Combinatorics, Probability and Computing (2007) 16, 127–144. © 2006 Cambridge University Press
doi:10.1017/S096354830600770X First published online 14 August 2006 Printed in the United Kingdom

On an Online Spanning Tree Problem in Randomly Weighted Graphs

JAN REMY, ALEXANDER SOUZA and ANGELIKA STEGER

Institut für Theoretische Informatik, ETH Zürich, Switzerland
(e-mail: jremy@inf.ethz.ch, asouza@inf.ethz.ch, steger@inf.ethz.ch)

Received 20 October 2004; revised 7 February 2005

This paper is devoted to an online variant of the minimum spanning tree problem in randomly weighted graphs. We assume that the input graph is complete and the edge weights are uniformly distributed over $[0, 1]$. An algorithm receives the edges one by one and has to decide immediately whether to include the current edge into the spanning tree or to reject it. The corresponding edge sequence is determined by some adversary. We propose an algorithm which achieves $\mathbb{E}[\mathrm{ALG}]/\mathbb{E}[\mathrm{OPT}] = O(1)$ and $\mathbb{E}[\mathrm{ALG}/\mathrm{OPT}] = O(1)$ against a fair adaptive adversary, *i.e.*, an adversary which determines the edge order online and is fair in a sense that he does not know more about the edge weights than the algorithm. Furthermore, we prove that no online algorithm performs better than $\mathbb{E}[\mathrm{ALG}]/\mathbb{E}[\mathrm{OPT}] = \Omega(\log n)$ if the adversary knows the edge weights in advance. This lower bound is tight, since there is an algorithm which yields $\mathbb{E}[\mathrm{ALG}]/\mathbb{E}[\mathrm{OPT}] = O(\log n)$ against the strongest-imaginable adversary.

1. Introduction

Let $G = (V, E)$ be a complete graph on n vertices which is *randomly weighted*, *i.e.*, the edge weights are uniformly and independently chosen from $[0, 1]$. Let the random variable OPT denote the weight of the minimum spanning tree in G. Frieze [6] proved that

$$\lim_{n\to\infty} \mathbb{E}[\mathrm{OPT}] = \sum_{j=1}^{\infty} \frac{1}{j^3} = \zeta(3).$$

Let $N = \binom{n}{2}$ and assume that the edges are indexed by increasing weight. Frieze bounds the indices of the edges Kruskal's algorithm takes from above and below. Since the expected weight of the ith-smallest edge in G is $i/(N+1)$, the value given above follows by linearity of expectation. Note that this argument relies heavily on the fact that the whole graph is known.

Consider the following online version of the minimum spanning tree problem, described as a game between an algorithm and an adversary. Assume that some adversary $\mathscr{ADV}$ administrates the set E and shows it edge by edge to an online algorithm $\mathscr{A}$. Without

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

any precise knowledge about the weights of unseen edges, the algorithm has to decide immediately whether or not to include the edge in the spanning tree. Let ALG denote the weight of the spanning tree generated by the algorithm. Of course, the goal of $\mathscr{A}$ is to minimize ALG, which means we seek an online algorithm that competes well with the offline optimum. Analogously to the deterministic definition (*cf.* [3]) of c-competitiveness, an online algorithm $\mathscr{A}$ is said to be *c-competitive* against some adversary $\mathscr{A}\mathscr{D}\mathscr{V}$ if there is a constant α such that

$$\mathbb{E}[\mathrm{ALG} - c \cdot \mathrm{OPT}] \leqslant \alpha.$$

Note that OPT is a random variable that only depends on the (randomly weighted) input graph. On the other hand, ALG depends on the input graph, the adversary and the algorithm itself. Since OPT is constant in expectation, it is reasonable to require that $\alpha \leqslant 0$, *i.e.*, $\mathscr{A}$ is *strictly c-competitive*. The *strict competitiveness coefficient* of $\mathscr{A}$ against $\mathscr{A}\mathscr{D}\mathscr{V}$ is defined as

$$C_{\mathscr{A}}^{\mathscr{A}\mathscr{D}\mathscr{V}} := \frac{\mathbb{E}[\mathrm{ALG}]}{\mathbb{E}[\mathrm{OPT}]}.$$

It remains to be specified how powerful the adversary is. In the standard literature [3, 16], two basic types of adversary models are used to analyse online algorithms in a probabilistic context. *Oblivious adversaries* determine the edge order in advance, as opposed to *adaptive adversaries* that choose the next edge based on the algorithm's actions so far. An adversary $\mathscr{A}\mathscr{D}\mathscr{V}$ is said to be *fair* if he discovers the weight of the edges at the same time as the algorithm. In contrast, $\mathscr{A}\mathscr{D}\mathscr{V}$ is *unfair* if he knows the edge weights in advance and uses this information to determine the edge order. We thus consider the following four adversaries: $\mathscr{F}\mathscr{O}$ (fair, oblivious), $\mathscr{F}\mathscr{A}$ (fair, adaptive), $\mathscr{U}\mathscr{O}$ (unfair, oblivious) and $\mathscr{U}\mathscr{A}$ (unfair, adaptive). It follows by inclusion that for every algorithm $\mathscr{A}$,

$$C_{\mathscr{A}}^{\mathscr{F}\mathscr{O}} \leqslant C_{\mathscr{A}}^{\mathscr{F}\mathscr{A}} \quad \text{and} \quad C_{\mathscr{A}}^{\mathscr{U}\mathscr{O}} \leqslant C_{\mathscr{A}}^{\mathscr{U}\mathscr{A}}. \tag{1.1}$$

In this paper we give an algorithm which is strictly $O(1)$-competitive against $\mathscr{F}\mathscr{A}$ (Theorem 2.1). In addition, we show that there is an algorithm which is strictly $O(\log n)$-competitive against $\mathscr{U}\mathscr{A}$ (Theorem 3.2). Furthermore, we demonstrate that for every algorithm $\mathscr{A}$, we have lower bounds $C_{\mathscr{A}}^{\mathscr{U}\mathscr{O}} = \Omega(\log n)$ (Theorem 3.1) and $C_{\mathscr{A}}^{\mathscr{F}\mathscr{O}} \geqslant c \cdot \zeta(3)$ (Theorem 2.14) for a fixed constant $c > 1$. Then, the relation given in (1.1) ensures that we have found for each of the four adversaries the best (in order of magnitude) possible online algorithm.

Related work

The problem of finding a minimum spanning tree in graphs is one of the classical problems in combinatorial optimization and thus well studied. For recent results on fast algorithms for this problem the reader is referred to [4] or [17] and references therein. The concept of the randomly weighted graph is heavily related to the theory of random graphs. For a comprehensive introduction see [2] and [13]. As mentioned earlier, Frieze [6] proved that the expectation of OPT is $\zeta(3)$. Further work considered the distribution of OPT [12] and its expectation on bipartite [7] and k-regular graphs [1, 9].

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

Also, the algorithmic aspect of random graphs and randomly weighted graphs received much attention. Some references can be found in [8]. Karp and Tarjan [14] have proposed an algorithm for random weighted graphs that requires $O(n+m)$ expected time. On the other hand, McDiarmid, Johnson and Stone [15] analysed the behaviour of Prim's algorithm. Similar results have been achieved for other graph problems. Recently, Hagerup [11] proposed a simple, $O(n+m)$ expected time algorithm that solves the single-source-shortest-path problem in randomly weighted but not necessarily complete graphs.

Outline of the paper

This paper is organized as follows. In Section 2 we prove upper and lower bounds for fair adversaries. Section 3 provides results on unfair adversaries. Finally, in Section 4 we will investigate the *expected competitive ratio*, *i.e.*, $\mathbb{E}[\rho]$ for $\rho = \mathrm{ALG}/\mathrm{OPT}$.

