



# Bounded Degree Group Steiner Tree Problems

Guy Kortsarz<sup>1</sup> and Zeev Nutov<sup>2</sup>(✉)

<sup>1</sup> Rutgers University, Camden, Camden, USA  
guyk@rutgers.edu

<sup>2</sup> The Open University of Israel, Ra'anana, Israel  
nutov@openu.ac.il

**Abstract.** Motivated by some open problems posed in [13], we study three problems that seek a low degree subtree  $T$  of a graph  $G = (V, E)$ . In the MIN-DEGREE GROUP STEINER TREE problem we are given a collection of node subsets (groups), and  $T$  should contain a node from every group. In the MIN-DEGREE STEINER  $k$ -TREE problem we are given a set  $R$  of terminals and an integer  $k$ , and  $T$  should contain  $k$  terminals. In both problems the goal is to minimize the maximum degree of  $T$ .

In the more general DEGREES BOUNDED MIN-COST GROUP STEINER TREE problem, we are also given edge costs and individual degree bounds  $\{b_v : v \in V\}$ . The output tree  $T$  should obey the degree constraints  $\deg_T(v) \leq b_v$  for all  $v \in V$ , and among all such trees we seek one of minimum cost. When the input is a tree, an  $O(\log^2 n)$  approximation for the cost is given in [10]. Our first result generalizes [10] – we give a bicriteria  $(O(\log^2 n), O(\log^2 n))$ -approximation algorithm for DEGREES BOUNDED MIN-COST GROUP STEINER TREE problem on tree inputs. This matches the cost ratio of [10] but also approximates the degrees within  $O(\log^2 n)$ . Our second result shows that if MIN-DEGREE GROUP STEINER TREE admits ratio  $\rho$  then MIN-DEGREE STEINER  $k$ -TREE admits ratio  $\rho \cdot O(\log k)$ . Combined with [12], this implies an  $O(\log^3 n)$ -approximation for MIN-DEGREE STEINER  $k$ -TREE on general graphs, in quasi-polynomial time. Our third result is a polynomial time  $O(\log^3 n)$ -approximation algorithm for MIN-DEGREE GROUP STEINER TREE on bounded treewidth graphs.

## 1 Introduction

We study the following three problems:

### MIN-DEGREE GROUP STEINER TREE

Input: A graph  $G = (V, E)$  and a collection of groups (subsets of  $V$ ).

Output: A subtree  $T$  of  $G$  that contains a node from every group and has minimal maximum degree.

MIN-DEGREE STEINER  $k$ -TREE

Input: A graph  $G = (V, E)$ , a set  $R \subseteq V$  of terminals, and an integer  $k \leq |R|$ .  
 Output: A subtree  $T$  of  $G$  that contains at least  $k$  terminals and has minimal maximum degree.

BOUNDED DEGREES MIN-COST GROUP STEINER TREE

Input: A graph  $G = (V, E)$  with edge costs  $\{c_e : e \in E\}$ , a collection of groups, and degree bounds  $\{b_v : v \in V\}$ .  
 Output: A subtree  $T$  that contains a node from every group and obeys the degree constraints  $\deg_T(v) \leq b_v$  for all  $v \in V$ , and has minimum costs among such subtrees.

Note that in the first two problems the edges have no costs, since the objective is to minimize the maximum degree. The third problem is more general and has both costs and degree bounds.

In the *The Eighth Workshop on Flexible Network Design, Amsterdam, 2016*, Hajiaghayi posed the following open problem:

*Can we obtain a polylogarithmic approximation ratio (in polynomial time) for the BOUNDED DEGREES MIN-COST GROUP STEINER TREE problem?*

In [12] is given a quasi-polynomial time bi-criteria  $(O(\log^4 n), O(\log^2 n))$ -approximation algorithm for the BOUNDED DEGREES MIN-COST DIRECTED STEINER TREE problem.<sup>1</sup> Hence the same holds for the group Steiner problems studied here. Our paper is motivated by the need to provide approximation algorithms that run in polynomial time, which is a standard definition of approximation. Our results are summarized in the following three theorems.

**Theorem 1.** BOUNDED DEGREES MIN-COST GROUP STEINER TREE on tree inputs admits a bicriteria randomized  $(O(\log^2 n), O(\log^2 n))$ -approximation algorithm. Namely, the algorithm computes a tree  $T$  that contains at least one node from every group, has expected cost  $O(\log^2 n)$  times the optimum cost, and with probability at least  $1 - 1/n$  we have  $\deg_T(v) = O(\log^2 n) \cdot b_v$  for all  $v \in V$ .

