Approximating Spanning Tree Congestion on Graphs with Polylog Degree

Petr Kolman[0000−0003−2235−0506]

Charles University, Faculty of Mathematics and Physics
Prague, Czech Republic
kolman@kam.mff.cuni.cz
https://kam.mff.cuni.cz/~kolman

Abstract. Given a graph G and a spanning tree T of G, the congestion of an edge $e \in E(T)$, with respect to G and T, is the number of edges uv in G such that the unique path in T connecting the vertices u and v traverses the edge e. Given a connected graph G, the *spanning tree congestion* problem is to construct a spanning tree T that minimizes its maximum edge congestion.

It is known that the problem is NP-hard, and that *every* spanning tree is an $n/2$-approximation, but it is not even known whether an $o(n)$-approximation is possible in polynomial time; by n we denote the number of vertices in the graph G.

We consider the problem on graphs with maximum degree bounded by $\Delta = polylog(n)$ and describe an $o(n)$-approximation algorithm; note that even on this restricted class of graphs the spanning tree congestion can be of order $n \cdot polylog(n)$.

Keywords: Graph sparsification · Congestion · Bisection · Spanning tree.

1 Introduction

The construction of a spanning tree of a given graph G is a cornerstone problem in graph theory and computer science, explored with various objectives over the past century [3]. In the spanning tree *congestion* problem, the objective is to construct a spanning tree T of G minimizing its maximum edge congestion where the congestion of an edge $e \in T$ is the number of edges uv in G such that the unique path between u and v in T passes through e. The problem can be viewed as an extreme instance of graph sparsification (represent the connectivity of a graph by its spanning tree), or as an instance of a graph embedding problem (embed a graph into a tree T, with the restriction that T is a spanning tree).

The problem was first considered, as far as we know, by Simonson [14] under the name Min cut (spanning) tree arrangement; the name spanning tree congestion was coined later by Ostrovskii [10]. Even though the problem has been studied for many years, its complexity is not much understood. It is known that it is NP-hard [7,12], and that *every* spanning tree is an $n/2$-approximation [11], but it is not even known whether an $o(n)$-approximation is possible in polynomial time.

1.1 Our Results

We restrict our attention on graphs in which the degree of every vertex is at most polylogarithmic in the number of vertices. For these graphs, the spanning tree congestion ranges between 1 and $\Theta(n \cdot polylog(n))$ (cf. the lower bound (3) below). We describe a polynomial-time $o(n)$-approximation algorithm, namely an $\tilde{\mathcal{O}}(n^{1-1/(\sqrt{\log n}+1)})$-approximation[1]. This provides a partial answer to the question (Problem 2.21) posted by Otachi [11].

The analysis of our algorithm exploits two new lower bounds on the spanning tree congestion that are of independent interest.

1.2 Selected Related Results

The spanning tree congestion problem is known to be NP-complete even for graphs with one vertex of unbounded degree and all other vertices of bounded degree [2]. For planar graphs, the problem is NP-hard as well [12]. Unless P=NP, no c-approximation with c smaller than 6/5 is possible [8]. The decision version k-STC of the problem, to determine whether a given graph has a spanning tree congestion at most k, is solvable in polynomial time for $k \leq 3$, and is NP-complete for $k \geq 5$ [8]; NP-completeness of 5-STC implies the above mentioned inapproximability bound.

The problem k-STC can be solved in linear time for every fixed k on apex-minor-free graphs, a general class of graphs containing planar graphs, graphs of bounded treewidth, and graphs of bounded genus [2] (cf. [12]). In the same paper, the authors also show that for every fixed d and k, there is a linear time algorithm for k-STC on graphs with maximum degree at most d; the result is based on a close connection between the spanning tree congestion and the treewidth that holds for bounded degree graphs.

Regarding algorithms for the hard instances, there is an exact exponential time algorithm for solving them [9].

For a more detailed overview of other related results, we refer to the survey paper by Otachi [11].