2. Competing with fair adversaries

2.1. An $O(1)$-competitive algorithm

In this section we present an $O(1)$-competitive online algorithm that competes with adversary $\mathscr{F}\mathscr{A}$ and seeks to minimize the total weight of the spanning tree constructed. More specifically, we prove the following result.

Theorem 2.1. *There exists an algorithm $\mathscr{A}$ achieving $C_{\mathscr{A}}^{\mathscr{F}\mathscr{A}} = O(1)$.*

In order to prove Theorem 2.1 we will actually prove something slightly stronger. Suppose we are given an online algorithm CONNECT that receives the edges of G one by one, decides whether to accept or to reject an edge before seeing the next, and guarantees to construct a *connected* subgraph of G. Then we can easily deduce an algorithm that ensures the construction of a spanning tree by including each edge e accepted by CONNECT into a spanning forest T of G, *unless* e closes a cycle in T. We have thus reduced the problem of designing an algorithm for finding a spanning tree with expected weight $O(1)$ to the problem of finding a connected subgraph of expected weight $O(1)$. Hence we will consider the latter problem only. Algorithm CONNECT (Algorithm 1) constructs a connected subgraph C of G and is analysed in Theorem 2.4 below.

Let us now explain the main ideas behind CONNECT. Let e be an edge that is exposed to the algorithm. There are exactly two circumstances under which e is *accepted*.

(1) *Regular acceptance*. If e satisfies the *weight criterion* for any one or both of its end-vertices as explained below, then e is accepted and called a *regular edge*.
(2) *Emergency acceptance*. If rejecting e *disconnects* the graph induced by the accepted and unseen edges, then e is accepted (regardless of its weight) and called an *emergency edge*.

If none of the two conditions is satisfied, then e is *rejected*.

Regular acceptance. Let t denote the *time* when exactly t edges $e_1, \ldots, e_t$ are exposed to CONNECT. For each vertex v, we define a function p stated below which takes values in the set $P = \{1, 2, \ldots, \lceil \log(\frac{n}{8d}) \rceil\}$, where n is the number of vertices, d a positive integer

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

constant (to be specified later), and log denotes the logarithm base 2. We assume $n > 8d$ throughout. Formally, the function p depends on v and the sequence $e_1, \dots, e_t$ of edges exposed until time t, but for simplicity of notation, we use $p(v,t)$ as a shorthand. Suppose that $e = \{u, v\}$ is shown at time t. We define that vertex v *accepts* e if

$$\omega(e) \leqslant \frac{8d}{n} 2^{p(v,t)},$$

otherwise, v *rejects* e. An edge $e = \{u, v\}$ is *accepted regularly* by CONNECT if u *or* v accept the edge.

We call an edge e a v-*edge*, if e is incident with the vertex v. We divide the sequence of v-edges of a vertex v shown by the adversary, into d subsequences of $\lfloor \frac{n}{d} \rfloor$ consecutive edges. We call the kth of these sequences the *degree-phase* k for vertex v. In each degree-phase, v must accept at least one v-edge. Each degree-phase is subdivided into *edge-phases*, where the ith edge-phase consists of $\lceil \frac{n}{4d} 2^{-i} \rceil$ consecutive v-edges, for $i = 1, \dots, \lceil \log(\frac{n}{8d}) \rceil$. If there are edges left after edge-phase $\lceil \log(\frac{n}{8d}) \rceil$, these are called the *remainder* of the degree-phase. Observation 2.2 below shows that this definition is indeed consistent.

Let us now define the function $p(v,t)$, which guarantees that each vertex v will have accepted at least d edges regularly at termination of CONNECT. If at time t the vertex v has *not* yet accepted an edge in its current degree-phase, then $p(v,t)$ takes the number of the current edge-phase of v. Otherwise, *i.e.*, if one edge was accepted or the vertex is in the remainder of the degree-phase, then we set $p(v,t) = 1$ until the next degree-phase begins.

The function $p(v,t)$ ensures that, by the end of a degree-phase, at least one edge will have been accepted regularly, see Lemma 2.3. The *first* edge to be accepted by a vertex in any degree-phase is called the *compulsory edge* of that degree-phase; the *other* edges accepted are called *bonus edges*. In addition, in each degree-phase, after having accepted one (compulsory) edge, the vertex will accept only those (bonus) edges that have weight at most $\frac{16d}{n}$ by setting $p(v,t) = 1$.

Emergency acceptance. Let e be the edge shown at time t and consider the graph $C = (V, A \cup U)$ induced by the set A of accepted edges and the set U of unseen edges *at time* t. If $C - e$ is disconnected, then e must be accepted in order to ensure the construction of a connected subgraph. In that case, e is called an *emergency edge* and accepted regardless of its weight.

Observation 2.2. *The choice of the constants and the assumption $n > 8d$ imply that edge-phases 1 to $\lceil \log(\frac{n}{8d}) \rceil$ of any degree-phase use in total less then $\lfloor \frac{n}{d} \rfloor$ edges and that the above definition is therefore consistent.*

Proof. The assumption $n > 8d$ implies $\log(\frac{n}{8d}) > 0$ and hence $\lceil \log(\frac{n}{8d}) \rceil \geqslant 1$. Consider an edge-phase i such that $1 \leqslant i \leqslant \lceil \log(\frac{n}{8d}) \rceil$. By $n > 8d$ and $i \leqslant \lceil \log(\frac{n}{8d}) \rceil \leqslant \log(\frac{n}{8d}) + 1$, we find $\lceil \frac{n}{4d} 2^{-i} \rceil \leqslant \frac{n}{2d} 2^{-i}$ and $\lfloor \frac{n}{d} \rfloor \geqslant \frac{n}{2d}$. By definition, such an edge-phase has length $\lceil \frac{n}{4d} 2^{-i} \rceil \geqslant 1$. Hence

$$\sum_{i=1}^{\lceil \log(\frac{n}{8d}) \rceil} \left\lceil \frac{n}{4d} 2^{-i} \right\rceil \leqslant \sum_{i=1}^{\lceil \log(\frac{n}{8d}) \rceil} \frac{n}{2d} 2^{-i} \leqslant \frac{n}{2d} \sum_{i=1}^{\infty} 2^{-i} \leqslant \frac{n}{2d} \leqslant \left\lfloor \frac{n}{d} \right\rfloor.$$

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

Algorithm 1 CONNECT

Step 1. Set $t = 0$, $A = \emptyset$, $U = \binom{V}{2}$, and $d = 10$.
Step 2. Let $C = (V, A \cup U)$. If $U = \emptyset$ output C and terminate. Otherwise set $t = t + 1$, get the next edge $e = \{u, v\}$, and set $U = U \backslash \{e\}$.
Step 3. If

$$\omega(e) \leqslant \frac{8d}{n} 2^{p(u,t)} \quad \text{or} \quad \omega(e) \leqslant \frac{8d}{n} 2^{p(v,t)}$$

goto Step 5.
Step 4. If $C - e$ is not connected goto Step 5, otherwise goto Step 2.
Step 5. Let $A = A \cup \{e\}$ and goto Step 2.

Therefore, the $\lfloor \frac{n}{d} \rfloor$ edges available for each degree-phase suffice for $\lceil \log(\frac{n}{8d}) \rceil$ edge-phases and each such phase contains at least one edge. □

Lemma 2.3. *For all $n > 8d$, algorithm* CONNECT *has the property that every vertex $v \in V$ accepts at least one edge regularly in each of its d degree-phases.*

Proof. In edge-phase number $\lceil \log(\frac{n}{8d}) \rceil$, an edge is accepted if $\omega(e) \leqslant \frac{8d}{n} 2^{\lceil \log(\frac{n}{8d}) \rceil}$. As $\frac{8d}{n} 2^{\lceil \log(\frac{n}{8d}) \rceil} \geqslant \frac{8d}{n} 2^{\log(\frac{n}{8d})} = 1$ and edge-weights are bounded by 1, we conclude that at the latest in this edge-phase an edge will be accepted regularly. □

Observe that Lemma 2.3 implies that the minimum degree of the graph constructed is d.

