2+\varepsilon}$. So, with a proper choice of ε the second property is fulfilled. This concludes the lemma. □

With the help of the previous lemma we will prove Proposition 1. Set ε large enough for the supplied values of α and β. Denote for sake of simplicity of notation $\phi(\alpha, c, \beta, \varepsilon) = \left(\frac{c(1+\beta/2)+1+\alpha+\log_m 1/\varepsilon}{c/2-2-\varepsilon}\right)$. Using the result from the former lemma we can approximate the following probability, for $R \subseteq \{m^{\phi(\alpha,c,\beta,\varepsilon)}, \ldots, r_{max}\}$ and $W \subseteq R$

$$\begin{aligned}
\mathbf{Pr}\left[\max_{w\in W \wedge r \in R} \#E(w,r) \geq c\right] &\leq \mathbf{Pr}\left[\bigcup_{w\in W \wedge r\in R} \#E(w,r) \geq c\right] \\
&\leq \sum_{w\in W \wedge r\in R \wedge w \leq r} \mathbf{Pr}[\#E(w,r) \geq c] \\
&\leq \sum_{w\in W \wedge r\in R \wedge w \leq r} \frac{1}{r^{2+\varepsilon} m^{\alpha - \log_m \varepsilon}} \leq \sum_{r\in R} r \frac{1}{r^{2+\varepsilon} m^{\alpha}} \\
&\leq \frac{1}{m^{\alpha-\log_m \varepsilon}} \sum_{r\in R} \frac{1}{r^{1+\varepsilon}} \\
&\leq \frac{1}{\varepsilon m^{\alpha-\log_m \varepsilon}} \leq \frac{1}{m^{\alpha}} \ .
\end{aligned}$$

The inequality holds because $\sum_{r\in R} \frac{1}{r^{1+\varepsilon}} \leq 1/\varepsilon$ (see Appendix A.2). This concludes the proof of Proposition 1.

We now want to compute the competitive ratio of the randomized algorithm SimpleMST for an input sequence which fulfills $\max_{r\in R \wedge w \in W} \#E(r,w) \leq c$, if c and R have been chosen according to the assumptions of Proposition 1.

We divide the input sequence σ in two parts and consider the competitive ratio of SimpleMST on both parts as on separate sequences. The first part σ_1 reaches from the first round to $m^{\phi(\alpha,c,\beta,\varepsilon)}$. The second part σ_2 consists of the rest of σ. We choose the initial tree for the input sequence σ_2 as the last $\mathcal{M}_{\mathrm{ALG}}$ computed by SimpleMST on σ_1. Then, we have $\mathcal{C}_{\mathrm{ALG}}(\sigma_1) + \mathcal{C}_{\mathrm{ALG}}(\sigma_2) = \mathcal{C}_{\mathrm{ALG}}(\sigma)$. For the second part σ_2 we have by Corollary 1 and the fact that we consider an input sequence consisting of rounds (and not meta-rounds)

$$\mathcal{C}_{\mathrm{ALG}}(\sigma_2) \leq 2c \cdot \mathcal{C}_{\mathrm{OPT}}(\sigma_2) + n \ . \tag{6}$$

Since σ_1 has a length of $m^{\phi(\alpha,c,\beta,\varepsilon)}$ it holds $\mathcal{C}_{\mathrm{ALG}}(\sigma_1) \leq m^{\phi(\alpha,c,\beta,\varepsilon)}$. Summing this with Eq. (6) we obtain

$$\mathcal{C}_{\mathrm{ALG}}(\sigma) \leq 2c \cdot \mathcal{C}_{\mathrm{OPT}}(\sigma_2) + n + m^{\phi(\alpha,c,\beta,\varepsilon)} \leq 2c \cdot \mathcal{C}_{\mathrm{OPT}}(\sigma) + m^{\phi(\alpha,c,\beta,\varepsilon)} + n \ . \tag{7}$$

Set $k = m^{\phi(\alpha,c,\beta,\varepsilon)} + n$. Consider now that the stochastic process creates a sequence of length r_{max}. If $r_{max} \leq m^{\phi(\alpha,c,\beta,\varepsilon)}$ then we obviously have $\mathcal{C}_{\mathrm{ALG}}(\sigma) \leq c \cdot \mathcal{C}_{\mathrm{OPT}}(\sigma) + k$ since k covers all possible cost of the algorithm on σ. If $r_{max} > m^{\phi(\alpha,c,\beta,\varepsilon)}$ then we can apply Proposition 1 and with probability of $1 - 1/m^{\alpha}$ we will be able to apply the result Eq. (7). This leads to the following theorem.