This result generalizes the one of Garg, Konjevod, and Ravi [10] that gave the same expected ratio  $O(\log^2 n)$  for the cost, but did not consider degree bounds. We note that a bicriteria  $(O(\log^2 n), O(\log^3 n))$  approximation is known to some researchers, but getting ratio  $O(\log^2 n)$  for the degrees requires some care.

**Theorem 2.** If MIN-DEGREE GROUP STEINER TREE admits approximation ratio  $\rho$  then MIN-DEGREE STEINER  $k$ -TREE admits ratio  $\rho \cdot O(\log k)$ . Thus (by [12]) MIN-DEGREE STEINER  $k$ -TREE admits an  $O(\log^3 n)$ -approximation algorithm that runs in quasi polynomial time.

**Theorem 3.** MIN-DEGREE GROUP STEINER TREE on bounded treewidth input graphs admits approximation ratio  $O(\log^3 n)$ .

<sup>1</sup> In private communication, B. Laekhanukit reported that this bi-criteria approximation was recently improved to  $(O(\log^2 n), O(\log^2 n))$ .

**Remark:** MIN-DEGREE STEINER  $k$ -TREE on bounded treewidth graphs admits an exact polynomial time algorithm using dynamic programming (folklore), but is NP-hard even on planar graphs (by a reduction from HAMILTONIAN PATH). However, MIN-DEGREE GROUP STEINER TREE (without costs) is SET-COVER hard even on stars, and thus is  $\Omega(\ln n)$  hard to approximate.

We mention some work on min-costs versions. The best ratio known for MIN-COST GROUP STEINER TREE on tree inputs is  $O(\log^2 n)$  [10]; for a combinatorial algorithm with ratio  $O(\log^{2+\epsilon} n)$  see [4]. This ratio for tree inputs is essentially tight due to the approximation threshold  $\Omega(\log^{2-\epsilon} n)$  of [14]. In the case of general graph inputs, the graph is embedded into a tree distribution with stretch  $O(\log n)$  [8, 10], which gives ratio  $O(\log^3 n)$ . The  $k$ -MST problem admits ratio 2 [9], and this immediately implies ratio 4 for MIN-COST  $k$ -STEINER TREE.

Why the degree bounded versions of these problem are hard to approximate? For many classic problems (without degree bounds), good ratios are achieved using the Iterative Rounding Method, see [16, 18]. This often allows to achieve good bicriteria ratios for the degree bounded versions. However, for many other problems, including the problems we consider, the existing approximation algorithms rely on different methods; e.g., in [18] it is mentioned that the Iterative Rounding Method seems to fail for problems when we need to connect only a specific number of terminals, as in the  $k$ -STEINER TREE problem. Another example is the MIN-COST DIRECTED STEINER TREE problem – the first step in all known approximation algorithms for this problem [3, 11, 17] is the height reduction of Zelikovsky [25], see also [15]. This gives unbounded degrees, as it works on the transitive closure of the graph. There is also a difficulty in dealing with the MIN-DEGREE GROUP STEINER TREE problem, because the known algorithms [4, 10] for the min-cost case first reduce the graph to a random tree [1, 8]. However, this increases the degrees, which means that this technique can not be used.

A logical step is to consider the easiest problems that are open. BOUNDED DEGREES MIN-COST GROUP STEINER TREE on bounded treewidth graphs is one of such problems. However even for this relatively simple problem no polylogarithmic ratio is known, see [12]. The MIN-COST GROUP STEINER TREE problem (without degree bounds) on bounded treewidth graph admits ratio  $O(\log^2 n)$  [2]. The min-degree case, which is the MIN-DEGREE GROUP STEINER TREE problem on bounded treewidth graphs (namely, bounding the degrees with no costs), remained open until our paper.

In the rest of the introduction we discuss some additional motivations for studying min-degree problem without edge costs.

**VLSI Network Design:** The MIN-COST GROUP STEINER TREE problem was motivated by VLSI design. The goal is to connect a set of terminals to a designated root  $r$  by a min-cost tree, where each terminal has a set of multiple ports it can be placed at (ports of two different terminals may intersect). The set of different ports in which a terminal may be placed at, defines a group. The different possible location may be due to rotating, or mirroring, or both. While low cost is highly desirable, the cost is payed once, and later the VLSI

circuit is applied constantly. In many cases low degrees allow faster computations. In [24], a natural VLSI problem is reduced to iteratively solving instances of the MIN-DEGREE STEINER  $k$ -TREE problem. This makes the latency of the VLSI computation low. Low degrees are also important for efficient layout of the VLSI circuit [22]. In the MULTICOMMODITY FACILITY LOCATION UNDER GROUP STEINER ACCESS problem [20], each facility belongs to a group Steiner tree. Short service times requires that such trees have low degrees.

