# Approximating the Degree-Bounded Minimum Diameter Spanning Tree Problem

J. Könemann[1⋆], A. Levin[2], and A. Sinha[3⋆⋆]

[1] GSIA, Carnegie Mellon University, Pittsburgh PA 15213, USA. jochen@cmu.edu
[2] Department of Statistics and Operations Research, Tel-Aviv University, Tel-Aviv 69978, Israel. levinas@post.tau.ac.il
[3] GSIA, Carnegie Mellon University, Pittsburgh PA 15213, USA. asinha@andrew.cmu.edu

**Abstract.** We consider the problem of finding a minimum diameter spanning tree with maximum node degree $B$ in a complete undirected edge-weighted graph. We prove that the problem is NP-complete, and provide an $O(\sqrt{\log_B n})$-approximation algorithm for the problem. Our algorithm is purely combinatorial, and relies on a combination of *filtering* and *divide and conquer*.

## 1 Introduction

The importance of algorithms for designing efficient networks in today's interconnected world can hardly be overstated. The operative word here is "efficient", and indeed, there are many (often conflicting) ways to measure the efficiency of a network. Suppose a telecommunication company is building a communication network. While budgeting constraints may require the company to minimize total cost, there are also quality of service and technological constraints which may require the network to have low diameter and low degree.

Low diameter is essential to ensure that any pair of nodes can communicate fast. It is also useful to force reliability constraints, as explained in the following (see also [8] and [13]): Assume that an edge $e$ fails with probability $1-p_e$, and that all failures occur independently. Then, the probability that a path $e_1, e_2, \ldots, e_k$ is operational is $p_{e_1} \times p_{e_2} \times \cdots \times p_{e_k}$. Given a certain threshold value for the desired reliability, there is a corresponding parameter $D$ such that the diameter of the network defined by edge length $(|\log p_e|)_{e \in E}$ is required to be at most $D$. Therefore, the reliability constraint is transformed into a diameter constraint.

Degree-constraints appear naturally in graph-theoretic abstractions of communication network design problems. As an example, consider the so called *IP multicast* [4, 5] problem where we would like to disseminate centrally stored information from a server node to a set of client hosts. The standard solution is

⋆ This material is based upon work supported by the National Science Foundation under Grant No. 0105548.

⋆⋆ This material is based upon work supported by the National Science Foundation under Grant No. 0105548 and a Carnegie Bosch Institute Fellowship.

to compute a tree in the given graph that spans the server node and all client nodes. We then send data packets from the root along each of its incident edges in the tree. An internal node forwards incoming information to its descendants in the tree. The degree of a node in this tree is proportional to the amount of work that the node has to do and it is hence natural to aspire to compute spanning trees of low maximum degree (see also [1–3]).

Our work is motivated by precisely these considerations. We proceed by defining our problem.

### 1.1 Problem definition

Formally, we consider the following BOUNDED DEGREE MINIMUM DIAMETER SPANNING TREE PROBLEM (BDST): given an undirected complete graph $G = (V, E)$ whose edges are endowed with a metric length function $\{l_e\}_{e\in E}$ and a parameter $B \geq 2$, we want to find a spanning tree $T$ of $G$ of maximum node-degree at most $B$. At the same time we want to minimize the *diameter* of $T$, i.e. we would like to minimize

$$\Delta(T) := \max_{u,v\in V} \mathtt{dist}_l^T(u, v)$$

where $\mathtt{dist}_l^T(u, v)$ denotes the $l$-length of the unique $u, v$-path in $T$.

Let the *height* of a tree $T$ rooted at node $r$ be the maximum number of edges on any $r, v$-path, where $v$ is a leaf node in $T$ and denote it by $\mathtt{height}(T)$. We also use $n$ and $m$ to denote $|V|$ and $|E|$, respectively.

For $B = 2$, BDST can be approximated within a constant using approximation algorithms for the *Traveling Salesperson* problem. In this paper we consider the case $B \geq 3$.

### 1.2 Results and paper outline

Our main result is an $O(\sqrt{\log_B n})$ approximation algorithm for BDST. The algorithm is described and analyzed in Section 2. There are two main ideas in the algorithm. First, we break up the graph into clusters of low diameter. For each cluster, we compute a balanced $(B-1)$-ary tree. We then compute a global tree over the clusters, and show that the resulting tree has low diameter.

Our algorithm is the first known sub-logarithmic approximation for this problem. An $O(\log_B n)$ approximation is trivial; any complete balanced $(B-1)$-ary spanning tree of the graph will do. We also prove the NP-completeness of BDST in Section 3. We conclude the paper with some open questions.

### 1.3 Related work

Ravi [12] considered the MINIMUM POISE SPANNING TREE PROBLEM defined as follows: given an unweighted graph $G = (V, E)$, we want to find a spanning tree $T = (V, E_T)$ such that the sum of the maximum degree of a node in $T$ and the diameter of $T$ is minimized. In order to provide an approximation algorithm for this problem, he presented an $(O(\log n), O(\log^2 n))$-bicriteria approximation for the BDST problem with a length metric defined by the distances in an unweighted graph $G$, with the restriction that we can only use edges from $G$.

