A Note on Approximation of Spanning Tree Congestion

Petr Kolman

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

Abstract. The *Spanning Tree Congestion* problem is an easy-to-state NP-hard problem: given a graph G, 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 optimum value for a given graph G is denoted $\mathsf{STC}(G)$.

It is known that *every* spanning tree is an $n/2$-approximation. A long-standing problem is to design a better approximation algorithm. Our contribution towards this goal is an $\mathcal{O}(\Delta \cdot \log^{3/2} n)$-approximation algorithm for the minimum congestion spanning tree problem where Δ is the maximum degree in G. For graphs with maximum degree bounded by polylog of the number of vertices, this is an exponential improvement over the previous best approximation. For graphs with maximum degree bounded by $o(n/\log^{3/2} n)$, we get $o(n)$-approximation; this is the largest class of graphs that we know of, for which sublinear approximation is known for this problem.

Our main tool for the algorithm is a new lower bound on the spanning tree congestion which is of independent interest. We prove that for every graph G, $\mathsf{STC}(G) \geq \Omega(hb(G)/\Delta)$ where $hb(G)$ denotes the maximum bisection width over all subgraphs of G.

1 Introduction

The spanning tree congestion problem has been studied from various aspects for more than twenty years, yet our ability to approximate it is still extremely limited. In particular, no $o(n)$-approximation algorithm for general graphs has been designed, as far as we know. The most general class of graphs for which $o(n)$-approximation algorithm exists are graphs with maximum degree bounded by polylog of the number of vertices [5].

On the other side, the strongest known hardness result is that no c-approximation with c smaller than $6/5$ is possible unless $P = NP$ [7]. Not many problems have the gap between the best upper and lower bounds of order $\Omega(n)$.

For a more detailed overview of other related results, we refer to the survey paper by Otachi [9] and to the recent paper by Kolman [5].

1.1 Our Results

Our contribution in this note is twofold. We describe an $\mathcal{O}(\Delta \cdot \log^{3/2} n)$-approximation algorithm for the minimum congestion spanning tree problem where Δ is the maximum degree in G and n the number of vertices. For graphs with maximum degree bounded by $\Delta = o(n/\log^{3/2} n)$, we get $o(n)$-approximation; this significantly extends the class of graphs for which sublinear approximation is known, and provides a partial answer to the open problem P2 from the recent paper [5]. Moreover, for graphs with stronger bound on the maximum degree, the approximation ratio is even stronger than $o(n)$. For example, for graphs with maximum degree bounded by polylog of the number of vertices, the approximation is polylogarithmic; the previous best apporoximation for this class of graphs was $\tilde{\mathcal{O}}(n^{1-1/(\sqrt{\log n}+1)})$ only[1] [5].

For planar graphs (and more generally any proper minor closed family of graphs) we get a slightly better bound of $\mathcal{O}(\Delta \cdot \log n)$ on the approximation ratio.

Our key tool in the algorithm design is a new lower bound on $\mathsf{STC}(G)$ which is our second contribution. In a recent paper, Kolman [5] proved that $\mathsf{STC}(G) \geq \frac{b(G)}{\Delta \cdot \log n}$ where $b(G)$ is the bisection

[1] In the Big-O-Tilde notation $\tilde{\mathcal{O}}$, we ignore polylogarithmic factors.

of G. We strengthen the bound and prove that $\mathsf{STC}(G) \geq \Omega(\frac{hb(G)}{\Delta})$ where $hb(G)$ is the *hereditary bisection* of G which is the maximum of $b(H)$ over all subgraphs H of G. This is a corollary of another new lower bound saying that $\mathsf{STC}(G) \geq \frac{\beta(H)\cdot n'}{3\cdot\Delta}$; here $\beta(H)$ is the expansion of H, n' is the number of vertices in H and the bound holds for every subgraph H of G.

1.2 Sketch of the Algorithm

The algorithm uses the standard *Divide and conquer* framework and is conceptually very simple: partition the graph by a $\frac{2}{3}$-balanced cut into two or more components of connectivity, solve the problem recursively for each of the components, and arbitrarily combine the spanning trees of the components into a spanning tree of the entire graph. The structure of the algorithm is the same as the structure of the recent $o(n)$-approximation algorithm [5] for graphs with maximum degree bounded by $polylog(n)$ - there is a minor difference in the tool used in the partitioning step and in the stopping condition for the recursion.