1.3 Sketch of our Approximation

Our algorithm uses the *Divide and conquer* framework. If the graph is small enough, we use any spanning tree. Otherwise, we partition the graph by an approximation of the minimum bisection into two or more components of connectivity, each with at most $n/2$ vertices, solve the problem recursively for each of the components, and then combine the spanning trees of the components into a spanning tree of the entire graph.

The challenge is to relate the congestion of the final spanning tree to the congestion of the optimal spanning tree. To be able to do so, we prove two

[1] In the Big-O-Tilde notation $\tilde{\mathcal{O}}$, we ignore polylogarithmic factors; note that for every $c > 0$, $\log^c n = o(n^{1/(\sqrt{\log n}+1)})$.

lower bounds on the spanning tree congestion. The first one provides a relation between the bisection of a graph and the spanning tree congestion (Corollary 1), the other provides a connection between the spanning tree congestion of the entire graph and the spanning tree congestion of its subgraph (Lemma 5).

The analysis of the algorithm is based on the following idea: if there is a component with *large* bisection, where the exact meaning of *large* depends on the recursion level, the combination of Corollary 1 and Lemma 5 yields a *strong* lower bound on the optimal spanning tree congestion ensuring that *any* spanning tree has the desired approximation property; on the other hand, if all components that are used by the algorithm have small bisections, then it is possible to get a stronger bound on the spanning tree congestion of the spanning tree constructed by the algorithm.

1.4 Preliminaries

Given an undirected graph $G = (V, E)$ and a subset of vertices $S \subset V$, we denote by $E_G(S, V \setminus S)$ the set of edges between S and $V \setminus S$ in G, and by $e_G(S, V \setminus S) = |E_G(S, V \setminus S)|$ the number of these edges; if the graph which we are referring to is clear from the context, we avoid the lower index G. An edge $\{u, v\} \in E$ is also denoted by uv for notational simplicity; when the vertices of an edge are not important, we simply talk about an edge $e \in E$. For a subset of vertices $S \subseteq V$, $G[S]$ is the subgraph induced by S. By $V(G)$, we mean the vertex set of the graph G and by $E(G)$ its edge set. By $d(v)$ we denote the degree of a vertex $v \in V$, and by $\Delta(i)$ the sum of the i largest vertex degrees in G. Given a graph $G = (V, E)$ and an edge $e \in E$, $G \setminus e$ is the graph $(V, E \setminus \{e\})$.

Let $G = (V, E)$ be a connected graph and $T = (V, E_T)$ be a spanning tree of G. For an edge $uv \in E_T$, we denote by $S_u, S_v \subset V$ the vertex sets of the two connected components of $T \setminus uv$. We define the *congestion* $c(uv)$ *of the edge* uv *with respect to* G *and* T as $c(uv) = e_G(S_u, S_v)$. The *congestion* $c(G, T)$ *of the spanning tree* T *of* G is defined as $\max_{e \in E_T} c(e)$, and the *spanning tree congestion* $\mathsf{STC}(G)$ *of* G is defined as the minimum value of $c(G, T)$ over all spanning trees T of G.

A *bisection* of a graph with n vertices is a partition of its vertices into two sets, S and $V \setminus S$, each of size at most $\lceil n/2 \rceil$; in approximations, this requirement is sometimes relaxed to $2n/3$ (or to some other fraction). The *width of a bisection* $(S, V \setminus S)$ is $e(S, V \setminus S)$. The minimum width of a bisection of a graph G is denoted $b(G)$.

There are several approximation and pseudo-approximation algorithms for the minimum bisection [1,4]. In our algorithm, the congestion of the constructed spanning tree grows exponentially with the depth of the recursion of the algorithm (cf. the proof of Lemma 7). Thus, when approximating the minimum bisection, the balance of the two parts is more important than the exact approximation factor. For this reason, we will employ the approximation algorithm by Räcke [13] and not the approximation by Arora, Rao and Vazirani [1], even though the latter has better approximation.

Theorem 1 (Räcke [13]). *A bisection of width within a ratio of $O(\log n)$ of the optimum can be computed in polynomial time.*

We will also need an existential result about the size of the minimum exact bisection in trees.