Theorem 1. *For the ODMST problem and for an input sequence σ chosen randomly by the independent stochastic process the competitive ratio of* SimpleMST *is constant with high probability.*

4 Dependent stochastic process

The independent stochastic process presented in the previous section has been designed so that exactly one edge weight is increased in each round, by splitting meta-rounds into rounds. On the other hand we can define a process, which increases only one edge weight at a round "by design". This is the most natural way to define a stochastic process generating input sequences for the ODMST problem.

The dependent process performs one random experiment per round, choosing one edge whose weight is increased in this round. The probability of choosing an edge e is equal to p_e. This values form the probability distribution $\pi = (p_e)_{e \in E}$, with $\sum_{e \in E} p_e = 1$.

We will use an appropriately parametrized independent stochastic $\mathcal{I}_\mathcal{P}$ process to simulate a dependent process $\mathcal{D}_\mathcal{P}$ with prob. dist. π. We will show that SimpleMST has with a high probability a constant competitive ratio on the input sequences generated by $\mathcal{I}_\mathcal{P}$. Furthermore this sequence will fulfill certain properties and so will be able to follow that SimpleMST has a constant competitive ratio on sequences generated by the dependent stochastic process. These properties are the following

- with high probability there is at most one edge weight increased per meta-round (*property A*),
- if exactly one edge weight is increased in a round, then this edge is chosen (approximately) according to the prob. dist. π (*property B*).

So, to obtain an input sequence distributed accordingly to $\mathcal{D}_\mathcal{P}$ one has to execute the $\mathcal{I}_\mathcal{P}$ process. From the created sequence all rounds with no edge weights increased can be removed. If the input sequence contains round with more than one edge weight increased then this input sequence be discarded and a new one should be generated (by property A this should lead to a success very fast). We will call an input sequence generated by $\mathcal{I}_\mathcal{P}$ valid if it contains at most one edge weight increase per meta-round. We will call a simulation of $\mathcal{D}_\mathcal{P}$ by $\mathcal{I}_\mathcal{P}$ proper, if property B is fulfilled.

Assume that $\mathcal{D}_\mathcal{P}$ has to generate a sequence with length r and probability distribution $\pi = (p_e)_{e \in E}$. Then the probability distribution for $\mathcal{I}_\mathcal{P}$ is defined to be $\pi' = (p'_e)_{e \in E}$, such that $p'_e = p_e/(r^{1/2} m^{1+\alpha})$ for $m = |E|$ and an $\alpha \geq 1$. The following two lemmas are proven in Appendix A.3. The first lemma shows property A, the second one shows that B is fulfilled.

Lemma 3. *Assume that the independent stochastic process generates a sequence of length r and it holds $p'_e \leq 1/(r^{1/2} m^{1+\alpha})$. Then with probability at least $1 - 1/m^\alpha$ this sequence contains no meta-rounds with more than one edge weight increased.*

Lemma 4. *Assume the independent stochastic process $\mathcal{I}_\mathcal{P}$ with $\pi' = (p'_e)_{e \in E}$ and $p'_e = p_e/k$ for the dependent stochastic process $\mathcal{D}_\mathcal{P}$ with $\pi = (p_e)_{e \in E}$ and a sufficiently large constant k. Consider a round generated by $\mathcal{I}_\mathcal{P}$ in which exactly one edge weight is increased. Then the weight of an edge e is increased with probability near to p_e.*

Let us now approximate the probability that SimpleMST will have a constant competitive ratio on an input sequence σ generated by $\mathcal{I}_{\mathcal{P}}$. For the selected probability distribution π' the input sequence σ is valid with a probability of at least $1 - 1/m^{\alpha}$. We neglect the fact that another input sequence is generated if σ is invalid and simply assume that SimpleMST has a non-constant competitive ratio in such a case. On the other hand, for a valid sequence SimpleMST has a competitive ratio with probability at least $1 - 1/m^{\alpha}$ by Theorem 1. So,

$$\mathbf{Pr}[\mathcal{C}_{\mathrm{ALG}}(\sigma) \leq c \cdot \mathcal{C}_{\mathrm{OPT}}(\sigma) + k] \geq \left(1 - \frac{1}{m^{\alpha}}\right)^2 \geq 1 - \frac{2}{m^{\alpha}} \ .$$

We can conclude with the following theorem.

Theorem 2. *For the ODMST problem and for an input sequence σ chosen randomly by the dependent stochastic process the competitive ratio of* SimpleMST *is constant with high probability.*

5 Conclusions

The investigation of stochastic adversaries for the ODMST problem yields new information about the competitive complexity of the ODMST problem. The results obtained for worst-case adversaries can be significantly improved if we enforce some restrictions on the adversaries. Furthermore, due to the variable distribution probabilities in the independent stochastic processes, the obtained results can be applied to many input sequences occurring in practice.