The MINIMUM DIAMETER SPANNING TREE PROBLEM is the following: given an undirected graph $G = (V, E)$ and length function defined over its edge set $\{l_e\}_{e \in E}$, we want to find a spanning tree of $G$ of minimum diameter. This problem is equivalent to finding the shortest paths tree from the absolute 1-center of $G$ (see [9]), and therefore, is solvable in $O(mn + n^2 \log n)$ time.

The MINIMUM DEGREE SPANNING TREE PROBLEM is the following (see [6]): given an undirected graph $G = (V, E)$, we want to find a spanning tree of $G$ whose maximum node-degree is minimized. Fürer and Raghavachari [6] provided a polynomial time algorithm which computes a spanning tree with maximum degree at most $\Delta^* + 1$ where $\Delta^*$ denotes the smallest possible maximum degree of any spanning tree of the input graph. The algorithm in [6] extends also to Steiner trees.

Könemann and Ravi [10, 11] studied the MINIMUM-COST DEGREE BOUNDED SPANNING TREE problem, where in addition to an undirected graph and non-negative edge-costs we are also given a parameter $B_v > 1$ for each node $v \in V$. The objective is to find a minimum-cost spanning tree where the degree of each vertex $v \in V$ is at most $B_v$. The authors show how to compute a tree $T$ where each node $v$ has degree $O(B_v + \log(n))$ and the cost of $T$ is $O(\texttt{opt})$ where $\texttt{opt}$ is the minimum cost of any tree obeying all degree-bounds exactly.

## 2 Algorithm and analysis

### 2.1 Overview

The main idea behind our algorithm is *filtering*. Let $\alpha > 0$ be a threshold, where distances more than $\alpha$ are called *long* and distances less than $\alpha$ are *short*. We partition the node set of $G$ into clusters such that the diameter of each cluster is short, but the number of clusters is also low. We do this by filtering the node set so that we retain one representative node for each cluster, and define an artificial degree bound for this representative node to account for the degree capacity of the entire cluster.

We obtain our performance guarantee from the following two observations. Since the number of clusters is small, any balanced tree which spans the representatives has a small number of long edges. And since each cluster has small diameter, the overhead added to any path by the expansion of the representative nodes into trees spanning the clusters is also small. The rest of this paper shows that such a threshold exists and yields our claimed performance guarantee.

### 2.2 Algorithm

Given an appropriately chosen threshold $\alpha$, the first step of our algorithm is to find representatives $R = \{v_1, \ldots, v_l\} \subseteq V$ and a partition of $V$ into pairwise disjoint sets:

**Algorithm 1** $\texttt{GlobTree}(\{(v_i, B_i)\}_{i=1}^{l})$: Compute a tree $T$ on the nodes $\{v_i\}_{i=1}^{l}$ such that node $v_i$ has node degree at most $B_i$ for all $i$.

Assume $B_1 \geq \ldots \geq B_l$.
$T \leftarrow \emptyset$.
$d_i \leftarrow B_i$ for all $1 \leq i \leq l$
**for** $i = 2$ to $l$ **do**
    Let $1 \leq j \leq i$ be smallest with $d_j > 0$.
    Add edge $(v_j, v_i)$ to $T$.
    $d_j \leftarrow d_j - 1$.
    $d_i \leftarrow d_i - 1$.
**end for**
**return** Tree $T$ with root $v_1$.

$$V = V_1 \cup \ldots \cup V_l \tag{1}$$

such that $v_i \in V_i$ and $\texttt{dist}_l(v_i, u) \leq 3 \cdot \alpha$ for all $1 \leq i \leq l$ and for all $u \in V_i$. Roughly speaking, we then construct a low-degree and low-diameter tree on the nodes of $R$. This tree determines the global structure of our solution. In addition we construct low-diameter degree-$B$-bounded trees for the nodes of each set $V_i$, $1 \leq i \leq l$. We finish by replacing the nodes from $R$ in the global solution by the respective spanning trees.

In the following we assume that we have a guess for the optimum diameter $\Delta$. This is justified since the diameter of an optimum tree is within the interval $[\max_{e \in E} l_e, n \cdot \max_{e \in E} l_e]$ and we can perform a binary search in order to find a proper approximate guess (i.e., a guess within twice the optimum diameter).

We now detail the process of finding the partition from (1). We proceed in iterations: in iteration $1 \leq i \leq l$, we compute the set $V_i$ and its representative $v_i$. For ease of notation, we use $U_i^{\gamma}$ to denote the set of nodes that are at a distance of at least $\gamma$ from the first $i-1$ representatives $\{v_1, \ldots, v_{i-1}\}$. In order to define these sets formally, let $\texttt{cov}_{\gamma}(v, U) = \{u \in U \ : \texttt{dist}_l(v, u) \leq \gamma\}$ be the set of nodes in $U$ that are within a distance of $\gamma$ of vertex $v$ (we also say that $v$ $\gamma$-covers the nodes in $\texttt{cov}_{\gamma}(v, U)$). Then, we let $U_1^{\gamma} = V$ for all $\gamma > 0$. For $i > 1$ we define $U_i^{\gamma} = V \setminus \bigcup_{1 \leq j \leq i-1} \texttt{cov}_{\gamma}(v_j, V)$.