Lemma 1 (Folklore, cf. [5]). *If $T = (V, E)$ is a tree on n vertices and maximum degree Δ, then there exists a bisection of width at most $\Delta \cdot \log n$. Moreover, for every $k < n$, there exists a cut $S \subseteq V$ such that $|S| = k$ and $e(S, V \setminus S) \leq \Delta \cdot \log n$.*[2]

2 Lower Bounds

An indispensable component of the analysis of any approximation algorithm is an appropriate lower bound. In this section, we provide several known and several new lower bounds on the spanning tree congestion. Even though we do not need the old lower bounds in the analysis of our algorithm, we provide them here for comparison. The new lower bounds are not difficult to prove but we are not aware of these results; both Corollary 1 and Lemma 5 are crucial ingredients for the analysis of the algorithm in the next section.

We need two more definitions. For each pair $\{u, v\}$ of vertices of G we denote by $m(u, v)$ the maximal number of edge-disjoint paths between u and v in G. The *conductance* $\phi(G)$ of a graph $G = (V, E)$ (a.k.a. the Cheeger constant of G) is defined as

$$\phi(G) = \min_{\substack{S \subset V \\ |S| \leq |V|/2}} \frac{e(S, V \setminus S)}{\sum_{v \in S} d(v)} .$$

Lemma 2 (Folklore, cf. [11]). *For every graph $G = (V, E)$ on n vertices with m edges,*

$$\mathsf{STC}(G) \geq \frac{2m}{n-1} - 1 . \tag{1}$$

Lemma 3 (Ostrovskii [10]). *For every graph $G = (V, E)$ of maximum degree Δ, minimum degree δ and diameter $diam(G)$,*

$$\mathsf{STC}(G) \geq \max_{u,v \in V} m(u, v) , \tag{2}$$

$$\mathsf{STC}(G) \geq \frac{\delta}{\Delta} \cdot \phi(G) \cdot (n-1) , \tag{3}$$

$$\mathsf{STC}(G) \geq \delta \cdot \phi(G) \cdot \left\lfloor \frac{diam(G)}{2} \right\rfloor . \tag{4}$$

[2] Both results hold even in a slightly stronger form where $\Delta \cdot \log n$ is replaced by $\Delta(\log n)$, the sum of the $\log n$ largest degrees.

Lemma 4. *For every graph $G = (V, E)$ on n vertices with maximum degree Δ, and for every $k < n$,*

$$\mathsf{STC}(G) \geq \frac{\min_{\substack{S \subset V \\ |S|=k}} e(S, V \setminus S)}{\Delta \cdot \log n} \ . \tag{5}$$

Proof. Let $C = \min_{\substack{S \subset V \\ |S|=k}} e(S, V \setminus S)$ be the minimum cut-size of a k-size cut in G and let T be the spanning tree of G with the minimum congestion. By Lemma 1, for the minimum k-size cut S of T, $e_T(S, V \setminus S) \leq \Delta \cdot \log n$. Thus, the congestion of the tree T is at least $C/(\Delta \cdot \log n)$. □

Corollary 1. *For every graph $G = (V, E)$ on n vertices with maximum degree Δ,*

$$\mathsf{STC}(G) \geq \frac{b(G)}{\Delta \cdot \log n} \ . \tag{6}$$

Before stating the next lemma, we note that the spanning tree congestion of a connected subgraph of G might be much larger then the spanning tree congestion of G; the lemma provides some bounds on the increase.