In the future, one should consider other ways of bounding the abilities of worst-case adversaries. Furthermore it is an interesting question whether the competitive ratios hold also for the case when input sequences are allowed to contain more than one edge increase per round.

References

1. Pankaj K. Agarwal, David Eppstein, Leonidas J. Guibas, and Monika R. Henzinger. Parametric and kinetic minimum spanning trees. In *FOCS '98: Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, page 596, Washington, DC, USA, 1998. IEEE Computer Society.
2. Marcin Bienkowski. Dynamic page migration with stochastic requests. In *Proceedings of the 17th annual ACM symposium on Parallelism in Algorithms and Architectures (SPAA)*, pages 270–278, 2005.
3. A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.
4. Allan Borodin, Jon Kleinberg, Prabhakar Raghavan, Madhu Sudan, and David P. Williamson. Adversarial queuing theory. *J. ACM*, 48(1):13–38, 2001.
5. Bernard Chazelle. A minimum spanning tree algorithm with inverse-ackermann type complexity. *J. ACM*, 47(6):1028–1047, 2000.
6. David Cheriton and Robert Endre Tarjan. Finding minimum spanning trees. In *SIAM Journal of Computing*, volume 5, 1976.
7. Francis Chin and David Houck. Algorithms for updating minimal spanning trees. In *Journal of Computer and System Sciences*, volume 16, pages 333–344, 1978.
8. Miroslaw Dynia, Miroslaw Korzeniowski, and Jaroslaw Kutylowski. Competitive maintenance of minimimum spanning trees in dynamic graphs. *manuscript*.
9. David Eppstein. Spanning trees and spanners. Technical Report ICS-TR-96-16, 1996.
10. David Eppstein, Zvi Galil, Giuseppe F. Italiano, and Amnon Nissenzweig. Sparsification a technique for speeding up dynamic graph algorithms. *J. ACM*, 44(5):669–696, 1997.
11. Greg N. Frederickson. Data structures for on-line updating of minimum spanning trees. In *STOC '83: Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 252–257, New York, NY, USA, 1983. ACM Press.
12. Monika Rauch Henzinger and Valerie King. Maintaining minimum spanning trees in dynamic graphs. In *ICALP '97: Proceedings of the 24th International Colloquium on Automata, Languages and Programming*, pages 594–604, London, UK, 1997. Springer-Verlag.
13. J.B. Kruskal. On the shortest spanning subtree of a graph and the traveling salesman problem. In *Proceedings of the American Mathematics Society*, volume 7, pages 48–50, 1956.
14. Seth Pettie and Vijaya Ramachandran. An optimal minimum spanning tree algorithm. *J. ACM*, 49(1):16–34, 2002.
15. R.C. Prim. Shortest connection networks and some generalizations. In *Bell System Technical Journal*, volume 36, pages 1389–1401, 1957.
16. Jen Remy, Alexander Souza, and Angelika Steger. On an online spanning tree problem in randomly weighted graphs. In *Combinatorics, Probability and Computing*, 2005.

A Appendix

A.1 Technical proofs for the analysis of SimpleMST

Proof (of Lemma 1). To prove this lemma we first introduce the following notations and then state an important claim. Let

$$E_H(w,r) = \{\, e \in \mathcal{M}^r_{\mathrm{ALG}} \mid w^r(e) = w \wedge h_e(1) = (e', r') \wedge w^{r'}(e') = w \,\} \ .$$

Informally speaking, $E_H(w,r) \subseteq \mathcal{M}^r_{\mathrm{ALG}}$ describes the set of edges having weight w in round r having a hit with weight w in their history. We will proceed with stating a claim, proving it and then following the lemma.

Claim. Fix some weight w and round r. If we have

$$\mathcal{A}_{\mathrm{W}}(E_H(w,r), r, w) \not\subset \mathcal{A}_{\mathrm{W}}(E_H(w, r-1), r-1, w))$$

then a hit occurred in round r on an edge with weight w.