Let $\alpha$ be a given contraction threshold. In iteration $i$, we then pick vertex $v_i \in U_i^{\alpha}$ that $\alpha$-covers most nodes in $U_i^{3\alpha}$, i.e. we let $v_i = \texttt{argmax}_{v \in U_i^{\alpha}} |\texttt{cov}_{\alpha}(v, U_i^{3\alpha})|$, and $V_i = \texttt{cov}_{3\alpha}(v_i, U_i^{3\alpha})$.

The algorithm stops as soon as all nodes in $V$ are within a distance of at most $3\alpha$ from some representative. We assume that this happens after $l$ iterations. We have $U_{l+1}^{3\alpha} = \emptyset$ and $U_i^{3\alpha} \neq \emptyset$ for all $1 \leq i \leq l$.

In order to compute the final tree, we go through two main steps. Assume that we have reordered the sets $\{V_i\}$ such that $|V_1| \geq |V_2| \geq \ldots \geq |V_l|$.

**Global structure** For each $1 \leq i \leq l$, let the degree bound of node $v_i$ be

$$B_i = \begin{cases} |V_i| \cdot (B-2) + 2 & : \quad i = 1 \\ |V_i| \cdot (B-2) + 1 & : \quad \text{otherwise.} \end{cases} \tag{2}$$

**Algorithm 2** $\texttt{BDST}(G, \Delta)$: Compute a degree $B$ tree of diameter no more than $O(\sqrt{\log_B n})\Delta$.

$\alpha \leftarrow \Delta/\sqrt{\log_B n}$.
$i \leftarrow 0$.
$U_1^{3\alpha} \leftarrow V$.
**while** $U_i^{3\alpha} \neq \emptyset$ **do**
  $v_i \leftarrow \texttt{argmax}_{v \in U_i^{\alpha}} |\texttt{cov}_{\alpha}(v, U_i^{3\alpha})|$.
  $V_i \leftarrow \texttt{cov}_{3\alpha}(v_i, U_i^{3\alpha})$.
  $i \leftarrow i + 1$.
**end while**
Reorder $\{v_1, v_2, \ldots, v_l\}$ so that $|V_1| \geq |V_2| \geq \ldots \geq |V_l|$.
Compute $B_i$ as defined in (2).
$T^g \leftarrow$**GlobTree**$(\{(v_i, B_i)\}_{i=1}^{l})$.
**for** $i = 1$ to $l$ **do**
  $T_i \leftarrow$ Tree spanning $V_i$ of degree at most $B$ and minimum height.
  Replace $v_i$ by $T_i$, and distribute the edges in $T^g$ incident on $v_i$ over the nodes of $T_i$ so that the maximum degree of any node in $T_i$ is at most $B$.
**end for**
**return** Resulting tree $T^{\,\texttt{apx}}$.

We then compute a tree $T^g$ = **GlobTree**$(\{(v_i, B_i)\}_{i=1}^{l})$ on the nodes $\{v_1, \ldots, v_l\}$ of low diameter. See Algorithm 1 for the details.

**Local structure** For each $1 \leq i \leq l$ we construct a tree $T_i$ spanning the nodes of $V_i$ of minimum height such that the degree of each node is at most $B$.

Finally, we compute the final tree $T^{\,\texttt{apx}}$ by taking the global tree $T^g$ which is rooted at $v_1$ and replacing each node $v_i$ by the tree $T_i$. We distribute the edges that are incident to $v_i$ in $T^g$ over the nodes of $T_i$ evenly, such that the maximum degree of any node of $T_i$ is as small as possible.

A listing outline of the algorithm is shown in Algorithm 2. Its output is always a tree of degree no more than $B$. We do a binary search over $\Delta$ to obtain a tree of minimum diameter. In the following, we analyze the performance of the algorithm, assuming the correct value for $\Delta$ is fed to Algorithm 2.

### 2.3 Performance ratio

**Theorem 1.** *Suppose that there is a tree $T^*$ with maximum node-degree $B$ and diameter $\Delta$. Then Algorithm $\texttt{BDST}(G, \Delta)$ produces a tree $T^{\,\texttt{apx}}$ with maximum node-degree $B$ and diameter $O(\sqrt{\log_B n} \cdot \Delta)$.*

Theorem 1 is the main result we are trying to prove. We prove it at the end of this section, using a sequence of lemmas which follow.