Lemma 5. *For every graph G and every subset of vertices $S \subset V$ that induces a connected subgraph,*

$$\mathsf{STC}(G) \geq \frac{\mathsf{STC}(G[S])}{e(S, V \setminus S)} \ . \tag{7}$$

Proof. Consider an arbitrary subset of vertices $S \subset V$. We start by showing a slightly different bound, namely

$$\mathsf{STC}(G) \geq \frac{\mathsf{STC}(G[S])}{1 + \frac{e(S, V \setminus S)}{2}} \ . \tag{8}$$

We do so by constructing a spanning tree of $G[S]$ of congestion at most $(1 + \frac{e(S,V \setminus S)}{2}) \cdot \mathsf{STC}(G)$. Let T be the spanning tree of G with congestion $\mathsf{STC}(G)$. Let $F \subset E$ be the subset of edges $uv \in E$ with $u, v \in S$ such that the path between u and v in T uses at least one vertex from $V \setminus S$. Note that for every edge $uv \in F$, the path between u and v in T has to use at least two edges from $E(S, V \setminus S)$. Thus, $|F| \leq \mathsf{STC}(G) \cdot e(S, V \setminus S)/2$.

Let T' be an arbitrary extension of the forest $T[S]$ into a spanning tree of $G[S]$. It follows from the above observation that the congestion $c(G[S], T')$ of the spanning tree T' of $G[S]$ is at most

$$c(G[S], T') \leq \mathsf{STC}(G) + |F| \leq \mathsf{STC}(G) + \mathsf{STC}(G) \cdot \frac{e(S, V \setminus S)}{2} \ ,$$

which completes the proof of the inequality (8).

At this point, we distinguish two cases. If $e(S, V \setminus S) = 1$, then $\mathsf{STC}(G) \geq \mathsf{STC}(G[S]) = \frac{\mathsf{STC}(G[S])}{e(S,V \setminus S)}$; if $e(S, V \setminus S) \geq 2$, then $1 + \frac{e(S,V \setminus S)}{2} \leq e(S, V \setminus S)$. Thus, the inequality (7) holds in both cases. □

3 Approximation for Graphs with Degrees Bounded by Polylog

Let $G = (V, E)$ be a connected graph on n vertices with maximum degree $\Delta \leq$ *polylog*(n) and let $k = k(n) = \lceil \sqrt{\log n} \rceil + 1$. We construct the spanning tree of G by the procedure ConstructST(H, s, σ); the procedure is called with parameters $H = G$, $s = \frac{n}{2^{k-1}}$, $\sigma = 1$.

Algorithm 1 ConstructST(H, s, σ)

construct an approximate bisection $(S, V(H) \setminus S)$ of H
$F \leftarrow E(S, V(H) \setminus S)$; $b \leftarrow |F|$
if $b/\sigma \geq n^{1/k}$ or $|V(H)| \leq s$ **then**
    return any spanning tree of H
for each connected component C of $H \setminus F$ **do**
    $T_C \leftarrow$ ConstructST$(C, s, \sigma + b)$
arbitrarily connect all the spanning trees T_C by edges from F to form a spanning tree T of H
return T

Let τ denote the tree representing the recursive decomposition of G (implicitly) constructed by the procedure ConstructST: The root r of τ corresponds to the graph G, and the children of a non-leaf node $t \in \tau$ associated with a set V_t correspond to the connected components of $G[V_t] \setminus F$ where F is the set of cut edges of an approximate bisection of $G[V_t]$ computed by Räcke's algorithm (Theorem 1); we denote by $b_t = |F|$ the width of this bisection of the subgraph $G[V_t]$. The *level* $l(r)$ of the root r of τ is zero, and the *level* of every child t' of $t \in \tau$ is $l(t') = l(t) + 1$. We denote by $G_t = G[V_t]$ the subgraph of G induced by the vertex set V_t, by T_t the spanning tree constructed for G_t by the procedure ConstructST; note that for every tree node $t \in \tau$, by construction the graph G_t is connected.

A tree node $t \in \tau$ at level l is *preferable* if either $l = 0$ and $b_t \geq n^{1/k}$, or $l \geq 1$ and $b_t \geq n^{1/k} \cdot \sum_{t' \in p} b_{t'}$ where p is the unique path in τ between the root r of τ and the immediate predecessor of the node t. A tree node that is not preferable is *tolerable*.