It is far from obvious that the *Divide and conquer* approach works for the spanning tree congestion problem. The difficulty is that there is no apparent relation between $\mathsf{STC}(G)$ and $\mathsf{STC}(H)$ for a subgraph H of G. Kolman [5] proved that $\mathsf{STC}(G) \geq \frac{\mathsf{STC}(H)}{e(H,G\setminus H)}$ where $e(H, G \setminus H)$ denotes the number of edges between the subgraph H and the rest of the graph G; this bound is one of the two main ingredients of his algorithm [5]. Note that the bound is very weak when $e(H, G\setminus H)$ is large. Also note that the bound is tight in the following sense: there exist graphs for which $\mathsf{STC}(G)$ and $\frac{\mathsf{STC}(H)}{e(H,G\setminus H)}$ are equal, up to a small multiplicative constant. For example, let G be a graph obtained from a 3-regular expander H on n vertices by adding a new vertex r and connecting it by an edge to every vertex of H. Then $\mathsf{STC}(H) = \Omega(n)$ [8] while $\mathsf{STC}(G) = O(1)$ (think about the spanning tree of G consisting only of all the edges adjacent to the new vertex r).

The main reason for the significant improvement of the bound on the approximation ratio of the algorithm is the new lower bound $\mathsf{STC}(G) \geq \Omega(\frac{hb(G)}{\Delta})$ that connects $\mathsf{STC}(G)$ and properties of subgraphs of G in a much tighter way. This connection yields a simpler algorithm with better approximation, wider applicability and simpler analysis.

1.3 Preliminaries

For an undirected graph $G = (V, E)$ and a subset of vertices $S \subset V$, we denote by $E(S, V \setminus S)$ the set of edges between S and $V \setminus S$ in G, and by $e(S, V \setminus S) = |E(S, V \setminus S)|$ the number of these edges. An edge $\{u, v\} \in E$ is also denoted by uv for notational simplicity. 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. 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 connectivity components of $T \setminus uv$. The *congestion* $c(uv)$ *of the edge* uv *with respect to* G *and* T, is the number of edges in G between S_u and 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 $\lceil n/2 \rceil$. 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 $b(G)$. The *hereditary bisection width* $hb(G)$ is the maximum of $b(H)$ over all subgraphs H of G. In approximation algorithms, the requirement that each of the two parts in a partition of V is of size at most $\lceil n/2 \rceil$ is sometimes relaxed to $2n/3$, or to some other fraction, and then we talk about balanced cuts. In particular, a *c-balanced cut* is a partition of the graph into two parts, each of size at most $c \cdot n$.

The *edge expansion* of G is

$$\beta(G) = \min_{A \subseteq V} \frac{e(A, V \setminus A)}{\min\{|A|, |V \setminus A|\}} \ .$$

There are several approximation and pseudo-approximation algorithms for bisection and balanced cuts [1,2]. In our algorithm, we will employ the algorithm by Arora, Rao and Vazirani [1], and for planar graphs (or more generally, for graphs excluding any fixed graph as a minor), a slightly stronger bound by Klein, Protkin and Rao [4].

Theorem 1 ([1,4]). *A 2/3-balanced cut of cost within a ratio of $\mathcal{O}(\sqrt{\log n})$ of the optimum bisection can be computed in polynomial time. For graphs excluding any fixed graph as a minor, even $\mathcal{O}(1)$ ratio is possible.*

We conclude this section with two more theorems that we will refer to later.

Theorem 2 (Jordan [3]). *Given a tree on n vertices, there exists a vertex whose removal partitions the tree into components, each with at most $n/2$ vertices.*

Lemma 1 (Kolman, Matoušek [6]). *Every graph G on n vertices contains a subgraph on at least $2n/3$ vertices with edge expansion at least $b(G)/n$.*

2 New Lower Bound

The main result of this section is captured in the following lemma and its corollary.

Lemma 2. *For every graph $G = (V, E)$ on n vertices with maximum degree Δ and every subgraph H of G on n' vertices, we have*

$$\mathsf{STC}(G) \geq \frac{\beta(H) \cdot n'}{3 \cdot \Delta} \ . \tag{1}$$

Before proving the lemma, we state a slight generalization of Theorem 2; for the sake of completeness, we also provide a proof of it though it is a straightforward extension of the standard proof of Theorem 2.

Claim. Given a tree T on n vertices with $n' \leq n$ vertices marked, there exists a vertex (marked or unmarked) whose removal partitions the tree into components, each with at most $n'/2$ marked vertices.