Theorem 2.4. *Against a fair adaptive adversary $\mathscr{FA}$, algorithm* CONNECT *outputs a connected subgraph C of G with $\mathbb{E}[\omega(C)] = O(1)$.*

We break the proof of Theorem 2.4 into two lemmas. Lemma 2.7 bounds the expected weight of the regular edges, while Lemma 2.8 bounds the expected weight of the emergency edges.

The following notation and definitions are used in the sequel. For a vertex v we denote by $E(v)$ the set of its incident edges, *i.e.*, the set of all v-edges. For any $W \subseteq V$ let $E(W) = \{\{v, w\} \in E : v, w \in W\}$ and $e(W) = |E(W)|$. For a set W, any pair $(X, \overline{X})$ with $X \subset W$, $\overline{X} = W \backslash X$ is called a *cut*. Let $E(X, \overline{X}) = \{\{v, \overline{v}\} \in E : v \in X, \overline{v} \in \overline{X}\}$ and $e(X, \overline{X}) = |E(X, \overline{X})|$ denote the number of *cut edges*. Furthermore, we define some random variables.

Indicator variables.

- $A(v, e)$ indicates whether vertex v accepts edge e regularly.
- $S(v, e, k)$ indicates whether edge e is shown to vertex v in degree-phase k.
- $D(W)$ indicates whether in a set $W \subseteq V$ *each* vertex $v \in W$ has accepted exactly d *compulsory* edges in $E(W)$ after the execution of CONNECT.

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

General random variables.

- $A(v,k,i)$ denotes the number of edges accepted regularly by v in edge-phase i of degree-phase k.
- For a vertex v and a subset $W \subseteq E(v)$ of its incident edges, let

$$A(v,W,k) = \sum_{e \in W} A(v,e)S(v,e,k)$$

denote the number of v-edges in W accepted regularly by v in degree-phase k, and let

$$A(v,W) = \sum_{k=1}^{d} A(v,W,k)$$

denote the total number of v-edges in W that are accepted regularly by v in some degree-phase.

- $P(v,k)$ denotes the value of the function $p(v,t)$ at the time when the compulsory edge of v of degree-phase k is accepted.
- $P(v,e)$ denotes the number of the *edge-phase* of v when the edge e is exposed.

The following observations follow immediately from the definition and the fact that edge-weights $\omega(e)$ are independent random variables.

Observation 2.5. *For all vertices $v \in V$ we have that the probability that* no *edge is accepted regularly in edge-phase i of degree-phase k is*

$$\Pr\{A(v,k,i) = 0 \mid A(v,k,1) = 0, \dots, A(v,k,i-1) = 0\} \leqslant \left(1 - \frac{8d}{n}2^i\right)^{\frac{n}{4d}2^{-i}} \leqslant \mathrm{e}^{-2}.$$

Observation 2.6. *For the random variable $P(v,k)$ we have $\Pr\{P(v,k) = i\} \leqslant \mathrm{e}^{-2(i-1)}$ by Observation 2.5.*

Lemma 2.7. *For a vertex v let $A(v) = \{e \in E(v) : A(v,e) = 1\}$ denote the set of regularly accepted v-edges during the execution of* Connect. *Then*

$$\mathbb{E}\left[\sum_{e \in A(v)} \omega(e)\right] = O\left(\frac{1}{n}\right).$$

Proof. Let W_k denote the weight of the compulsory edge accepted by v in degree-phase k. Observe that if the compulsory edge is accepted in edge-phase i, then all edges in the previous $i-1$ edge-phases must have been rejected. Thus, by Observation 2.6 and $\mathbb{E}[W_k \mid P(v,k) = i] \leqslant \frac{8d}{n}2^i$ we find that

$$\mathbb{E}[W_k] = \sum_{i=1}^{\lceil \log(\frac{n}{8d}) \rceil} \mathbb{E}[W_k \mid P(v,k) = i] \Pr\{P(v,k) = i\} \leqslant \sum_{i=1}^{\infty} \frac{8d}{n}2^i \mathrm{e}^{-2(i-1)} \leqslant \frac{22d}{n}.$$

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

A bonus edge is accepted if its weight is at most $\frac{16d}{n}$, which happens with probability $\frac{16d}{n}$. Thus the total contribution of all bonus edges accepted by v is at most $\frac{16^2d^2}{n}$. Hence it follows that

$$\mathbb{E}\left[\sum_{e\in A(v)}\omega(e)\right] \leqslant \sum_{k=1}^{d}\mathbb{E}[W_k] + \frac{16^2d^2}{n} \leqslant \frac{22d^2}{n} + \frac{16^2d^2}{n} = O\left(\frac{1}{n}\right),$$

which completes the proof, assuming that d is a constant. □

Lemma 2.8. *Let F denote the number of emergency edges that were accepted during the execution of* Connect, *then* $\Pr\{F \geqslant 1\} = O\left(n^{-\frac{3}{2}}\right)$ *and* $\mathbb{E}[F] = o(1)$.

Let us first sketch the idea behind the proof of Lemma 2.8. If an emergency edge has to be accepted during the execution of the algorithm, then the following bad event must have occurred. There must exist a set $W \subseteq V$ such that *all* (bonus) edges over the cut $(W, \overline{W})$ were rejected and *each* vertex in W has accepted its (compulsory) edges within W. We intend to show that if the set W is 'large', then the probability that all edges over the cut $(W, \overline{W})$ are rejected is small, and if W is 'small', then the probability for such a dense subgraph is small. The actual proof of Lemma 2.8 is subdivided into several technical claims which we now state.

Claim 2.9. *For any edge $e = \{v, w\}$ and any degree-phase k we have that*

$$\Pr\{A(v,e) = 1 \mid S(v,e,k) = 1\} \leqslant \frac{38d}{n}.$$

Proof. Suppose that vertex v is in degree-phase k when e is exposed, *i.e.*, $S(v,e,k) = 1$. Observe that when edge e is exposed it is either a possible bonus edge (if vertex v has already accepted its compulsory edge for degree-phase k) or it is a possible compulsory edge (if all edges in the previous edge-phases have been rejected). The indicator variable $B(e,v,k)$ takes the value 1 if the first event occurs and 0 otherwise, *i.e.*, if the second event occurs.

Suppose $B(e,v,k) = 1$, *i.e.*, v has already accepted its compulsory edge of degree-phase k. Then, at time t when e is shown, we have $p(v,t) = 1$ and the probability of accepting e is hence $\frac{16d}{n}$. Otherwise, we have $B(e,v,k) = 0$ for the current edge-phase $P(v,e) = i$ and all edges in the $i-1$ previous edge-phases must have been rejected. Therefore, by Observation 2.6,

$$\Pr\{B(e,v,k) = 0 \mid P(v,e) = i\} \leqslant \Pr\{P(v,k) \geqslant i\} \leqslant \mathrm{e}^{-2(i-1)}.$$

Moreover, if $B(e,v,k) = 0$ and $p(v,t) = P(v,e) = i$ then the edge is accepted with probability $\frac{8d}{n}2^i$. Observe that in this argument we heavily used that the adversary $\mathscr{FA}$ is fair, *i.e.*, does not know the edge-weights in advance.