Lemma 6. *If there is at least one preferable node $t \in \tau$, then the spanning tree constructed by the procedure* ConstructST *is an $\tilde{\mathcal{O}}(n^{1-1/k(n)})$-approximation.*

Proof. Consider a preferable node $t \in \tau$ and let $l = l(t)$ denote its level. If $l = 0$, by the definition of the preferable node and by the $\mathcal{O}(\log n)$ bound on the approximation ratio of Räcke's algorithm for minimum bisection (Theorem 1) we know that the minimum bisection of G is of size $\Omega\left(\frac{n^{1/k}}{\log n}\right)$. Thus, by Corollary 1,

$$\mathsf{STC}(G) = \Omega\left(\frac{n^{1/k}}{\Delta \cdot \log^2 n}\right) .$$

If $l \geq 1$, then by the construction of the sets $V_{t'}$ by the algorithm, it follows that $e(V_t, V \setminus V_t) \leq \sum_{t' \in p} b_{t'}$ where p is again the unique path between the root r and the immediate predecessor of t. Combining this bound with Lemma 5, Corollary 1, the definition of the preferable node, and the $\mathcal{O}(\log n)$ bound on the approximation of minimum bisection, we obtain

$$\mathsf{STC}(G) \geq \frac{\mathsf{STC}(G_t)}{e(V_t, V \setminus V_t)} = \Omega\left(\frac{b(G_t)}{\Delta \cdot \log n \cdot \sum_{t' \in p} b_{t'}}\right) = \Omega\left(\frac{n^{1/k}}{\Delta \cdot \log^2 n}\right) .$$

In both cases, the lower bound on $\mathsf{STC}(G)$ immediately yields the claim of the lemma as the congestion of any spanning tree T of G is at most $O(\Delta \cdot n)$. □

Lemma 7. *Let $t \in \tau$ be a tolerable node at level $l = l(t)$ such that all its predecessors $t_{l-1}, \ldots, t_1, t_0 = r$ (i.e., the nodes on the path from t to the root r in τ) are tolerable. Then (with $t_l = t$)*

$$\sum_{i=0}^{l} b_{t_i} = \mathcal{O}(b_r \cdot n^{1-1/k}) . \tag{9}$$

Proof. By induction on the level l of the node $t \in \tau$, we first prove that

$$\sum_{i=0}^{l} b_{t_i} \leq b_r \cdot (1 + n^{1/k})^l . \tag{10}$$

For $l = 0$, the relation (10) asserts simply $b_r \leq b_r$, and this holds.

For the inductive step, assume that the relation (10) holds for l and we want to prove it for $l+1$. As we are looking at a tolerable node, by the assumption of the lemma, we have $b_{t_{l+1}} < n^{1/k} \cdot \sum_{i=0}^{l} b_{t_i}$. Thus, using the inductive assumption, we derive

$$\begin{aligned}\sum_{i=0}^{l+1} b_{t_i} &= \sum_{i=0}^{l} b_{t_i} + b_{t_{l+1}} \\ &\leq \sum_{i=0}^{l} b_{t_i} \cdot (1 + n^{1/k}) \\ &\leq b_r \cdot (1 + n^{1/k})^{l+1}\end{aligned}$$

which completes the proof of the inequality (10).

By construction, for every node $t \in \tau$, the size of V_t is bounded by

$$|V_t| \leq \frac{n}{2^{l(t)}} .$$

Considering the condition in step 3 of the procedure ConstructST and our choice of the value of $s = \frac{n}{2^{k-1}}$, we conclude that $l(t) \leq k - 1$ for every node