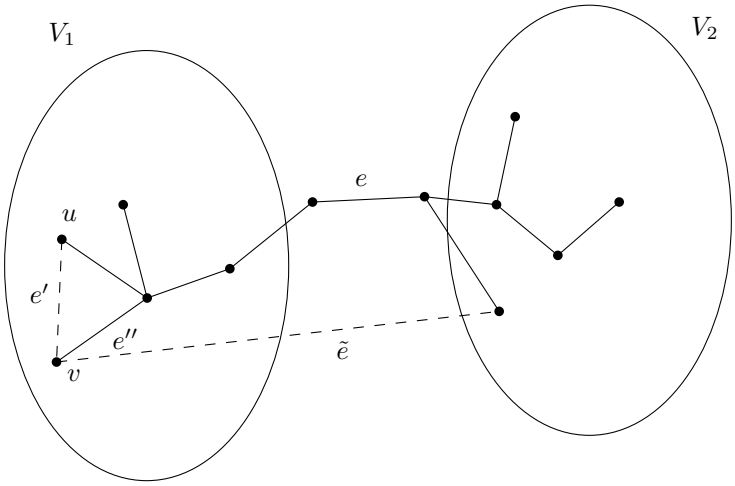

Fig. 1. Tree $\mathcal{M}^{r-1}_{\mathrm{ALG}}$

Since $\mathcal{A}_{\mathrm{W}}(E_H(w,r), r, w) \not\subset \mathcal{A}_{\mathrm{W}}(E_H(w, r-1), r-1, w))$ there exists an edge $e' = (u, v)$ with $e' \in \mathcal{A}_{\mathrm{W}}(E_H(w,r), r, w) \wedge e' \notin \mathcal{A}_{\mathrm{W}}(E_H(w, r-1), r-1, w))$ and $w^r(e') = w$. Consider an edge $e \in E_H(w,r)$ with $w^r(e) = w$, such that $e' \notin \mathcal{A}_{\mathrm{W}}(e, r-1, w)$ but $e' \in \mathcal{A}_{\mathrm{W}}(e, r, w)$. Take the tree $\mathcal{M}^{r-1}_{\mathrm{ALG}}$ as shown on Fig. 1, with vertex sets V_1 and V_2 split by e. Since $e' \notin \mathcal{A}_{\mathrm{W}}(e, r-1, w)$ both vertices u, v are included in one of the sets V_1, V_2. Assume without loss of generality that $u, v \in V_1$.

Since $e' \in \mathcal{A}_{\mathrm{W}}(e, r, w)$ the structure of $\mathcal{M}_{\mathrm{ALG}}$ must have changed so that one of the vertices u, v changed to V_2 in round r. Once again assume w.l.o.g that $v \in V_2$ after round r. Then, some edge $e'' \in \mathcal{M}^{r-1}_{\mathrm{ALG}}$ which connected a subtree of $\mathcal{M}^{r-1}_{\mathrm{ALG}}$ containing v to the rest of V_1 must have been substituted with an edge $\tilde{e}$ in $\mathcal{M}^r_{\mathrm{ALG}}$. Edge $\tilde{e}$ must connect the mentioned

subtree with v to a vertex from V_2. So, the weight of e'' must have been increased in round r. Obviously $\tilde{e}$ cannot have a weight smaller than e in round r, since we could substitute e with $\tilde{e}$ in $\mathcal{M}^{r-1}_{\text{ALG}}$ and decrease the weight of $\mathcal{M}^{r-1}_{\text{ALG}}$. On the other hand, if $w^r(\tilde{e}) > w^r(e) = w^r(e')$ we could substitute $\tilde{e}$ with e' in $\mathcal{M}^{r}_{\text{ALG}}$ decreasing its weight. Thus, the only possibility is that $w^r(\tilde{e}) = w^r(e)$. So, we have $w^{r-1}(e'') = w^r(e)$.

Consider now the edge e'' and the two subtrees of $\mathcal{M}^{r-1}_{\text{ALG}}$ obtained by removing e'' from $\mathcal{M}^{r-1}_{\text{ALG}}$. One of these subtrees contains u and the other one contains v (by the definition of edge e''). Thus, e' is an alternative for e'' and by previous considerations it has the same weight, so that $e' \in \mathcal{A}_{\text{W}}(e'', r-1, w^{r-1}(e''))$. We now want to consider two cases. If $e'' \in E_H(w, r-1)$ we have a contradiction, since then $e' \in \mathcal{A}_{\text{W}}(E_H(w, r-1), r-1, w)$.

Thus assume that $e'' \notin E_H(w, r-1)$. Then, by increasing the weight of e'' in round r a hit occurs on e'', as stated in the definition of the claim.