**The  $k$ -Multicast Problem in the Telephone Model:** One of our main motivations for studying the MIN-DEGREE STEINER  $k$ -TREE problem is the TELEPHONE  $k$ -MULTICAST problem [23]. In this problem we are given an undirected graph, a node  $r$ , and a target  $k$  of terminals. We want to send a message from the root  $r$ , to at least  $k$  terminals, under the telephone model. In this model, the nodes that know the message can call at most one neighbor in a round, and send the message to this neighbor. This means that a round is a matching between nodes which know the message to nodes which do not. Note that every broadcasting scheme results in a directed tree in which the parent of a node  $u$ , is the node  $v$ , which sent  $u$  the message. The maximum degree in this multicast tree is a lower bound on the optimum, because at every round we can send the message to at most one child. Hence we need trees with  $k$  terminals and low maximum degree. Also note that the MIN-DEGREE STEINER  $k$ -TREE problem is the minimum degree (without cost) variant of two important and well studied problems: the  $k$ -MST and the  $k$ -STEINER TREE problems. Since these two problems are considered important, so are their minimum degree versions.

**On-Line Degree Bounded Problems:** Recently, MIN-COST/DEGREE GROUP STEINER TREE problems has been studied in the online setting [5–7]. Dehghani et al. [6] showed that it is not possible to approximate both cost and degrees in the on-line model, even when the input graph is a star. Namely, there exists an input demand sequence that forces any algorithm to pay a factor of  $\Omega(n)$  for the cost or the degree violation. However the above papers are able to give polylogarithmic competitive ratios if there are only degree bounds but no costs, similarly to the problems we consider.

## 2 Degrees Bounded Min-Cost Group Steiner Tree Problem on Tree Inputs (Theorem 1)

We will assume that we know a node  $r$  that belongs to some optimal solution. We root the input tree  $T$  at  $r$ . For a group  $S$  let  $\mathcal{A}_S = \{A \subseteq V : r \notin A, S \subseteq A\}$  be the family of cuts that separate the group  $S$  from  $r$ . Let  $\mathcal{A} = \cup_{S \in \mathcal{S}} \mathcal{A}_S$  be the family of all cuts that separate  $r$  from some group. The edges with exactly one endpoint in a set  $A$  are denoted by  $\delta(A)$ . Also recall that  $c_e$  is the cost of an edge  $e$ , and let  $x_e$  be an indicator variable whether  $e$  is included in the solution. The algorithm of Garg, Konjevod, and Ravi [10] uses the following natural LP for the MIN-COST GROUP STEINER TREE problem

$$\begin{aligned} \min & c \cdot x \\ \text{s.t.} & x(\delta(A)) \geq 1 \quad \forall A \in \mathcal{A} \\ & x_e \geq 0 \quad \forall e \in E \end{aligned}$$

The authors of [10] give a special rounding method. For  $e \in E$  let  $p(e)$  be the parent edge of  $e$ ,  $p^2(e) = p(p(e))$  the parent edge of  $p(e)$ , and so on; namely,  $p^i(e)$  is the  $i$ th edge on the path from  $e$  to the root. Add a dummy parent edge  $f$  of the root  $r$  and set  $x_f = 1$ . The algorithm of [10] connects a fraction of groups to the root by choosing every edge  $e \in E$  with probability  $x_e/x_{p(e)}$ . Then the probability that an edge  $e$  of depth  $i$  is connected to the root is

$$\frac{x_e}{x_{p(e)}} \cdot \frac{x_{p(e)}}{x_{p(p(e))}} \cdots \frac{x_{p^{i-1}(e)}}{x_{p^i(e)}} \cdot \frac{x_{p^i(e)}}{x_f} = \frac{x_e}{x_f} = x_e.$$

Thus the expected cost of the edges that are connected to  $r$  is bounded by the value  $c \cdot x$  of the LP solution. The key statement in [10] is:

**Theorem 4** ([10]). *The probability that a specific group is connected the root by the above random process is  $\Omega(1/\log N)$ , where  $N$  is the maximum group size.*

Thus the expected number of iterations required to connect all groups to the root is  $O(\log N \cdot \log k) = O(\log^2 n)$ , where  $k$  is the number of groups, and therefore, this is the expected approximation ratio.