**Lemma 1.** *The maximum degree of $T^{\,\texttt{apx}}$ is no more than $B$.*

*Proof.* For any $i$, the degree of vertex $v_i$ in the global tree $T^g$ is bounded by $(|V_i| - 2) \cdot B + 2$ for all $1 \leq i \leq l$. Also, tree $T_i$ has $|V_i|$ nodes each with capacity

$B$ and there are exactly $|V_i| - 1$ edges in $T_i$. Hence the total available capacity of the nodes in $V_i$ for edges outside $T_i$ is $|V_i| \cdot (B-2) + 2$. Hence, there is a way of distributing the edges of $T^g$ that are incident to node $v_i$ over all nodes of $T_i$ such that the maximum degree in $T^{\mathtt{apx}}$ is at most $B$.

We now prove that $T^{\mathtt{apx}}$ has diameter $O(\sqrt{\log_B n} \cdot \Delta)$. In the following, we say that an edge $uv \in E$ is *short* if $u, v \in V_i$ for some $1 \leq i \leq l$, and $uv$ is *long* otherwise. Our proof of the diameter bound has two parts: the first part shows that the maximum number of long edges on any root,leaf-path in $T^{\mathtt{apx}}$ is $O(\sqrt{\log_B n})$. The second part shows that there are $O(\log_B n)$ short edges on any root,leaf-path in $T^{\mathtt{apx}}$. This suffices, because the length of any edge in our input graph is at most $\Delta$ and the length of a short edge in $G$ is at most $6\alpha$ (using triangle inequality).

First, we prove that any root,leaf-path contains at most $O(\sqrt{\log_B n})$ long edges. We begin by creating a partition of $V$ using $T^*$'s structure. We root $T^*$ at $v_1^*$ (chosen arbitrarily), and let $V_1^*$ be the set of nodes $u \in V$ such that the unique $(v_1^*, u)$ path in $T^*$ has length at most $\alpha$. We let $S^* = \{v_1^*\}$, and let the set of *uncovered nodes* be $U = V \setminus V_1^*$ initially.

We continue until there are no uncovered nodes remaining. In iteration $i > 1$, let $v_i^* \in U$ be an uncovered node of smallest height in $T^*$ (i.e. $v_i^*$'s parent in $T^*$ is already covered). We then say that a node $u$ is *covered* by $v_i^*$ if $u$ is a descendant of $v_i^*$ and the length of the path from $v_i^*$ to $u$ in $T^*$ is at most $\alpha$. We let $V_i^*$ be the set of nodes in $U$ that are covered by $v_i^*$. We remove $V_i^*$ from $U$ and repeat.

Assume that the final partition has sets $V_1^*, \ldots, V_q^*$ and representatives $v_1^*, \ldots, v_q^*$. Since the subtree $T^*[V_i^*]$ of $T^*$ induced by the nodes of $V_i^*$ is connected, a counting argument shows that the nodes of $V_i^*$ have at most

$$B_i^* = \begin{cases} |V_i^*| \cdot (B-2) + 2 & : \quad i = 1 \\ |V_i^*| \cdot (B-2) + 1 & : \quad \text{otherwise.} \end{cases} \tag{3}$$

children from $V \setminus V_i^*$ in $T^*$. Order the sets such that $|V_1^*| \geq \ldots \geq |V_q^*|$ and let $T$ be the tree produced by $\mathbf{GlobTree}(\{v_i^*\}_{i=1}^q, \{B_i^*\}_{i=1}^q)$.

**Definition 1.** $\{(\overline{v}_i, \overline{V}_i)\}_{i=1}^p$ *is called a* proper collection of $V$ *for a given node set* $V$ *if the following conditions hold:*

1. $\overline{V}_i \subset V$ *and* $\overline{v}_i \in \overline{V}_i$ *for all* $1 \leq i \leq p$.
2. $\overline{V}_i \cap \overline{V}_j = \emptyset$ *for all* $1 \leq i < j \leq p$.
3. $|\overline{V}_1| \geq \ldots \geq |\overline{V}_p|$.
4. $\mathtt{dist}_l(\overline{v}_i, u) \leq \alpha$ *for all* $1 \leq i \leq p$ *and for all* $u \in \overline{V}_i$.

The following lemma is useful in order to prove that the height of the global tree $T^g$ is at most that of $T$.

**Lemma 2.** *Let* $\{(V_i, v_i)\}_{i=1}^l$ *be a partition of* $V$ *together with a corresponding set of representatives created by Steps 1-8 of Algorithm 2. Let* $\{(\overline{v}_i, \overline{V}_i)\}_{i=1}^l$ *be a proper collection of* $G$ *as defined in Definition 1. We then must have*

$$\sum_{i=1}^{j} |V_i| \geq \sum_{i=1}^{j} |\overline{V}_i| \tag{4}$$

*for all* $1 \leq j \leq \max\{l, p\}$.

*Proof.* We prove the lemma by induction over $|V| = n$. For $n = 1$ the lemma is trivially satisfied since in this case $V_1 = \overline{V}_1 = V$. For $n > 1$, assume that the lemma holds for all node sets with at most $n-1$ nodes.