$t \in \tau$. Combining this bound with the inequality (10), we obtain the desired estimate

$$\sum_{i=0}^{l(t)} b_{t_i} \leq b_r \cdot (1 + n^{1/k})^{k-1} = O(b_r \cdot n^{1-1/k})$$

where the last inequality uses the bound

$$\left(1 + n^{1/(\sqrt{\log n}+1)}\right)^{\sqrt{\log n}} \leq 3 \cdot n^{1-1/(\sqrt{\log n}+1)}$$

which follows from the analytical properties of the functions involved. □

Lemma 8. *Given a graph $G = (V, E)$, a subset $S \subset V$, a spanning tree T_S of $G[S]$, a spanning tree $T_{V \setminus S}$ of $G[V \setminus S]$, and an edge $f \in E(S, V \setminus S)$, let $T = (V, E(T_S) \cup E(T_{V \setminus S}) \cup \{f\})$. Then T is a spanning tree of G and*

$$c(G, T) \leq \max\{c(G[S], T_S), c(G[V \setminus S], T_{V \setminus S})\} + e(S, V \setminus S) \ .$$

Proof. The fact that T is a spanning tree of G follows immediately from the construction of T. For every $e \in E(T_S)$, the congestion $c(e)$ of e with respect to G and T is at most $c(e) \leq c(G[S], T_S) + e(S, V \setminus S)$, similarly, for every $e \in E(T_{V \setminus S})$, the congestion $c(e)$ of e with respect to G and T is at most $c(e) \leq c(G[V \setminus S], T_{V \setminus S}) + e(S, V \setminus S)$, and the congestion of f is at most $c(f) \leq e(S, V \setminus S)$. □

Corollary 2. *Given a connected graph $G = (V, E)$ and a subset $F \subset E$ of edges, let $V_1, V_2, \ldots, V_k$ be the vertex sets of the connected components of $G \setminus F$ ordered in such a way that for each $l = 2, \ldots, k$, there is at least one edge f_l in G between $\bigcup_{i=1}^{l-1} V_i$ and V_l. If for each $i = 1, \ldots, k$, T_i is a spanning tree of $G[V_i]$, then $T = (V, \bigcup_{i=1}^{k} E(T_i) \cup \{f_2, f_3, \ldots, f_k\})$ is a spanning tree of G and*

$$c(G, T) \leq \max_{i=1,\ldots,k}\{c(G[V_i], T_i)\} + |F| \ .$$

Proof. For $i = 1, \ldots, k$, let $G_i = G[\bigcup_{j=1}^{i} V_j]$ and $T'_i = (V(G_i), \bigcup_{j=1}^{i} E(T_j) \cup \{f_2, f_3, \ldots, f_i\})$. By induction on i, we are going to show that T'_i is a spanning tree of G_i and

$$c(G_i, T'_i) \leq \max_{j=1,\ldots,i}\{c(G[V_j], T_j)\} + \sum_{j=2}^{i} e(\bigcup_{l=1}^{j-1} V_l, V_j) \ . \tag{11}$$

As $G_k = G$, $T'_k = T$ and $F \subseteq \bigcup_{j=2}^{k} E(\bigcup_{l=1}^{j-1} V_l, V_j)$, this will complete the proof.

For $i = 1$, the inequality (11) simply states that $c(G_1, T'_1) \leq (G_1, T_1)$ which holds, as $T'_1 = T_1$.

For the inductive step, note that by Lemma 8,

$$c(G_{i+1}, T'_{i+1}) \leq \max\{c(G_i, T'_i), c(G[V_{i+1}], T_{i+1})\} + e(\bigcup_{j=1}^{i} V_j, V_{i+1}) \ .$$

Applying the inductive assumption on $c(G_i, T'_i)$ and using the fact that for any three non-negative numbers a, b, c, $\max\{a + b, c\} \leq \max\{a, c\} + b$, we obtain the desired bound on $c(G_{i+1}, T'_{i+1})$:

$$c(G_{i+1}, T'_{i+1}) \leq \max \left\{ \max_{j=1,\ldots,i} \{c(G[V_j], T_j)\} + \sum_{j=2}^{i} e(\bigcup_{l=1}^{j-1} V_l, V_j),\ c(G[V_{i+1}], T_{i+1}) \right\} + e(\bigcup_{j=1}^{i} V_j, V_{i+1})$$

$$\leq \max_{j=1,\ldots,i+1} \{c(G[V_j], T_j)\} + \sum_{j=2}^{i+1} e(\bigcup_{l=1}^{j-1} V_l, V_j)$$