Proof. Start with an arbitrary vertex $v_0 \in T$ and set $i = 0$. We proceed as follows. If the removal of v_i partitions the tree into components such that each contains at most $n'/2$ marked vertices, we are done. Otherwise, one of the components, say a component C, has strictly more then $n'/2$ marked vertices. Let v_{i+1} be the neighbor of v_i that belongs to the component C. Note that for every $i > 0$, v_i is different from all the vertices $v_0, v_1, \ldots, v_{i-1}$. As the number of vertices in the tree is bounded, eventually this process has to stop and we get to a vertex with the desired properties.

Proof of Lemma 2. Let T be the spanning tree of G with the minimum congestion, and let n' denote the number of vertices in H, $n' = |V(H)|$. By Claim 2, there exists a vertex $z \in T$ whose removal partitions the tree T into components, each with at most $n'/2$ verices from H. We organize the components of $T \setminus z$ into two parts in such a way that the total number of vertices from H in each of the two parts is at least $n'/3$; let $C \subseteq V(H)$ be the vertices from H in one of the two parts. Then, by the definition of expansion, $e(C, V(H) \setminus C) \geq \beta(H) \cdot n'/3$. As for each edge $uv \in E(C, V(H) \setminus C)$, the path connecting u and v in T uses at least one edge adjacent to z, we conclude that

$$\mathsf{STC}(G) \geq \frac{e(C, V(H) \setminus C)}{\Delta} \geq \frac{\beta(H) \cdot n'}{3 \cdot \Delta} \quad .$$

Combining Lemma 2 with Lemma 1, we obtain the next lower bound.

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

$$\mathit{STC}(G) \geq \frac{2 \cdot hb(G)}{9 \cdot \Delta} \quad . \tag{2}$$

3 Approximation Algorithm

Given a connected graph $G = (V, E)$, we construct the spanning tree of G by the recursive procedure CongSpanTree called on the graph G. By the ARV-algorithm, we refer to the algorithm of Theorem 1.

Let τ denote the tree representing the recursive decomposition of G (implicitly) constructed by the procedure CongSpanTree: 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 connectivity components of $G[V_t] \setminus F$ where F is the set of cut edges of the $\frac{2}{3}$-balanced cut of $G[V_t]$ from step 4; recall that $|F| \leq \mathcal{O}(\sqrt{\log n}) \cdot$

Algorithm 1 CongSpanTree(H)

```
1: if |V(H)| = 1 then
2:     return H
3: construct, by the ARV-algorithm, a 2/3-balanced cut (S, V(H) \ S) of H
4: F ← E(S, V(H) \ S)
5: for each connected component C of H \ F do
6:     T_C ← CongSpanTree(C)
7: arbitrarily connect all the spanning trees T_C by edges from F to form a spanning
                                                                     tree T of H
8: return T
```

$b(G[V_t])$. 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 CongSpanTree. The *height* $h(t)$ of a tree node $t \in \tau$ is the number of edges on the longest path from t to a leaf in its subtree (i.e., to a leaf that is a descendant of t).

Lemma 3. *Let $t \in \tau$ be a node of the decomposition tree and $t_1, \dots, t_k$ its children. Then*

$$c(G_t, T_t) \le \max_i c(G_{t_i}, T_{t_i}) + \mathcal{O}(\sqrt{\log n}) \cdot b(G_t) \ . \tag{3}$$

Proof. Let F be the set of edges cut by the $\frac{2}{3}$-balanced cut of G_t (steps 3 and 4). We will show that for every edge $e \in E(T_t)$, its congestion $c(e)$ with respect to G_t and T_t is at most $\max_i c(G_{t_i}, T_{t_i}) + |F|$; as $|F| \le \mathcal{O}(\sqrt{\log n}) \cdot b(G[V_t])$, this will prove the Lemma. Recall that $E(T_t) \subseteq \bigcup_{i=1}^{k} E(T_{t_i}) \cup F$, as the spanning tree T_t is constructed (step 7) from the spanning trees $T_{t_1}, \dots, T_{t_k}$ and the set F.

Consider first an edge $e \in E(T_t)$ that belongs to a tree T_{t_i}, for some i. The only edges from $E(G)$ that may contribute to the congestion $c(e)$ of e with respect to G_t and T_t are the edges in $E(G_{t_i}) \cup F$; the contribution of the edges in $E(G_{t_i})$ is at most $c(G_{t_i}, T_{t_i})$, the contribution of the edges in F is at most $|F|$. Thus, the congestion $c(e)$ of the edge e with respect to G_t and T_t is at most $c(G_{t_i}, T_{t_i}) + |F|$.

Consider now an edge $e \in F \cap E(T_t)$. As the only edges from $E(G)$ that may contribute to the congestion $c(e)$ of e with respect to G_t and T_t are the edges in F, its congestion is at most $|F|$.