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

Further note that the events $P(v,e)=i$ for $i=1,\dots,\lceil\log(\frac{n}{8d})\rceil$ and $B(e,v,k)=b$ with $b\in\{0,1\}$ induce a partition of the probability space. Now we calculate

$$\begin{aligned}
&\Pr\{A(v,e)=1 \mid S(v,e,k)=1\}\\
&\quad=\sum_{b\in\{0,1\}}\sum_{i=1}^{\lceil\log(\frac{n}{8d})\rceil}\Pr\{A(v,e)=1 \mid S(v,e,k)=1, P(v,e)=i, B(e,v,k)=b\}\\
&\qquad\qquad\cdot\Pr\{P(v,e)=i, B(e,v,k)=b\}\\
&\quad\leqslant\frac{16d}{n}\sum_{i=1}^{\lceil\log(\frac{n}{8d})\rceil}\Pr\{P(v,e)=i, B(e,v,k)=1\}\\
&\qquad\qquad+\sum_{i=1}^{\lceil\log(\frac{n}{8d})\rceil}\frac{8d}{n}2^i\Pr\{P(v,e)=i, B(e,v,k)=0\}\\
&\quad\leqslant\frac{16d}{n}+\sum_{i=1}^{\lceil\log(\frac{n}{8d})\rceil}\frac{8d}{n}2^i\Pr\{B(e,v,k)=0 \mid P(v,e)=i\}\Pr\{P(v,e)=i\}\\
&\quad\leqslant\frac{16d}{n}+\sum_{i=1}^{\infty}\frac{8d}{n}2^i\mathrm{e}^{-2(i-1)}\leqslant\frac{38d}{n}
\end{aligned}$$

and the proof is complete. □

The essence of Claim 2.9 is that no matter when the adversary exhibits e, the probability of acceptance is at most $O\left(\frac{1}{n}\right)$. Henceforth we will use $p:=\frac{38d}{n}$ as a shorthand.

Claim 2.10. *Let v be a vertex, let k be some degree-phase, and let $W\subseteq E(v)$ be a fixed set of v-edges; then* $\Pr\{A(v,W,k)\geqslant 1\}\leqslant p|W|$.

Proof. The key idea of the proof is that we condition on the set of v-edges exposed in the degree-phases $1,\dots,k-1$. To formalize this we show that for each set $U\subseteq E(v)$

$$\Pr\{A(v,W,k)\geqslant 1 \mid U \text{ exposed}\}\leqslant p|W\setminus U| \tag{2.1}$$

holds, where we wrote 'U exposed' as shorthand for 'U is the set of v-edges exposed in the first $k-1$ degree-phases of vertex v'.

In order to prove (2.1) we recall that $A(v,W,k)=\sum_{e\in W}A(v,e)S(v,e,k)$ and thus, by Claim 2.9 and Boole's inequality,

$$\begin{aligned}
&\Pr\{A(v,W,k)\geqslant 1 \mid U \text{ exposed}\}\\
&\quad\leqslant\sum_{e\in W\setminus U}\Pr\{A(v,e)=1 \mid S(v,e,k)=1\}\Pr\{S(v,e,k)=1\}\leqslant p|W\setminus U|.
\end{aligned}$$

Now (2.1) yields

$$\begin{aligned}
\Pr\{A(v,W,k)\geqslant 1\}&=\sum_{U\subseteq E(v)}\Pr\{A(v,W,k)\geqslant 1 \mid U \text{ exposed}\}\Pr\{U \text{ exposed}\}\\
&\leqslant\sum_{U\subseteq E(v)}p|W\setminus U|\Pr\{U \text{ exposed}\}\leqslant p|W|\sum_{U\subseteq E(v)}\Pr\{U \text{ exposed}\}=p|W|,
\end{aligned}$$

which completes the proof. □

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

Let $G = (W, A)$ be a directed graph. For any set $U \subseteq W$ define the *in-neighbourhood of v in U* by $N^-(v, U) = \{w \in U : (w, v) \in A\}$ and the respective *in-degree* by $\deg^-(v, U) = |N(v, U)|$. Denote by $N^-(v) = N^-(v, W)$ and $\deg^-(v) = \deg^-(v, W)$ the *in-neighbourhood* and *in-degree*, respectively.

Claim 2.11. *Let $G = (W, A)$ be a directed graph on $|W| = s$ vertices such that each vertex has in-degree exactly d, where d is even. Then there is a cut $(X, \overline{X})$ in W with $|X| \geqslant \frac{s}{4}$ such that all vertices in X each have at least $\frac{d}{2}$ in-neighbours in $\overline{X}$.*

Proof. Consider the following experiment. Each vertex is placed in a set Y or in $\overline{Y} = W \setminus Y$ with respective probability $\frac{1}{2}$, independently of all other vertices. Define $X = \{v \in Y : \deg^-(v, \overline{Y}) \geqslant \frac{d}{2}\}$, and $\overline{X} = W \setminus X$. By independence of vertex placement and as d is even, it holds that

$$\Pr\left\{\deg^-(v, \overline{Y}) \geqslant \frac{d}{2}\right\} = \frac{1}{2}.$$

Therefore, the expected number of vertices that are in Y and that have at least $\frac{d}{2}$ in-neighbours in $\overline{Y}$ is $\frac{s}{4}$. Now it follows from the probabilistic method that there exists a cut $(X, \overline{X})$ in W with $|X| \geqslant \frac{s}{4}$, such that all vertices in X each have at least $\frac{d}{2}$ in-neighbours in $\overline{X}$. □

The following claim bounds the probability that each vertex in a certain set $W \subseteq V$ has accepted its (compulsory) edges within W. From now on we omit $\lceil\ \rceil$ for the sake of simplicity.

Claim 2.12. *Let $W \subseteq V$ with $|W| = s$, then $\Pr\{D(W) = 1\} \leqslant \left(4\mathrm{e}(3ps)^{\frac{d}{2}}\right)^{\frac{s}{4}}$.*

Proof. Consider the graph induced by compulsory edges after the execution of Connect. Recall that $D(W) = 1$ holds for a set W if each vertex in W has accepted exactly d compulsory edges in $E(W)$. Let $(X, \overline{X})$ be a cut in W with $|X| = \frac{s}{4}$. Then $C(X, \overline{X})$ indicates whether all vertices in X have each accepted at least $\frac{d}{2}$ neighbours in $\overline{X}$. By Claim 2.11 there exists a cut $(X, \overline{X})$ in W such that $C(X, \overline{X}) = 1$, as we may orient each compulsory edge towards the vertex which accepted it, doubling edges that were accepted by two vertices. Now Boole's inequality yields

$$\begin{aligned}\Pr\{D(W) = 1\} &\leqslant \Pr\left\{\sum_{(X,\overline{X}),|X|=\frac{s}{4}} C(X, \overline{X}) \geqslant 1\right\} \\ &\leqslant \sum_{(X,\overline{X}),|X|=\frac{s}{4}} \Pr\{C(X, \overline{X}) = 1\} = \binom{s}{\frac{s}{4}} \Pr\{C(Z, \overline{Z}) = 1\},\end{aligned}$$

where $(Z, \overline{Z})$ is any cut in W with $|Z| = \frac{s}{4}$. For any vertex $v \in Z$ use $\overline{Z}(v) = E(\{v\}, \overline{Z})$ as a shorthand. For any degree-phase k Claim 2.10 yields that $\Pr\{A(v, \overline{Z}(v), k) = 1\} \leqslant \frac{3}{4}ps$. The event $A(v, \overline{Z}(v), k) = 1$ must have happened in at least $\frac{d}{2}$ degree-phases and thus Boole's

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

inequality yields

$$\Pr\left\{A(v,\overline{Z}(v)) \geqslant \frac{d}{2}\right\} = \Pr\left\{\sum_{k=1}^{d} A(v,\overline{Z}(v),k) \geqslant \frac{d}{2}\right\} \leqslant \binom{d}{\frac{d}{2}}\left(\frac{3}{4}ps\right)^{\frac{d}{2}}.$$

As this holds for all vertices in Z, the estimate $\binom{d}{\frac{d}{2}} \leqslant 4^{\frac{d}{2}}$ yields

$$\Pr\{C(Z,\overline{Z}) = 1\} \leqslant \left(\binom{d}{\frac{d}{2}}\left(\frac{3}{4}ps\right)^{\frac{d}{2}}\right)^{\frac{s}{4}} \leqslant \left((3ps)^{\frac{d}{2}}\right)^{\frac{s}{4}}.$$

The standard estimate $\binom{n}{k} \leqslant \left(\frac{ne}{k}\right)^k$ and hence $\binom{s}{\frac{s}{4}} \leqslant (4e)^{\frac{s}{4}}$ yields the claimed bound. □

Now we state Claim 2.13, which bounds the probability that for a set $W \subseteq V$ all (bonus) edges $E(W,\overline{W})$ are rejected.