□

Lemma 9. *If there is no preferable node $t \in \tau$, then the spanning tree T constructed by the* ConstructST *procedure is an $\tilde{O}(n^{1-1/k})$-approximation.*

Proof. For a node $t \in \tau$, let $L(t)$ denote the set of all leaves of the subtree of τ rooted in t, and for a leaf $s \in L(t)$, let $p(s, t)$ denote the set of all nodes on the unique path between s and t in the tree τ. Recall that for a non-leaf node $t \in \tau$, b_t is the size of the bisection of the graph G_t that was used by the procedure ConstructST; for notational simplicity of the next steps, we define b_t for every leaf $t \in \tau$ as well, namely as $b_t = 0$. For every node $t \in \tau$, we are going to prove the following bound:

$$c(G_t, T_t) \leq \max_{s \in L(t)} \left\{ c(G_s, T_s) + \sum_{u \in p(s,t), u \neq t} b_u \right\} + b_t \tag{12}$$

We proceed by induction, from bottom up in the tree τ. Consider first a leaf $t \in \tau$. In this case, the inequality (12) simplifies to $c(G_t, T_t) \leq c(G_t, T_t) + b_t$ which holds as $b_t = 0$. Next, consider a non-leaf node $t \in \tau$ such that the inequality (12) holds for all its children; let $t_1, \ldots, t_k$ be the children of t, for some $k \geq 2$. By applying Corollary 2 on the graph G_t, we obtain

$$c(G_t, T_t) \leq \max_{i=1,\ldots,k} \{c(G_{t_i}, T_{t_i})\} + b_t \ .$$

Plugging in the bound (12) for each of the subgraphs G_{t_i}, a simple manipulation yields the desired bound (12) for the node t as well.

Note that for each leaf $s \in \tau$,

$$c(G_s, T_s) = O(\Delta \cdot n^{1-1/k}) \ , \tag{13}$$

and,

$$\sum_{u \in p(s,r)} b_u = O(n^{1-1/k}) \cdot b_r \ ; \tag{14}$$

the first bound follows from construction as the algorithms ensures $|V_s| \le \frac{n}{2^{k-1}} \le n^{1-1/k}$ for each leaf s, and the second bound is given in Lemma 7. Thus, by application of the inequality (12) for the root r of the tree τ, we obtain, using (13) and (14),

$$c(G,T) \le \max_{s \in L(r)} \{c(G_s, T_s)\} + O(b_r \cdot n^{1-1/k}) = O((\Delta + b_r) \cdot n^{1-1/k}) \ . \qquad (15)$$

Exploiting once more the bound on the approximation ratio of Räcke's algorithm for the minimum bisection (Theorem 1), namely $b(G) = \Omega(\frac{b_r}{\log n})$, Corollary 1 yields

$$\mathsf{STC}(G) = \Omega\left(\frac{b_r}{\Delta \cdot \log^2 n}\right) \ .$$

Combining this lower bound on the minimum spanning tree congestion of G with the upper bound (15) on the congestion $c(G,T)$ of the spanning tree constructed by the algorithm, the proof is completed. □

Lemmas 6 and 9 yield the main theorem.

Theorem 2. ConstructST *is an $\tilde{\mathcal{O}}(n^{1-1/k})$-approximation algorithm for the minimum congestion spanning tree problem for graphs with maximum degree bounded by $\Delta = polylog(n)$.*

4 Conclusion

We have designed an $o(n)$-approximation algorithm for the spanning tree congestion problem for graphs with maximum degree bounded by $polylog(n)$. Apparently, there are many open problems concerning the spanning tree congestion. Here are some of them:

- P1 Is the decision problem 4-STC solvable in polynomial time, or is it NP-complete [8]?
- P2 Is it possible to extend the algorithm ConstructST to work for graphs of unbounded degree?
- P3 Is the spanning tree congestion problem NP-hard for graphs with all degrees bounded by a constant? The known NP-hardness proofs require at least one vertex of unbounded degree.