Assume now, for the sake of contradiction, that the lemma does not hold. Let $j$ be the minimum index such that $\sum_{i=1}^{j} |V_i| < \sum_{i=1}^{j} |\overline{V}_i|$. Then, there must exist an index $j_0 \leq j$ such that

$$\overline{V}_{j_0} \not\subseteq \bigcup_{1 \leq i \leq j} V_i$$

and hence $\overline{V}_{j_0} \not\subseteq \mathtt{cov}_{3\alpha}(v_i, V)$ for all $1 \leq i \leq j$. Notice that this implies $\overline{v}_{j_0} \in U_j^{\alpha}$.

Now consider the application of the induction hypothesis for the set of nodes $V' = V \setminus \overline{V}_{j_0}$. Since $\overline{V}_{j_0} \cap \mathtt{cov}_{\alpha}(v_i, V) = \emptyset \; \forall i$, the application of our algorithm with $V'$ yields the exact same set of the first $j-1$ representatives $v_1, v_2, \ldots, v_{j-1}$ and the corresponding subsets $\{V_i \setminus \overline{V}_{j_0}\}_{i=1}^{j-1}$ of $V'$. Note that $\{\overline{V}_i\}_{i=1}^{p} \setminus \{\overline{V}_{j_0}\}$ is a proper collection of $V'$. Therefore, by the induction hypothesis, we conclude that

$$\sum_{i=1}^{j-1} |V_i \setminus \overline{V}_{j_0}| \geq \sum_{i=1}^{j} |\overline{V}_i| - |\overline{V}_{j_0}|. \tag{5}$$

Let us now lower-bound the difference $\sum_{i=1}^{j} |V_i| - \sum_{i=1}^{j-1} |V_i \setminus \overline{V}_{j_0}|$.

This difference can be expressed as the sum of two terms: the size of the set $V_j$ and the increase of the sizes of the biggest $j-1$ sets of our partition. Hence, we obtain

$$\sum_{i=1}^{j} |V_i| \geq \sum_{i=1}^{j-1} |V_i \setminus \overline{V}_{j_0}| + \left| \overline{V}_{j_0} \cap \bigcup_{1 \leq i < j} \mathtt{cov}_{3\alpha}(v_i, V) \right| + |V_j|. \tag{6}$$

Observe that in the $j$-th iteration of our algorithm we could have chosen $\overline{v}_{j_0}$ as a representative instead of $v_j$ since $\overline{v}_{j_0} \in U_j^{\alpha}$. Therefore, we must have that

$$|V_j| = |\mathtt{cov}_{\alpha}(v_j, U_j^{3\alpha})| \geq |\mathtt{cov}_{\alpha}(\overline{v}_{j_0}, U_j^{3\alpha})|.$$

Using (5) together with (6) and noting that

$$\left| \overline{V}_{j_0} \cap \bigcup_{1 \leq i < j} \mathtt{cov}_{3\alpha}(v_i, V) \right| + |\mathtt{cov}_{\alpha}(\overline{v}_{j_0}, U_j^{3\alpha})| \geq |\overline{V}_{j_0}| \tag{7}$$

finally yields

$$\sum_{i=1}^{j} |V_i| \geq \sum_{i=1}^{j} |\overline{V}_i|.$$

This contradicts our assumption, and the lemma follows.

**Corollary 1.** *Let $\{V_i\}_{i=1}^{l}$ be the partition of $V$ generated by Step 9 of Algorithm 2, and $\{V_i^*\}_{i=1}^{l}$ be the partition of $V$ generated from the optimum tree. For all $1 \le j \le \max\{l, p\}$, we have:*

$$\sum_{i=1}^{j} |V_i| \ge \sum_{i=1}^{j} |V_i^*| \tag{8}$$

*Proof.* The statement in (8) clearly holds for the partition $\{V_i'\}_{i=1}^{l}$ generated by steps 1–8 of Algorithm 2, noting that $\{(v_i^*, V_i^*)\}_{i=1}^{q}$ is a proper collection of $V$ as defined in Definition 1.

The corollary follows by observing that reordering the sets of the partition by non-increasing size increases the left hand side of (8) and does not change the right hand side.

We can now prove that the height of the global tree $T^g$ is at most the height of the tree $T$.

**Lemma 3.** *When $T$ is constructed from $T^*$ by* **GlobTree**$(\{v_i^*\}_{i=1}^{q}, \{B_i^*\}_{i=1}^{q})$, *we have* $\texttt{height}(T^g) \le \texttt{height}(T)$.

*Proof.* We say that the *level* of node $v$ of $T$ is the number of edges in the unique path from the root of $T$ to $v$. We now claim that the level of node $v_i$ in $T^g$ is at most the level of node $v_i^*$ in $T$ for all $1 \le i \le l$. We use induction over $i$ to prove the claim.