Claim 2.13. *Let $W \subseteq V$, $|W| = s$ and let $(W,\overline{W})$ be a cut, then*

$$\Pr\{e(W,\overline{W}) = 0\} \leqslant \exp\left(-\frac{16ds(n-s)}{n}\right),$$

where with abuse of notation $e(W,\overline{W})$ denotes the random variable that counts the number of edges in $(W,\overline{W})$ that were accepted regularly.

Proof. By the weight criterion, *each* edge is accepted with probability at least $\frac{16d}{n}$, and thus rejected with probability at most $1 - \frac{16d}{n}$. Hence the probability that all $s(n-s)$ edges over the cut $(W,\overline{W})$ are rejected is at most

$$\Pr\{e(W,\overline{W}) = 0\} \leqslant \left(1 - \frac{16d}{n}\right)^{s(n-s)} \leqslant \exp\left(-\frac{16ds(n-s)}{n}\right)$$

as claimed. □

Proof of Lemma 2.8. Let $e = \{v,w\}$ be an arbitrary edge. Suppose that e is an emergency edge, and consider the graph induced by *regular* edges after execution of Connect. Then there is a cut $(W,\overline{W})$ such that $|W| \leqslant \frac{n}{2}$, all edges over the cut were rejected, and $v \in W$, $w \in \overline{W}$. It is hence necessary that $D(W) = 1$ and that $e(W,\overline{W}) = 0$, otherwise e cannot be an emergency edge. Therefore, the probability that there exists an emergency edge is bounded by the probability that such a set W exists. Hence by Boole's inequality we have that

$$\begin{aligned}\Pr\{F \geqslant 1\} &\leqslant \Pr\left\{\exists W \subseteq V : |W| \leqslant \frac{n}{2}, D(W) = 1, e(W,\overline{W}) = 0\right\} \\ &\leqslant \sum_{W \subseteq V, |W| \leqslant \frac{n}{2}} \Pr\{e(W,\overline{W}) = 0 \wedge D(W) = 1\}.\end{aligned}$$

Now we distinguish two cases. If $|W| \leqslant \frac{n}{10^{20}d}$ we apply Claim 2.12, otherwise we apply Claim 2.13. Notice also that K_{d+1} is the smallest-possible graph such that $D(W) = 1$.

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

Hence we may assume $|W| \geqslant d+1$. With $p = \frac{38d}{n}$ we obtain

$$\begin{aligned}\Pr\{F \geqslant 1\} &\leqslant \sum_{W\subseteq V, d+1\leqslant|W|\leqslant\frac{n}{10^{20}d}} \Pr\{D(W)=1\} + \sum_{W\subseteq V, \frac{n}{10^{20}d}\leqslant|W|\leqslant\frac{n}{2}} \Pr\{e(W,\overline{W})=0\} \\ &\leqslant \underbrace{\sum_{d+1\leqslant s\leqslant\frac{n}{10^{20}d}} \binom{n}{s}\left(4\mathrm{e}\left(114\frac{sd}{n}\right)^{\frac{d}{2}}\right)^{\frac{s}{4}}}_{A} + \underbrace{\sum_{\frac{n}{10^{20}d}\leqslant s\leqslant\frac{n}{2}} \binom{n}{s}\exp\left(-\frac{16ds(n-s)}{n}\right)}_{B}.\end{aligned}$$

We consider both sums A and B separately. Firstly, we find that

$$A = \sum_{d+1\leqslant s\leqslant\frac{n}{10^{20}d}} \binom{n}{s}\left(4\mathrm{e}\left(114\frac{sd}{n}\right)^{\frac{d}{2}}\right)^{\frac{s}{4}} \leqslant n \max_{d+1\leqslant s\leqslant\frac{n}{10^{20}d}} \left(\frac{n\mathrm{e}}{s}(4\mathrm{e})^{\frac{1}{4}}\left(114\frac{sd}{n}\right)^{\frac{d}{8}}\right)^{s}$$

Assuming $d \geqslant 10$ and comparing two consecutive values for s yields that the function is monotone decreasing in the interval $d+1 \leqslant s \leqslant \frac{n}{10^{20}d}$. If we actually set $d = 10$ we obtain

$$A \leqslant n\left(114d\mathrm{e}(4\mathrm{e})^{\frac{1}{4}}\left(114\frac{(d+1)d}{n}\right)^{\frac{d}{8}-1}\right)^{d+1} = O\left(n^{-\frac{3}{2}}\right).$$

Secondly, we can deduce that

$$\begin{aligned}B &= \sum_{\frac{n}{10^{20}d}\leqslant s\leqslant\frac{n}{2}} \binom{n}{s}\exp\left(-\frac{16ds(n-s)}{n}\right) \leqslant n \max_{\frac{n}{10^{20}d}\leqslant s\leqslant\frac{n}{2}} \left(\frac{\mathrm{e}n}{s}\right)^{s}\exp\left(-\frac{16ds(n-s)}{n}\right) \\ &\leqslant n \max_{\frac{2}{d}\leqslant c\leqslant 10^{20}d} \exp\left(\frac{16}{c}n\left(-1+\frac{1}{cd}+\frac{1}{16d}\ln(\mathrm{e}cd)\right)\right).\end{aligned}$$

It is easily verified that $\frac{1}{cd} + \frac{1}{16d}\ln(\mathrm{e}cd) \leqslant \frac{1}{2}$ holds for $d = 10$ and $c \in [\frac{2}{d}, 10^{20}d]$, from which we find $B = \exp(-\Theta(n))$.

As both sums are $O\left(n^{-\frac{3}{2}}\right)$, we have $\mathbb{E}[F] = \sum_{i=0}^{n-1} i\Pr\{F=i\} \leqslant n\Pr\{F \geqslant 1\} = o(1)$, which completes the proof. □

We remark that no serious attempt for finding best-possible constants for d and the weight criterion was made.

Proof of Theorem 2.4. First we prove that C is connected. For the sake of contradiction suppose that C is not connected. Then there exists a cut $(W, \overline{W})$ with $e(W, \overline{W}) = 0$ in C. Let t be the time when the last edge e over $(W, \overline{W})$ is rejected by CONNECT. At time t all accepted edges A and all unseen edges U are within W or $\overline{W}$. Thus, at that time the graph $C = (V, A \cup U)$ is disconnected. Therefore e is an emergency edge, contradicting the fact that it is rejected.

