Approximation of Spanning Tree Congestion Using Hereditary Bisection

Petr Kolman
Department of Applied Mathematics, Faculty of Mathematics and Physics, Charles University, Czech Republic

Abstract

The *Spanning Tree Congestion* (STC) problem is the following 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 optimal value for a given graph G is denoted $\mathrm{STC}(G)$.

It is known that *every* spanning tree is an $n/2$-approximation for the STC problem. 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 where Δ is the maximum degree in G and n the number of vertices. For graphs with a maximum degree bounded by a polylog of the number of vertices, this is an exponential improvement over the previous best approximation.

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

2012 ACM Subject Classification Mathematics of computing → Discrete mathematics; Theory of computation → Design and analysis of algorithms

Keywords and phrases Spanning Tree Congestion, Bisection, Expansion, divide and conquer

1 Introduction

The spanning tree congestion problem has been studied from various viewpoints for more than twenty years, yet our ability to approximate it is still extremely limited. It has been shown that every spanning tree is an $n/2$-approximation [9] but no $o(n)$-approximation for *general* graphs is known. For graphs with $\omega(n \log^2 n)$ edges, Chandran et al. [2] described an algorithm that constructs in polynomial time a spanning tree with congestion at most $\mathcal{O}(\sqrt{mn})$; combined with the trivial lower bound $\Omega(m/n)$ on the spanning tree congestion, this yields an $\mathcal{O}(n/\log n)$-approximation. There is also an $\tilde{\mathcal{O}}(n^{1-1/(\sqrt{\log n}+1)})$-approximation[1] algorithm for graphs with maximum degree bounded by polylog of the number of vertices [5].

On the hardness side, the strongest known lower bound states that no c-approximation with c smaller than $6/5$ is possible unless $P = NP$ [8]. The $\Omega(n)$ gap between the best upper and lower bounds is highly unsatisfactory.

For a detailed overview of other related results, we refer to the survey paper by Otachi [9], to our recent paper [5], and to the new paper by Lampis et al. [7] that deals with the STC problem from the perspective of parameterized complexity.

1.1 Our Results

Our contribution in this paper is twofold. We describe an $\mathcal{O}(\Delta \cdot \log^{3/2} n)$-approximation algorithm for the spanning tree congestion 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

[1] The Big-O-Tilde notation $\tilde{\mathcal{O}}$ ignores logarithmic factors.

approximation is known, and provides a partial answer to the open problem P2 from our recent paper [5]. Moreover, for graphs with a stronger bound on the maximum degree, the approximation ratio is even better than $o(n)$. For example, for graphs with a maximum degree bounded by polylog of the number of vertices, the approximation is polylogarithmic which is an exponential improvement over the previous best approximation [5].

For graphs excluding any fixed graph as a minor (e.g., planar graphs or bounded genus 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 $\mathrm{STC}(G)$ which is our second contribution. In the recent paper [5], we proved that $\mathrm{STC}(G) \geq \frac{b(G)}{\Delta \cdot \log n}$ where $b(G)$ is the bisection of G. We strengthen the bound and prove that $\mathrm{STC}(G) \geq \Omega\left(\frac{hb(G)}{\Delta}\right)$ 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 for every subgraph H of G, $\mathrm{STC}(G) \geq \frac{\beta(H) \cdot n'}{3 \cdot \Delta}$; here $\beta(H)$ is the expansion of H and n' is the number of vertices in H.