To state the next problem, we need one more definition. Following Law and Ostrovskii [6], given a graph $G = (V,E)$ and an integer c, we define the value $f(G,c)$ as

$$f(G,c) = \min\{e(S, V \setminus S) \mid S \subset V, |S| = c, \ G[S] \text{ is connected}\} \ . \qquad (16)$$

They observe that

$$\mathsf{STC}(G) \ge \min\left\{ f(G,c) \ \middle| \ \left\lceil \frac{n-1}{\Delta} \right\rceil \le c \le \frac{n}{2} \right\} \ . \qquad (17)$$

The above lower bound is based on the size of a cut S satisfying three properties: i) the subgraph induced by S is connected, ii) the subset S has a prescribed size, iii) the number of edges $e(S, V \setminus S)$ is the smallest among all subsets satisfying the properties i) and ii); we are going to call the task of finding such a cut the *minimum connected c-cut* problem. As far as we know, if only any two of these three properties are required (i.e., minimum c-cut, not necessarily connected, or minimum connected cut, not necessarily of size c, or connected cut of size c, not necessarily minimum), then the problem of finding such a cut is solvable, or at least reasonably approximable, in polynomial time; however, we are not aware of any non-trivial approximation if all three requirements have to be at least approximately satisfied.

P4 Design an approximation algorithm for the minimum connected c-cut.

Acknowledgments The author thanks Marek Chrobak for introducing him to the spanning tree congestion problem.

References

1. S. Arora, S. Rao, and U. V. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):5:1–5:37, 2009. Preliminary version in *Proc. of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, 2004.
2. H. L. Bodlaender, F. V. Fomin, P. A. Golovach, Y. Otachi, and E. J. van Leeuwen. Parameterized complexity of the spanning tree congestion problem. *Algorithmica*, 64(1):85–111, 2012.
3. O. Borůvka. O jistém problému minimálním (About a certain minimal problem). *Práce Moravské přírodovědecké společnosti*, III(3):37–58, 1926.
4. U. Feige and R. Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM J. Comput.*, 31(4):1090–1118, 2002.
5. C. G. Fernandes, T. J. Schmidt, and A. Taraz. On minimum bisection and related cut problems in trees and tree-like graphs. *J. Graph Theory*, 89(2):214–245, 2018.
6. H.-F. Law and M. Ostrovskii. Spanning tree congestion: duality and isoperimetry; with an application to multipartite graphs. *Graph Theory Notes of New York*, 58:18–26, 2010.
7. C. Löwenstein. *In the Complement of a Dominating Set.* PhD thesis, TU Ilmenau, April 2010.
8. H. Luu and M. Chrobak. Better hardness results for the minimum spanning tree congestion problem. In *Proc. of 17th International Conference and Workshops on Algorithms and Computation (WALCOM)*, volume 13973 of *Lecture Notes in Computer Science*, pages 167–178, 2023.
9. Y. Okamoto, Y. Otachi, R. Uehara, and T. Uno. Hardness results and an exact exponential algorithm for the spanning tree congestion problem. *J. Graph Algorithms Appl.*, 15(6):727–751, 2011.
10. M. Ostrovskii. Minimal congestion trees. *Discrete Mathematics*, 285(1):219–226, 2004.
11. Y. Otachi. A survey on spanning tree congestion. In *Treewidth, Kernels, and Algorithms: Essays Dedicated to Hans L. Bodlaender on the Occasion of His 60th Birthday*, volume 12160 of *Lecture Notes in Computer Science*, pages 165–172, 2020.

12. Y. Otachi, H. L. Bodlaender, and E. J. van Leeuwen. Complexity results for the spanning tree congestion problem. In *Proc. of 36th International Workshop on Graph Theoretic Concepts in Computer Science (WG)*, volume 6410 of *Lecture Notes in Computer Science*, pages 3–14, 2010.
13. H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proc. of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 255–264. ACM, 2008.
14. S. Simonson. A variation on the min cut linear arrangement problem. *Math. Syst. Theory*, 20(4):235–252, 1987.