For any vertex v, let $A(v)$ denote the set of edges that were accepted regularly by v and let F denote the number of emergency edges. As the weight of each (emergency) edge is

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

bounded by 1, we have

$$\mathbb{E}[\omega(C)] \leqslant \sum_{v \in V} \mathbb{E}\left[\sum_{e \in A(v)} \omega(e)\right] + \mathbb{E}[F].$$

Application of Lemmas 2.7 and 2.8 completes the proof. □

2.2. A lower bound

$\mathscr{F}\mathscr{O}$ is the weakest of the four adversaries considered in this paper. By Theorem 2.1 we know that there is an algorithm which is strictly $O(1)$-competitive against $\mathscr{F}\mathscr{O}$. However, it is still possible that there is an algorithm $\mathscr{A}$ which is strictly $(1+\varepsilon)$-competitive for arbitrary $\varepsilon > 0$ or (which is even weaker) $\varepsilon = o(1)$. This is not the case.

Theorem 2.14. *There is a $c > 1$ such that $C_{\mathscr{A}}^{\mathscr{F}\mathscr{O}} \geqslant c$ for every online algorithm $\mathscr{A}$.*

Assume that we wish to select online from a set $\{X_1, \dots, X_k\}$ of independent random variables, uniformly distributed in $[0,1]$, exactly one value of minimum weight. Let W_k denote the expected objective of the best-possible online algorithm $\mathscr{B}$ for this problem. If $k = 1$ then W_1 is trivially $1/2$. Otherwise, if $k > 1$ we assume that $\mathscr{B}$ receives the value of X_1. In the optimal strategy, the algorithm will accept if X_1 is smaller than W_{k-1}, since W_{k-1} is by definition the value which can be achieved by the optimal algorithm. The proof of Theorem 2.14 heavily relies on the following result.

Lemma 2.15. *W_k is monotone decreasing and satisfies*

$$\lim_{k \to \infty} k \cdot W_k = 2.$$

Proof. By $W_1 = 1/2$ and since the optimal algorithm accepts X_1 if and only if $X_1 \leqslant W_{k-1}$, we obtain the following recursion:

$$\begin{aligned} W_k &= \mathbb{E}[X_1 \mid X_1 \leqslant W_{k-1}] \Pr\{X_1 \leqslant W_{k-1}\} + \Pr\{X_1 > W_{k-1}\} W_{k-1} \\ &= \frac{W_{k-1}}{2} W_{k-1} + (1 - W_{k-1}) W_{k-1} = W_{k-1} - \frac{W_{k-1}^2}{2}. \end{aligned}$$

Let $g_k = k \cdot W_k$. By series expansion, it follows that $g_k \to 2$ as $k \to \infty$. □

Proof of Theorem 2.14. Assume $\mathscr{F}\mathscr{O}$ fixes some order $e_1, e_2 \dots, e_N$ on the edges. Let $\mathscr{A}^*$ be the best-possible online algorithm for the minimum spanning tree problem and adversary $\mathscr{F}\mathscr{O}$. Furthermore, let $T^* = \mathrm{sol}(\mathscr{A}^*)$ denote the spanning tree which is returned by $\mathscr{A}^*$. Let $\hat{T}$ be obtained from T^* by keeping for every vertex only the edge which was accepted first. For $u \in V$ we denote this edge by a_u. If $a_u = a_v = \{u, v\}$ for two vertices $u, v \in V$, we simply double $\{u, v\}$. We define

$$\omega(\hat{T}) := \frac{1}{2} \sum_{u \in V} \omega(a_u). \tag{2.2}$$

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

On the other hand, assume we use algorithm $\mathscr{B}$ to choose online and independently exactly one edge b_u for each vertex $u \in V$. Let $\omega(\mathscr{B}) := \frac{1}{2}\sum_{u\in V}\omega(b_u)$. By Lemma 2.15, we obtain

$$\mathbb{E}[\omega(\mathscr{B})] = \frac{1}{2}\sum_{u\in V}\mathbb{E}[\omega(b_u)] = \frac{1}{2}\sum_{u\in V} W_{n-1} = \frac{1}{2}nW_{n-1} \longrightarrow 1$$

as $n \to \infty$.

We will now prove that $\mathbb{E}[\omega(a_u)] \geqslant W_{n-1}$ for all $u \in V$. Although this is intuitively clear, we have to take into account that $\mathscr{B}$ and $\mathscr{A}^*$ are defined on different probability spaces. Consider two vertices u and v. Assume that both algorithms choose for u the edge $\{u, v\}$. Then $\mathscr{B}$ will also make some choice for v while it is possible that $\mathscr{A}^*$ does nothing. Thus we can not exclude that $\mathscr{B}$ performs worse than $\mathscr{A}^*$. Now, let $u \in V$. We write $i(e)$ for the *index of an edge e at u* which is defined as follows. If e is the jth edge incident to u in the order fixed by $\mathscr{FO}$ then $i(e) = j$. In other words, we take all edges incident to u and index them by 1 to $n-1$. Let $i(a_u)$ and $i(b_u)$ denote the index at u of the edges a_u and b_u, respectively. We can rewrite the expectation of $\omega(a_u)$ as

$$\begin{aligned}\mathbb{E}[\omega(a_u)] = {} & \mathbb{E}[\omega(a_u) \mid i(a_u) < i(b_u)]\Pr\{i(a_u) < i(b_u)\} \\ & + \mathbb{E}[\omega(a_u) \mid i(a_u) \geqslant i(b_u)]\Pr\{i(a_u) \geqslant i(b_u)\}. \end{aligned} \tag{2.3}$$

First, we bound the expectation of $\omega(a_u)$ given that $i(a_u) < i(b_u)$. If this is the case then algorithm $\mathscr{B}$ rejects the edge a_u. Therefore $\omega(a_u)$ is at least $W_{n-i(a_u)+1}$ by definition of $\mathscr{B}$. Hence, we have

$$\mathbb{E}[\omega(a_u) \mid i(a_u) < i(b_u)] \geqslant W_{n-i(a_u)+1} \geqslant W_{n-1}. \tag{2.4}$$

Otherwise, if we condition on $i(a_u) \geqslant i(b_u)$ the expectation of $\omega(a_u)$ can be bounded as follows:

$$\begin{aligned}\mathbb{E}[\omega(a_u) \mid i(a_u) \geqslant i(b_u)] &\geqslant \sum_{k=1}^{n-1}\mathbb{E}[\omega(a_u) \mid i(a_u) \geqslant i(b_u) \wedge i(b_u) = k]\Pr\{i(b_u) = k\} \\ &\geqslant \sum_{k=1}^{n-1} W_{n-k}\Pr\{i(b_u) = k\} \geqslant \sum_{k=1}^{n-1} W_{n-1}\Pr\{i(b_u) = k\} = W_{n-1}. \end{aligned} \tag{2.5}$$

By plugging (2.4) and (2.5) into (2.3), we obtain by (2.2)

$$\mathbb{E}[\omega(\hat{T})] = \frac{1}{2}\sum_{u\in V}\mathbb{E}[\omega(a_u)] \geqslant \frac{n}{2}W_{n-1} \longrightarrow 1$$

as $n \to \infty$. In the definition of $\hat{T}$ in (2.2), for every vertex we have weighted the corresponding edge by $1/2$. This means we only count the equivalent of $\lceil n/2 \rceil$ edges in $\hat{T}$. In order to obtain a lower bound on $\omega(T^*)$, we may add the weight of the $\lfloor n/2 \rfloor - 1$ cheapest edges in the G. Hence, for large enough n we have

$$\mathbb{E}[\omega(T^*)] \geqslant \mathbb{E}[\omega(\hat{T})] + \sum_{i=1}^{\lfloor n/2\rfloor - 1}\frac{i}{N+1} \longrightarrow \frac{5}{4} = c \cdot \zeta(3)$$

for an appropriate $c > 1$. This completes the proof. □

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

3. Bounds for unfair adversaries

In this section, we shall see that it is impossible to compete successfully with $\mathscr{UO}$, *i.e.*, there is no algorithm which is strictly $O(1)$-competitive. Owing to (1.1), this negative result also holds for $\mathscr{UA}$.