The claim is clear for $i = 1$. For $i > 1$, assume that `GlobTree` connects node $v_i^*$ to node $v_p^*$ for some $p < i$. It follows from (2),(3) and Corollary 1 that

$$\sum_{j=1}^{p} B_j^* \le \sum_{j=1}^{p} B_j$$

and hence there must exist a $1 \le p' \le p$ such that $d_{p'} > 0$ in `GlobTree` at the time when node $v_i$ is connected. By the induction hypothesis, we know that the level of $v_{p'}$ in $T^g$ is at most that of node $v_{p'}^*$ in $T$. It follows from the definition of `GlobTree` that the level of $v_{p'}^*$ is at most the level of $v_p^*$. Hence, the level of $v_i$ in $T^g$ is at most the level of $v_i^*$ in $T$ and this finishes the induction.

Observe that the height of $T^g$ is equivalent to the level of node $v_l$ in $T^g$, and that the height of $T$ equals the level of $v_q^*$ in $T$. This implies the lemma.

**Lemma 4.** *Let $T^g$ be a tree returned by* `GlobTree`$(\{(v_i, B_i)\}_{i=1}^{l})$. *Then, $T^g$ must be a tree of minimum height among all trees that satisfy the given degree constraints.*

*Proof.* Given a tree $T$, we define the following total order of the nodes in $T$. The order is a breadth-first-search order, with the refinement that the nodes of each level are ordered in non-increasing order of their corresponding sets $V_i$. In particular, the nodes of $T$ are ordered $v_1, v_2, \ldots, v_l$ such that $i < j$ if $\texttt{level}(v_i) <$

$\texttt{level}(v_j)$ or $\texttt{level}(v_i) = \texttt{level}(v_j)$ and $|V_i| \geq |V_j|$. By construction of $T^g$, we have that if $i < j$ in the total order of the nodes in $T^g$, then $|V_i| \geq |V_j|$, regardless of their levels. Moreover, every tree of minimum height for which this holds must have the same height as $\texttt{height}(T^g)$.

Assume for the sake of contradiction that there is a tree $T'$ such that $\deg_{T'}(v_i) \leq B_i$ for all $1 \leq i \leq l$ and $\texttt{height}(T') < \texttt{height}(T^g)$. Let $v'_1, \ldots, v'_l$ be the total order induced by $T'$, as defined above. By the observation in the preceding paragraph, for some $i < j$, we have $|V'_i| < |V'_j|$. We call this an *inversion*, and without loss of generality, assume that $T'$ is a tree with the fewest number of inversions among all trees that satisfy the degree constraints and have height less than $\texttt{height}(T^g)$.

We show that we can reduce the number of inversions in $T'$ without increasing the tree's height. This contradicts the inversion-minimality of $T'$.

Let $\langle v'_i, v'_j \rangle$ be an inversion in $T'$. We swap labels: relabel node $v'_i$ as $v'_j$ and relabel $v'_j$ as $v'_i$. The resulting tree may now violate the degree constraints at node $v'_i$. We counter this by moving a sufficient number of $v'_i$'s children to $v'_j$. This does not increase $\texttt{height}(T')$, and reduces the number of inversions in $T'$, which is a contradiction.

**Lemma 5.** *Any root,leaf-path in $T^{\texttt{apx}}$ has at most $\sqrt{\log_B n}$ long edges.*

*Proof.* Let $d^*$ denote the maximum number of long edges on any root,leaf-path in $T^*$. It follows from Lemma 4 that $\texttt{height}(T) \leq d^*$ and hence, together with Lemma 3, we have that $\texttt{height}(T^g) \leq d^*$.

By the construction of the partition $V_1^*, \ldots, V_q^*$, we know that a root,leaf-path $P$ in $T^*$ that contains $d^*$ long edges must have length at least $\alpha \cdot d^*$. Since $T^*$ has diameter at most $\Delta$ it then follows that $d^* \leq \Delta/\alpha = \sqrt{\log_B n}$ by our choice of $\alpha$.

Lemma 5 bounds from above the contribution of long edges to the diameter of $T^{\texttt{apx}}$. We bound the contribution of short edges in the next lemma. For a root,leaf-path $P$ in $T^{\texttt{apx}}$, let $|P|_s$ denote the number of short edges in $P$.

**Lemma 6.** *Let $P$ be an arbitrary root,leaf-path in $T^{\texttt{apx}}$. Then,*

$$|P|_s = O(\log_B n).$$

*Proof.* Let $P_1$ and $P_2$ be two root,leaf-paths in $T^{\texttt{apx}}$, and let $P_1^g$ and $P_2^g$ be their images in $T^g$, i.e. $P_1^g = \langle v_1^1, \ldots, v_{l_1}^1 \rangle$ and $P_2^g = \langle v_1^2, \ldots, v_{l_2}^2 \rangle$.

We define a relation $\prec$ on two root,leaf-paths as follows. We say that $P_1 \prec P_2$ if $|V_j^1| \geq |V_j^2|$ for all $1 \leq j \leq \max\{l_1, l_2\}$, with $|V_j^i| = 0$ if $V_j^i$ does not exist. By construction of $T^g$, for every two paths $P_1$ and $P_2$ at least one of the following holds: $P_1 \prec P_2$ or $P_2 \prec P_1$. Moreover, if $P_1 \prec P_2$ then $l_2 \leq l_1 \leq l_2 + 1$.