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 connected components, 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 our 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 $\mathrm{STC}(G)$ and $\mathrm{STC}(H)$ for a subgraph H of G. In the paper [5], we proved that $\mathrm{STC}(G) \geq \frac{\mathrm{STC}(H)}{e(H, G \setminus H)}$ where $e(H, G \setminus H)$ denotes the number of edges between the subgraph H and the rest of the graph G. 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 $\mathrm{STC}(G)$ and $\frac{\mathrm{STC}(H)}{e(H, G \setminus H)}$ 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 $\mathrm{STC}(H) = \Omega(n)$ (cf. Lemma 4) while $\mathrm{STC}(G) = O(1)$ (consider 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 $\mathrm{STC}(G) \geq \Omega\left(\frac{hb(G)}{\Delta}\right)$ that connects $\mathrm{STC}(G)$ and properties of subgraphs of G in a much tighter way. This connection yields a simpler algorithm with better approximation, broader 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 connected components of

$T \setminus uv$ containing u and v, resp. 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* $\mathrm{STC}(G)$ *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 vertices into two sets, each of size at most $c \cdot n$. The *edge expansion* of G is

$$\beta(G) = \min_{A \subseteq V} \frac{e(A, V \setminus A)}{\min\{|A|, |V \setminus A|\}} \ . \tag{1}$$

There are several approximation and pseudo-approximation algorithms for bisection and balanced cuts. In our algorithm, we will employ the algorithm by Arora, Rao and Vazirani [1], and for graphs excluding any fixed graph as a minor (e.g., planar graphs), a slightly stronger algorithm 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 statements that will be used 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 3** (Kolman and Matoušek [6]). *Every graph* G *on* n *vertices contains a subgraph on at least* $2 \cdot n/3$ *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 4.** *For every graph* $G = (V, E)$ *on* n *vertices with maximum degree* Δ *and every subgraph* H *of* G *on* n' *vertices, we have*

$$\mathrm{STC}(G) \geq \frac{\beta(H) \cdot n'}{3 \cdot \Delta} \ . \tag{2}$$

▶ **Corollary 5.** *For every graph* $G = (V, E)$ *with maximum degree* Δ,

$$\mathrm{STC}(G) \geq \frac{2 \cdot hb(G)}{9 \cdot \Delta} \ . \tag{3}$$

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

▶ **Lemma 6.** *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 than $n'/2$ marked vertices. Let v_{i+1} be the neighbour 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 4. Let T be the spanning tree of G with the minimum congestion. By Lemma 6, 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 so that the total number of vertices from H in the smaller part is at least $n'/3$; such a partition can be found greedily. Let $C \subseteq V(H)$ be the vertices from H in the smaller part. Then, by the definition of expansion (1), $e(C, V(H) \setminus C) \geq \beta(H) \cdot n'/3$. 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

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

Proof of Corollary 5. Consider a subgraph H' of G such that $b(H') = hb(G)$. By Lemma 3, there is a subgraph H of H', such that $|V(H)| \geq 2 \cdot |V(H')|/3$ and $\beta(H) \geq b(H')/|V(H')|$. Since H is a subgraph of G, by Lemma 4,

$$\mathrm{STC}(G) \geq \frac{\beta(H) \cdot |V(H)|}{3 \cdot \Delta} \geq \frac{2 \cdot b(H')}{3 \cdot 3 \cdot \Delta} = \frac{2 \cdot hb(G)}{9 \cdot \Delta} \; . \qquad \blacktriangleleft$$

3 Approximation Algorithm

Given a connected graph $G = (V, E)$, we construct the spanning tree of G by the recursive algorithm CongSpanTree called on the graph G. In step 3, one of the algorithms of Theorem 1 is used: for general graphs, the algorithm by Arora, Rao and Vazirani [1], for graphs excluding any fixed graph as a minor, the algorithm by Klein, Protkin and Rao; by $\alpha(n)$ we denote the respective pseudo-approximation factor.