Theorem 3.1. *Let $\mathscr{A}$ be an online algorithm. Then $C_{\mathscr{A}}^{\mathscr{UO}} = \Omega(\log n)$.*

Proof. For the sake of brevity we write $\beta_a := \frac{\log n}{2n}$ and $\beta_b := \frac{\log n}{\sqrt{n}}$. We partition E into two subsets E_1 and $E_2 = E \setminus E_1$, such that

$$E_1 := \{e \in E : \beta_a \leqslant \omega(e) \leqslant \beta_b\}.$$

The strategy of $\mathscr{UO}$ is to show $\mathscr{A}$ first all edges in E_1 then those in E_2. Let V_B denote the set of vertices that are not incident to any edge of weight less than β_a. We have

$$\begin{aligned}\mathbb{E}[|V_B|] &= n \Pr\{\forall u \in V \setminus \{v\} : \omega(\{u,v\}) \geqslant \beta_a\} \\ &= n\left(1 - \frac{\log n}{2n}\right)^{n-1} = \Theta(\sqrt{n}).\end{aligned}$$

Furthermore, let the random variable $Y(v)$ count the number of edges in E_1 that are incident to v. By Chernoff's inequality (*cf.* [16]) $Y(v)$ is sharply concentrated around its expectation $\mathbb{E}[Y(v)] = \Theta(\sqrt{n}\log n)$. Hence, there is a.a.s. no vertex v with $Y(v) \geqslant \frac{n}{2}$.

In order to construct a spanning tree, $\mathscr{A}$ must at least find a subgraph of minimum degree 1. Thus $\mathscr{A}$ has to choose an edge at each vertex. Assume that $\mathscr{A}$ rejects all edges in E_1. In this case, the expected total weight is at least

$$\frac{|V_B|}{2}\beta_b = \Theta(\log n),$$

since we need at least $|V_B|/2 = \Theta(\sqrt{n})$ edges from E_2 to cover V_B.

Therefore $\mathscr{A}$ has to accept several edges in E_1. It is impossible for $\mathscr{A}$ to determine whether $v \in V_B$, since the weights of the remaining edges incident to v are independent of the edge weights in E_1. Hence, all $\mathscr{A}$ can do is to accept some edges in E_1 and hope that they cover a sufficiently large fraction of V_B. Even by accepting $n/4$ edges, $\mathscr{A}$ expects to cover at most half of V_B, since

$$\Pr\{v \text{ not covered}\} \geqslant \frac{\binom{n-1}{a}}{\binom{n}{a}} = \frac{n-a}{n} \geqslant \frac{1}{2},$$

where a is the number of vertices covered with the $n/4$ edges. Thus the cost in E_2 is still $\Theta(\log n)$. This means that $\mathscr{A}$ has to accept more than $n/4$ edges in E_1. On the other hand, the minimum weight in E_1 is at least β_a and the cost for accepting $n/4$ edges in E_1 is already $\Theta(\log n)$. Hence, the algorithm pays in expectation at least $\Theta(\log n)$. □

Next, we prove that the lower bound claimed in Theorem 3.1 is asymptotically tight. In particular, we show that there is an algorithm which is strictly $O(\log n)$-competitive against $\mathscr{UA}$. By equation (1.1), this algorithm is also strictly $O(\log n)$-competitive against $\mathscr{UO}$.

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

Theorem 3.2. *There is an algorithm $\mathscr{A}$ achieving* $C_{\mathscr{A}}^{\mathscr{U},\mathscr{A}} = 3\log n + o(1)$.

Proof. Consider the following simple algorithm $\mathscr{A}$. Accept every edge which does not form a cycle with the forest constructed so far and has weight less than $3\log n/n$. Furthermore, we accept edges that are necessary to gain connectedness (emergency edges). If the algorithm is not forced to take edges of weight greater than $3\log n/n$, then $\mathrm{ALG} \leqslant 3\log n$. Otherwise, if the algorithm accepts emergency edges the weight of the solution is still at most $n-1$. We claim that

$$\Pr\{\mathscr{A} \text{ accepts bad edges}\} = O(n^{-3/2}).$$

That is, we may conclude that $\mathbb{E}[\mathrm{ALG}] = 3\log n + o(1)$. The probability that $\mathscr{A}$ accepts emergency edges is equal to the probability that the edges of weight at most $3\log n/n$ do not induce a connected subgraph in G. As the edge weights are uniformly and independently drawn from $[0,1]$, this probability is equal to the probability that a random graph with edge probability $p = 3\log n/n$ is not connected. That the latter is $O\left(n^{-3/2}\right)$ is well known and can also be easily checked using the first moment method. □

4. Expected competitive ratio

The expectation of $\rho := \mathrm{ALG}/\mathrm{OPT}$ is called the *expected competitive ratio*. It was recently considered by Scharbrodt, Schickinger and Steger [18], and Souza and Steger [19] to analyse stochastic online-scheduling problems. The same measure was used before by Coffman and Gilbert [5] in a slightly different context. Specifically, they bounded the average-case performance of the well-known list-scheduling algorithm of Graham [10]. The expected competitive ratio offers several advantages. For example, it shows with high probability the right order of magnitude of ρ, since the probability for $\rho \geqslant t \cdot \mathbb{E}[\rho]$ is by Markov's inequality $1/t$. In addition, $\mathbb{E}[\mathrm{ALG}]$ can be dominated by some 'hard' instances where OPT is also unusually large and therefore we intend to show that Connect also achieves a good expected competitive ratio, *i.e.*, that $\mathbb{E}[\rho] = O(1)$.

Although it is very unlikely, OPT can take very small values, which could mean that $\mathbb{E}[\rho] \to \infty$. The next lemma states that this is not the case and therefore ensures that the expectation of ρ exists.

Lemma 4.1. $\mathbb{E}\left[\rho \mid \mathrm{OPT} \leqslant \frac{1}{3}\right] \Pr\{\mathrm{OPT} \leqslant \frac{1}{3}\} = o(1)$.

Proof. In the course of the proof we will bound the probability for $\mathrm{OPT} \in I_j$, where $I_0 := (n^{-1}, 1/3]$ and $I_j := (n^{-(j+1)}, n^{-j}]$ for $j \geqslant 1$. We define the *contribution* of I_j as $\mathbb{E}[\rho \mid \mathrm{OPT} \in I_j] \Pr\{\mathrm{OPT} \in I_j\}$ and claim that (i) the contribution of I_0 and I_1 is $o(1)$ and (ii) if $j \geqslant 2$ the contribution of I_j is $\mathrm{e}^{-\Theta(n(j-1)\ln n)}$. This implies that

$$\begin{aligned}\mathbb{E}\left[\rho \mid \mathrm{OPT} \leqslant \frac{1}{3}\right] \Pr\left\{\mathrm{OPT} \leqslant \frac{1}{3}\right\} &= \sum_{j=0}^{\infty} \mathbb{E}[\rho \mid \mathrm{OPT} \in I_j] \Pr\{\mathrm{OPT} \in I_j\} \\ &\leqslant o(1) + \sum_{j=2}^{\infty} \mathrm{e}^{-\Theta(n(j-1)\ln n)} = o(1).\end{aligned}$$

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

To complete the proof, it remains to bound the contributions of the I_j. First we consider the case $j = 0$. Let the random variable Y count the number of edges in the graph that have weight at most $1/n$. Clearly $\mu_Y = N/n = (n-1)/2$. Furthermore, let the random variable L count the number of edges in the optimal spanning tree that have weight at most $1/n$. According to these definitions $(n-1-L)n^{-1}$ is a lower bound of OPT, since L edges have positive weight and the remaining $(n-1-L)$ edges have weight at least $1/n$. Hence, assuming that OPT $\leqslant \frac{1}{3}$, this implies $L \geqslant (n-1) - \frac{n}{3}$. With $L \leqslant Y$ we get

$$\Pr\{\mathrm{OPT} \in I_0\} \leqslant \Pr\left\{\mathrm{OPT} \leqslant \frac{1}{3}\right\} \leqslant \Pr\left\{Y \geqslant \frac{2}{3}n\right\} = \Pr\{Y \geqslant (1+\delta)\mu_Y\}$$

for $\delta = \frac{4n}{3(n-1)} - 1$. This probability can be bounded from above by Chernoff's inequality leading to a tail bound of $\mathrm{e}^{-\Theta(n)}$. Since $\mathbb{E}[\rho \mid \mathrm{OPT} \in I_0] \leqslant n^2$, we conclude that the contribution of I_0 to $\mathbb{E}[\rho]$ is $o(1)$.