Recall that $T_i$ denotes the *local* tree that spans the nodes of $V_i$. For the purpose of this proof, we assume that all edges of the form $(v_i, v_j)$ in $T^g$ such that $v_i$ is a parent of $v_j$ are attached to leaf nodes in $T_i$. This assumption only increases the number of short edges in root,leaf-paths, and hence is valid.

Consider two paths $P_1$ and $P_2$ such that $P_1 \prec P_2$. Since each $T_i$ is a balanced $(B-1)$-ary tree, we have $|P_2|_s \leq |P_1|_s + |P_1| \leq |P_1|_s + \log_B n$. We also have $|V_i^1| \leq |V_{i-1}^2|$ for $i > 1$, by construction of $T^g$. Therefore, $|P_1|_s \leq |P_2|_s + |P_1| + \mathtt{height}(T_{v_1^1}) \leq |P_2|_s + 2\log_B n$. Hence, there exists a $\gamma$ such that $|P|_s \in [\gamma, \gamma + 2\log_B n]$ for all root,leaf-paths $P$ in $T^{\mathtt{apx}}$.

Observe that on any root,leaf-path $P$ in $T^{\mathtt{apx}}$, all but at most $O(\log_B n)$ of the short edges must be incident to nodes of degree $B$. This follows from the fact that $T^g$ has $O(\log_B n)$ levels. Since there are $n$ nodes in our graph, we must have that $\gamma = O(\log_B n)$. This finishes the proof of the lemma.

We are now ready to prove Theorem 1.

*Proof. (of Theorem 1)* Lemma 1 shows that $T^{\mathtt{apx}}$ has maximum degree $B$.

Long edges in $T^{\mathtt{apx}}$ have length no more than $\Delta$, since the graph has a spanning tree of diameter $\Delta$ and we are assuming we have the correct guess of $\Delta$. Hence, it follows from Lemma 5 that the contribution of long edges to the diameter of $T^{\mathtt{apx}}$ is no more than $2\Delta\sqrt{\log_B n}$.

Short edges in $T^{\mathtt{apx}}$ have length no more than $6\alpha = 6\Delta/\sqrt{\log_B n}$. Lemma 6 bounds the number of short edges in any root,leaf-path, so the total contribution of short edges to the diameter of $T^{\mathtt{apx}}$ is no more than $O(\alpha \log_B n) = O(\Delta\sqrt{\log_B n})$.

All edges are either long or short; this completes the proof of the theorem.

## 3 Hardness

In this section we prove that for any $B \geq 3$ the BDST problem is NP-hard. We prove the NP-completeness of BDST by reducing SET COVER to it.

Suppose we are given an instance of SET COVER $\mathcal{S}$, specified by subsets $\{S_1, \ldots, S_m\}$ of a universe $U = \{u_1, \ldots, u_n\}$, and a number $C$. We want to find out if there is a sub-collection of at most $C$ subsets that covers $U$. We fix a parameter $B \leq C$, and convert $\mathcal{S}$ into a graph $G(\mathcal{S})$ as follows.

The graph has four kinds of nodes. It has one node for every element of $U$. For each set $S_j$, the graph has $\left\lceil \frac{|S_j|}{B-1} \right\rceil$ nodes. Before we describe the other sets of nodes, we describe the edges between these two sets. Every element-set pair $(u_i, S_j)$ such that $u_i \in S_j$ is represented by an edge of length 1 between $u_i$ and one of the nodes representing $S_j$, such that every node that represents a subset has at most $(B-1)$ such adjacent edges.

$G(\mathcal{S})$ has a set of *artificial* nodes, as follows. There is a special artificial node called the root, denoted $r$. For each set $S_j$, we build a tree of degree at most $B$ such that the nodes representing $S_j$ are at the leaves of this tree. The root of the tree is connected to $r$ by a single edge of length 1. All other edges of these trees have length zero.

There are more artificial nodes to *cover* the artificial nodes described above. We build yet another degree $B$ tree, whose leaves include all the nodes in the tree built for each set. The root of this tree is also connected to $r$ with an edge of length 1. In this tree, all edges incident to leaves have length 1, and all other edges have length 0.

We replace the root by a minimum height $(B-1)$-ary tree with $\left\lceil \frac{C+2}{B-1} \right\rceil$ leaves and where all the inner nodes have exactly $(B-1)$ children. Let $r$ be the root of this tree. The edges of the tree has zero length, and for every neighbor of the old root it is now a neighbor of all the leaves of this tree with edges of length 1. We denote the node set of this tree by $ROOT$.

We add another $(B-1)\left\lceil \frac{C+2}{B-1} \right\rceil - C$ *extra nodes* that are connected to the nodes of $ROOT$ with edges of length 2. Note that there are at least 2 extra nodes. The BDST instance is defined by the metric closure of the above distances.