Thus, for every edge $e \in E(T_t)$, its congestion with respect to G_t and T_t is at most $\max_i c(G_{t_i}, T_{t_i}) + |F|$, and the proof of the lemma is completed.

Corollary 2. *Let* $T = \textsc{CongSpanTree}(G)$*. Then*

$$c(G,T) \le \mathcal{O}(\log^{3/2} n) \cdot hb(G) \ . \tag{4}$$

Proof. For technical reasons, we extend the notion of the spanning tree congestion also to the trivial graph $H = (\{v\}, \emptyset)$ consisting of a single vertex and no edge (and having a single spanning tree $T_H = H$) by defining $c(H, T_H) = 0$.

By induction on the height of vertices in the decomposition tree τ, we prove the following auxiliary claim: for every $t \in \tau$,

$$c(G_t, T_t) \le h(t) \cdot \mathcal{O}(\sqrt{\log n}) \cdot hb(G) \ . \tag{5}$$

Consider first a node $t \in \tau$ of height zero, that is, a node t that is a leaf. Then both sides of (5) are zero and the inequality holds.

Consider now a node $t \in \tau$ such that for all his children the inequality (5) holds. Let t' be the child of the node t for which $c(G_{t'}, T_{t'})$ is the largest among the children of t. Then, as $b(G_t) \le hb(G)$ by the definition of hb, by Lemma 3 we get

$$c(G_t, T_t) \le c(G_{t'}, T_{t'}) + \mathcal{O}(\sqrt{\log n}) \cdot hb(G) \ .$$

By the inductive assumption applied on the node t',

$$c(G_{t'}, T_{t'}) \le h(t') \cdot \mathcal{O}(\sqrt{\log n}) \cdot hb(G) \ .$$

Because $h(t) \ge h(t')+1$, the proof of the auxiliary claim is completed.

Observing that the height of the root of the decomposition tree τ is at most $\mathcal{O}(\log n)$, as all cuts used by the algorithm are balanced, the proof is completed.

Theorem 3. *Given a graph G with maximum degree Δ, the algorithm* CongSpanTree *constructs an $\mathcal{O}(\Delta \cdot \log^{3/2} n)$-approximation of the minimum congestion spanning tree.*

Proof. By Corollary 1, for every graph G, $\Omega(hb(G)/\Delta)$ is a lower bound on $\mathsf{STC}(G)$. By Corollary 2, the algorithm $\textsc{CongSpanTree}(G)$ constructs a spanning tree T of congestion at most $\mathcal{O}(\log^{3/2} n) \cdot hb(G)$. Combining these two results yields the theorem: $c(G,T) \le \mathcal{O}(\log^{3/2} n) \cdot hb(G) \le \mathcal{O}(\log^{3/2} n \cdot \Delta) \cdot \mathsf{STC}(G)$.

Note that for graphs excluding any fixed graph as a minor, replacing the ARV-algorithm by the algorithm of Klein, Plotkin and Rao in the algorithm CongSpanTree, we get $\mathcal{O}(\Delta \cdot \log n)$-approximation.

4 Open Problems

A self-suggesting question is whether it is possible to eliminate the dependency of the approximation ratio of the algorithm on the largest degree Δ in the graph and obtain an $o(n)$-approximation algorithm for STC for all graphs.

References

1. Sanjeev Arora, Satish Rao, and Umesh 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. Uriel Feige and Robert Krauthgamer. A polylogarithmic approximation of the minimum bisection. *SIAM J. Comput.*, 31(4):1090–1118, 2002.
3. Camille Jordan. Sur les assemblages de lignes. *Journal für die reine und angewandte Mathematik*, 70:185–190, 1869.
4. Philip N. Klein, Serge A. Plotkin, and Satish Rao. Excluded minors, network decomposition, and multicommodity flow. In *Proc. of the Twenty-Fifth Annual ACM Symposium on Theory of Computing (STOC)*, pages 682–690, 1993.
5. Petr Kolman. Approximating spanning tree congestion on graphs with polylog degree. In *Proc. of International Workshop on Combinatorial Algorithms, IWOCA*, pages 497–508, 2024.
6. Petr Kolman and Jiří Matoušek. Crossing number, pair-crossing number, and expansion. *Journal of Combinatorial Theory, Series B*, 92(1):99–113, 2004.
7. Huong Luu and Marek 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.
8. Mikhail I. Ostrovskii. Minimal congestion trees. *Discrete Mathematics*, 285(1):219–226, 2004.
9. Yota 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.