We now proceed with the proof of the lemma. Assume that a hit H occurred on an edge with weight w in round r. Then all edges with weight w in round r have been assigned to H, in particular all edges in $\mathcal{A}_{\text{W}}(E_H(w, r), r, w)$. Till the next hit H' in round r' we have be the previous claim $\mathcal{A}_{\text{W}}(E_H(w, r''), r'', w) \subseteq \mathcal{A}_{\text{W}}(E_H(w, r), r, w)$ for all $r > r'' > r'$. So no of the edges out of $E_H(w, r'')$ can use any edges which obtained weight w after r as an alternative and thus there are no costs of SimpleMST incured on edges which are not assigned to H. This proves the lemma. □

A.2 Proofs of technical properties for independent stochastic process

We show that

$$\sum_{x=1}^{n} \frac{1}{x^\alpha} \leq \frac{1}{\alpha} \ ,$$

for any $\alpha > 1$ and $n \geq 1$. Obviously we have $1/(x-1) \geq 1/[x]$ for all real numbers $x > 1$, where $[x]$ stands for the greatest integer number not larger than x. Using this property we can upper bound

$$\sum_{x=1}^{n} \frac{1}{x^\alpha} \leq \int_1^n \frac{1}{(x-1)^\alpha}\, dx \ .$$

Evaluating the definite integral we obtain

$$\int_1^n \frac{1}{(x-1)^\alpha}\, dx \leq \frac{1}{\alpha - 1} \ .$$

A.3 Proofs of technical properties for dependent stochastic process

Proof (of Lemma 3). Consider a meta-round r' and define $I(r')$ the set of edges which increase their weight in r'. For a fixed i it holds

$$\begin{aligned} \mathbf{Pr}[I(r') = i] &= \sum\nolimits_{\substack{E' \subseteq E \\ |E'| = i}} \left[\prod\nolimits_{e \in E'} p'_e \prod\nolimits_{e \in E \setminus E'} (1 - p'_e) \right] \\ &\leq m^i \left(\max_{e \in E} p'_e \right)^i =: \widetilde{\mathbf{Pr}}[I(r') = i] \end{aligned} \tag{8}$$

Equation (8) holds analogically as Eq. (2) from Section 3. For $p'_e < 1/m$ we have $\widetilde{\mathbf{Pr}}[I(r') = i] \leq \widetilde{\mathbf{Pr}}[I(r') = i']$ for $i' > i$. Since $\max_{e \in E} p'_e \leq 1/(r^{1/2}m^2)$, it holds

$$\widetilde{\mathbf{Pr}}[I(r') = 2] \leq m^2 \left(\frac{1}{r^{1/2}m^2} \right)^2 \leq \frac{1}{rm^2} \ .$$

Estimating the probability $\mathbf{Pr}[I(r') \geq 2]$ we obtain

$$\mathbf{Pr}[I(r') \geq 2] \leq m \, \widetilde{\mathbf{Pr}}[I(r') = 2] \leq \frac{1}{rm} \ ,$$

and summing up over all rounds we have

$$\mathbf{Pr}\left[\forall_{r' \in \{1,\ldots,r\}} I(r') \leq 1\right] \geq 1 - \frac{1}{m} \ .$$

□

Proof (of Lemma 4). Consider a round generated by $\mathcal{I}_{\mathcal{P}}$ in which exactly one edge weight is increased. Consider the probability that the weight of a fixed edge e is increased

$$\begin{aligned}
&\mathbf{Pr}\left[\text{weight of } e \text{ increased} \mid \text{weight of one edge increased}\right] \\
&= \frac{\mathbf{Pr}[\text{weight of } e \text{ increased} \wedge \text{weight of one edge increased}]}{\mathbf{Pr}[\text{weight of one edge increased}]} \\
&= \frac{p'_e \left(\prod_{\widehat{e} \in E \setminus \{e\}} \left(1 - p'_{\widehat{e}}\right) \right)}{\sum_{e' \in E} \left(p'_{e'} \prod_{\widehat{e} \in E \setminus \{e'\}} \left(1 - p'_{\widehat{e}}\right) \right)}
\end{aligned}$$

As in the assumption of the lemma, let $p'_e = p_e/k$. Then, for a sufficiently large k we have $p'_{\widehat{e}} = p_{\widehat{e}}/k \approx 0$. Then it holds

$$\frac{p'_e \left(\prod_{\widehat{e} \in E \setminus \{e\}} \left(1 - p'_{\widehat{e}}\right) \right)}{\sum_{e' \in E} \left(p'_{e'} \prod_{\widehat{e} \in E \setminus \{e'\}} \left(1 - p'_{\widehat{e}}\right) \right)} \approx \frac{p'_e}{\sum_{e' \in E} p'_{e'}} = \frac{p_e}{\sum_{e' \in E} p_{e'}} = p_e \ ,$$

and the lemma is proven. □