For the case $j > 0$ the random variable X counts the number of edges having weight less than n^{-j}. Clearly $\mu_X = \frac{n-1}{2}n^{1-j}$. Now, if OPT $\leqslant n^{-j}$, the graph must contain at least $(n-1)$ edges of weight less than n^{-j}. We obtain

$$\Pr\{\mathrm{OPT} \in I_j\} \leqslant \Pr\{\mathrm{OPT} \leqslant n^{-j}\} \leqslant \Pr\{X \geqslant n-1\} = \Pr\{X \geqslant (1+\delta)\mu_X\}$$

for $\delta = 2n^{j-1} - 1$. Once again we can bound this probability with Chernoff's inequality, *i.e.*, $\Pr\{X \geqslant (1+\delta)\mu_X\} \leqslant \mathrm{e}^{\alpha(n)}$ where $\alpha(n) = \mu_X(\delta - (\delta+1)\ln(\delta+1))$. We obtain

$$\begin{aligned}
\alpha(n) &= \frac{n-1}{2}n^{1-j}\left(2n^{j-1} - 1 - 2n^{j-1}\ln(2n^{j-1})\right) \\
&= (n-1) - \frac{n-1}{2}n^{1-j} - (n-1)\ln(2n^{j-1}) \\
&= \begin{cases} -\Theta(n), & j = 1, \\ -(j-1)n\ln n + O(n), & j \geqslant 2. \end{cases}
\end{aligned}$$

Since $\mathbb{E}[\rho \mid \mathrm{OPT} \in I_j] \leqslant n^{j+2} = \mathrm{e}^{(j+2)\ln n}$, the contribution of I_1 is $o(1)$ and for $j \geqslant 2$ the contribution of I_j is $\mathrm{e}^{-\Theta(n(j-1)\ln n)}$. □

We are now ready to prove that the performance of CONNECT is also good, if it is measured with the expected competitive ratio. Specifically, we have the following.

Theorem 4.2. *Algorithm* CONNECT *yields* $\mathbb{E}[\rho] = O(1)$.

Proof. The expectation of ρ is given by

$$\mathbb{E}[\rho \mid \mathrm{OPT} \leqslant 1/3]\Pr\{\mathrm{OPT} \leqslant 1/3\} + \mathbb{E}[\rho \mid \mathrm{OPT} > 1/3]\Pr\{\mathrm{OPT} > 1/3\}.$$

Applying Lemma 4.1 and $\Pr\{\mathrm{OPT} \geqslant 1/3\} \leqslant 1$ immediately yields

$$\mathbb{E}[\rho] = \mathbb{E}[\rho \mid \mathrm{OPT} > 1/3] + o(1).$$

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

On the other hand, $\mathbb{E}[\rho \mid \mathrm{OPT} > 1/3]$ is at most

$$\begin{aligned} 3 \cdot \mathbb{E}[\mathrm{ALG} \mid \mathrm{OPT} > 1/3] &= 3 \int_{1/3}^{\infty} \Pr\{\mathrm{ALG} \geqslant t \mid \mathrm{OPT} > 1/3\} \, \mathrm{d}t \\ &\leqslant \frac{3}{\Pr\{\mathrm{OPT} \geqslant 1/3\}} \int_{1/3}^{\infty} \Pr\{\mathrm{ALG} \geqslant t\} \, \mathrm{d}t \\ &\leqslant \frac{3}{1 - o(1)} \mathbb{E}[\mathrm{ALG}] = O(1), \end{aligned}$$

where the second-to-last inequality follows from Lemma 4.1 and $\rho \geqslant 1$. □

Acknowledgement

We would like to thank Johan Håstad for contributing ideas that helped us to improve our results substantially.

References

[1] Beveridge, A., Frieze, A. and McDiarmid, C. (1998) Random minimum length spanning trees in regular graphs. *Combinatorica* **18** 311–333.
[2] Bollobás, B. (2001) *Random Graphs*, 2nd edn, Vol. 73 of *Cambridge Studies in Advanced Mathematics*, Cambridge University Press.
[3] Borodin, A. and El-Yaniv, R. (1998) *Online Computation and Competitive Analysis*, Cambridge University Press.
[4] Chazelle, B. (2000) A minimum spanning tree algorithm with inverse-Ackermann type complexity. *J. Assoc. Comput. Mach.* **47** 1028–1047.
[5] Coffman, E. G. and Gilbert, E. N. (1985) On the expected relative performance of list scheduling. *Operations Research* **33** 548–561.
[6] Frieze, A. (1985) On the value of a random minimum spanning tree. *Discrete Appl. Math.* **10** 47–56.
[7] Frieze, A. and McDiarmid, C. (1989) On random minimum length spanning trees. *Combinatorica* **9** 363–374.
[8] Frieze, A. and McDiarmid, C. (1997) Algorithmic theory of random graphs. *Random Struct. Alg.* **10** 5–42.
[9] Frieze, A., Ruszinko, M. and Thomas, L. (2000) A note on random minimum length spanning trees. *Electron. J. Combin.* **7**.
[10] Graham, R. (1966) Bounds for certain multiprocessing anomalies. *Bell System Technical J.* **45** 1563–1581.
[11] Hagerup, T. (2004) Simpler computation of single-source shortest paths in linear average time. In *Proc. 21th Symposium on Theoretical Aspects of Computer Science*, pp. 362–369.
[12] Janson, S. (1995) The minimal spanning tree in a complete graph and a functional limit theorem for trees in a random graph. *Random Struct. Alg.* **7** 337–356.
[13] Janson, S., Łuczak, T. and Ruciński, A. (2000) *Random Graphs*, Wiley.
[14] Karp, R. M. and Tarjan, R. E. (1980) Linear expected-time algorithms for connectivity problems. *J. Algorithms* **1** 374–393.
[15] McDiarmid, C., Johnson, T. and Stone, H. (1997) On finding a minimum spanning tree in a network with random weights. *Random Struct. Alg.* **10** 187–204.
[16] Motwani, R. and Raghavan, P. (1995) *Randomized Algorithms*, Cambridge University Press.

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press

[17] Pettie, S. and Ramachandran, V. (2002) An optimal minimum spanning tree algorithm. *J. Assoc. Comput. Mach.* **49** 16–34.
[18] Scharbrodt, M., Schickinger, T. and Steger, A. (2002) A new average case analysis for completion time scheduling. In *Proc. 34th Annual ACM Symposium on Theory of Computing*, pp. 170–178.
[19] Souza, A. and Steger, A. (2004) The expected competitive ratio for weighted completion time scheduling. In *Proc. 21th Symposium on Theoretical Aspects of Computer Science*, pp. 620–631.

https://doi.org/10.1017/S096354830600770X Published online by Cambridge University Press