**Lemma 7.** *$\mathcal{S}$ has a set cover of size $C$ if and only if $G(\mathcal{S})$ has a degree $B$ bounded spanning tree with diameter no more than 4.*

*Proof.* Given a set cover of size $B$, we can embed it into $G(\mathcal{S})$ in the obvious way, with the BDST being rooted at $r$. Since every element is covered by the set cover, there is a path of length 2 from $ROOT$ to every element of $U$. The edges that connect the extra nodes provide a path of length 2 to these nodes as well. The artificial tree constructed above provides a path of length 2 to all nodes which do not participate in the set cover. The degree constraints are automatically satisfied by construction. Hence, if $\mathcal{S}$ has a set cover of size $C$, then $G(\mathcal{S})$ has a BDST of diameter 4.

Conversely, suppose $G(\mathcal{S})$ has a BDST of degree $B$ and diameter no more than 4. By construction of $G(\mathcal{S})$, it is impossible for a tree of diameter 3 or less to span it; hence we may assume that the diameter of the tree is exactly 4. In this case, the tree must have a node such that all other nodes in the graph are at distance no more than 2 from it. We call this the *center* of the tree. Since there are at least two extra nodes and the only nodes that are within a distance of at most 2 from the extra nodes are $ROOT$, then the center must be at $ROOT$.

If a BDST centered at $ROOT$ spans the entire graph with paths of length no more than 2, then it must reach all the elements via sets which include those elements, and also it must reach the "extra nodes" directly. The construction of $G(\mathcal{S})$ ensures that these paths induce a valid set cover, and the degree constraint ensures that this set cover has size no more than $C$. Hence if $G(\mathcal{S})$ has a BDST of degree $B$ and diameter 4, then it has a set cover of size no more than $C$.

Since SET COVER is NP-hard [7], so is BDST. Clearly BDST is in NP, since it is easy to check whether a tree has diameter $\Delta$ and degree no more than $B$. We conclude as follows:

**Corollary 2.** *BDST is NP-complete.*

**Corollary 3.** *If $P \neq NP$, then there is no approximation algorithm with performance guarantee of less than $\frac{5}{4}$.*

The reduction is not approximation preserving, so even though SET COVER cannot be approximated to within a logarithmic factor, such a result is not implied for BDST.

## 4 Open questions

In some situations, rather than bounding the diameter of the tree, it is required to bound the *dilation* of every pair of nodes. The dilation of a pair of nodes is defined as the ratio between their distance in the tree and their distance in the original metric. An approximation algorithm for degree bounded minimum dilation spanning trees is still open. This problem is closely related to the well-studied problem of approximating a general metric space by a tree metric.

Our algorithm crucially uses the fact that the input graph is a complete metric. In particular, our algorithm does not work if we are given an (incomplete) input graph and a metric induced by the edge-lengths of its edges (and we are enforced to use only edges from the input graph). Thus, an improvement over the bicriteria $(O(\log n), O(\log^2 n))$ approximation algorithm from [12] for this case is still open.

## References


1. F. Bauer and A. Varma. Degree-Constrained Multicasting in Point-to-Point Networks. In *Proc. of the 14th Annual Joint Conference of the IEEE Computer Communications Societies (INFOCOMM'95)*, 369-376, 1995.
2. Y. Chu, S. G. Rao, S. Seshan, and H. Zhang. Enabling conferencing applications on the internet using an overlay multicast architecture. In *Proceedings of SIGCOMM*, pages 55–68, 2001.
3. W. De Zhong. A copy network with shared buffers for large-scale multicast ATM switching. *IEEE/ACM Transactions on Networking*, 1(2):157-165, 1993.
4. S. E. Deering and D. R. Cheriton. Multicast routing in datagram internetworks and extended LANs. *ACM Transactions on Computer Systems*, 8(2):85, May 1990.
5. S. Deering, D. Estrin, and D. Farinacci. An architecture for wide-area multicast routing. In *Proceedings of SIGCOMM*, 1994.
6. M. Fürer and B. Raghavachari. An NC approximation algorithm for the minimum degree spanning tree problem. In *Proc. of the $28^{th}$ Annual Allerton Conference on Communication, Control and Computing*, 274-281, 1990.
7. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.
8. L. Gouveia. Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research*, 95:178-190, 1996.
9. R. Hassin and A. Tamir. On the minimum diameter spanning tree problem. *Information Processing Letters*, 53:109-111, 1995.
10. J. Könemann and R. Ravi. A matter of degree: Improved approximation algorithms for degree-bounded minimum spanning trees. *SIAM Journal of Computing*, 31(6):1783-1793, 2002.
11. J. Könemann and R. Ravi. Primal-dual algorithms come of age: Approximating MST's with nonuniform degree bounds. To appear in *Proc. of the $35^{th}$ ACM Symposium on Theory of Computing*, 2003.
12. R. Ravi. Rapid rumor ramification: Approximating the minimum broadcast time. In *Proc. of the $35^{th}$ Annual IEEE Symposium on Foundations of Computer Science*, 202-213, 1994.
13. S. Voss. The Steiner tree problem with hop constraints. *Annals of Operations Research*, 86:321-345, 1999.