Algorithm 1 CongSpanTree(H)

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

Let τ denote the tree representing the recursive decomposition of G (implicitly) constructed by the algorithm 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 connected components of $G[V_t] \setminus F$ where F is the set of edges of the $\frac{2}{3}$-balanced cut of $G[V_t]$ from step 4; by Theorem 1, $|F| \leq \alpha(n) \cdot b(G[V_t])$. 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 algorithm 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 7.** *Let $t \in \tau$ be a node of the decomposition tree and $t_1, \ldots, t_k$ its children. Then*

$$c(G_t, T_t) \leq \max_i c(G_{t_i}, T_{t_i}) + \alpha(n) \cdot b(G_t) \ . \tag{4}$$

Proof. Let F be the set of edges of the $\frac{2}{3}$-balanced cut of G_t from step 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 $|F| \leq \alpha(n) \cdot b(G[V_t])$, 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 $T_{t_1}, \ldots, T_{t_k}$ 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. ◀

▶ **Lemma 8.** *Let $T = \textsc{CongSpanTree}(G)$. Then*

$$c(G, T) \leq \mathcal{O}(\alpha(n) \cdot \log n) \cdot hb(G) \ . \tag{5}$$

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 $t \in \tau$,

$$c(G_t, T_t) \leq h(t) \cdot \alpha(n) \cdot hb(G) \ . \tag{6}$$

Consider first a node $t \in \tau$ of height zero, that is, a node t that is a leaf. Then both sides of (6) are zero and the inequality holds.

Consider now a node $t \in \tau$ such that for all his children the inequality (6) 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 $b(G_t) \leq hb(G)$ by the definition of hb, by Lemma 7 we get

$$c(G_t, T_t) \leq c(G_{t'}, T_{t'}) + \alpha(n) \cdot hb(G) \ .$$

By the inductive assumption applied on the node t',

$$c(G_{t'}, T_{t'}) \leq h(t') \cdot \alpha(n) \cdot hb(G) \ .$$

Because $h(t') + 1 \leq h(t)$, 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 9.** *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; for graphs excluding any fixed graph as a minor, the approximation is $\mathcal{O}(\Delta \cdot \log n)$.*

Proof. By Corollary 5, for every graph G, $\Omega(hb(G)/\Delta)$ is a lower bound on $\mathrm{STC}(G)$. By Lemma 8, the algorithm CongSpanTree(G) constructs a spanning tree T of congestion at most $\mathcal{O}(\alpha(n) \cdot \log n) \cdot hb(G)$. Combining these two results yields the theorem: $c(G,T) \leq \mathcal{O}(\alpha(n) \cdot \log n) \cdot hb(G) \leq \mathcal{O}(\alpha(n) \cdot \log n \cdot \Delta) \cdot \mathrm{STC}(G)$. Plugging in the bounds on $\alpha(n)$ from Theorem 1 yields the theorem. ◀

4 Open Problems

The inevitable 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 the STC problem 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. doi:10.1145/1502793.1502794.

2 L. Sunil Chandran, Yun Kuen Cheung, and Davis Issac. Spanning tree congestion and computation of generalized Győri-Lovász partition. In *Proc. of 45th International Colloquium on Automata, Languages, and Programming (ICALP)*, volume 107 of *LIPIcs*, pages 32:1–32:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPICS.ICALP.2018.32.

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. doi:10.1145/167088.167261.

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. doi:10.1007/978-3-031-63021-7_38.

6 Petr Kolman and Jiří Matoušek. Crossing number, pair-crossing number, and expansion. *Journal of Combinatorial Theory, Series B*, 92(1):99–113, 2004. doi:10.1016/j.jctb.2003.09.002.

7 Michael Lampis, Valia Mitsou, Edouard Nemery, Yota Otachi, Manolis Vasilakis, and Daniel Vaz. Parameterized spanning tree congestion, 2024. URL: https://arxiv.org/abs/2410.08314, arXiv:2410.08314.

8 Huong Luu and Marek Chrobak. Better hardness results for the minimum spanning tree congestion problem. *Algorithmica*, pages 1–18, 2024. Preliminary version in *Proc. of 17th International Conference and Workshops on Algorithms and Computation (WALCOM)*, 2023. doi:10.1007/s00453-024-01278-5.

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. doi:10.1007/978-3-030-42071-0_12.