We use the same rounding as [10]. Since we need to bound the degrees of  $n$  nodes, we will require  $\Theta(\log^2 n)$  iterations of the basic procedure. Let  $\delta(v)$  be all the edges that lead from  $v$  to one of its children. Let  $e_v$  be the edge entering  $v$  from his parent. To deal with the degree bounds, we add the following valid constrains to the [10] LP:

$$x(\delta(v)) \leq x_{e_v} \cdot b_v \quad \forall v \in V. \tag{1}$$

To see that these are valid inequalities, consider the characteristic vector  $x$  of an inclusion minimal feasible solution  $T$ . If  $x_{e_v} = 0$  then  $x(\delta(v)) = 0$ , since  $v \notin T$ . If  $x_{e_v} = 1$  then  $x(\delta(v)) \leq b_v = x_{e_v} \cdot b_v$ .

**Corollary 1.** *For every node  $v$  in the tree,  $x(v)/x(e_v) \leq b_v$ .*

The rounding process of [10] gives expected degree  $x(v)/x(e_v) \leq b_v$  in every iteration. Adding degree constraints do not change the expected cost. We analyze the degrees approximation separately using the Chernoff bound (c.f. [19]). If  $X$  is a sum of  $n$  independent Bernoulli variables with mean  $\mu$ , then for any  $\rho > 0$

$$\Pr [X > (1 + \rho)\mu] \leq \left( \frac{e^\rho}{(1 + \rho)^{1+\rho}} \right)^\mu. \tag{2}$$

The degree of  $v$  results by  $O(\log^2 n)$  iterations. In each round we have a Bernoulli sum of all the children of  $v$  that did not reach the root yet. The difficulty here is that the random Bernoulli variables are dependent. For simplicity of the analysis, we bound the degree by  $O(\log^2 n)$  independent Bernoulli

sums, that contains all neighbors of  $v$  in every round. This random variable may be *strictly* larger than the “real” degree. A child  $u$  can contribute more than 1 to the degree. However our random process gives a sum of independent Bernoulli variable which makes the analysis simpler. For a node  $v$ , we have a sum of  $\delta(v) \cdot O(\log^2 n)$  independent Bernoulli variables. The expected degree is  $\tau_v = O(\log^2 n) \cdot x(\delta(v))/x_{e_v}$  (see Corollary 1) and note that  $x(\delta(v))/x_{e_v} \leq b_v$  is implied by the valid inequalities described above. Thus the expected degree is at most  $O(\log^2 n) \cdot b_v$ . We now bound the expectation of  $\tau_v$  by three claims.

*Claim.* If  $\tau_v \geq C \cdot \log n$  for some constant  $C$  then with probability  $1 - 1/n^2$ ,  $\deg(v) = O(\log^2 n) \cdot b_v$ .

*Proof.* We have

$$\Pr[\deg(v) > 2\tau_v] \leq \left(\frac{e}{4}\right)^{C \log n} \leq \frac{1}{n^2}.$$

The last inequality holds for large enough  $C$ . Note that this implies that with probability  $1 - 1/n^2$ ,  $\deg(v) = O(\log^2 n) \cdot b_v$  (see Corollary 1). The ratio  $O(\log^2 n)$  follows. □

We now deal with nodes for which  $1 \leq \tau_v \leq C \cdot \log n$  for some constant  $C$ .

*Claim.* If  $1 \leq \tau_v \leq C \cdot \log n$ , then  $\deg(v) = O(\log^2 n)$  with probability  $\geq 1 - 1/n^2$ .

*Proof.* We know that  $\tau_v \leq C \log n$ . Set  $(1 + \rho) = \log n$ .

First we note that if we prove that  $\Pr[\deg(v) \geq (1 + \rho)\tau_v] \leq 1/n^2$ , then since  $\rho = O(\log n)$  and  $\tau_v = O(\log n)$  we get that with probability  $1 - 1/n^2$  that  $\deg(v) = O(\log^2 n)$ . Since  $b_v \geq 1$  this gives ratio  $O(\log^2 n)$ . We now prove the required inequality.

Since  $\tau_v \geq 1$  we get from the Chernoff bound that:

$$\Pr[\deg(v) \geq (1 + \rho)\tau_v] \leq \frac{e^{\log n}}{(\log n)^{\log n}}.$$

For large enough  $n$  this probability is at most  $1/n^2$ . □

The last case is  $\tau_v < 1$ .

*Claim.* If  $\tau_v < 1$  then with probability  $1 - 1/n^2$ ,  $\deg(v) = O(\log n) \cdot b_v$ .

*Proof.* We set  $(1 + \rho) = \log n/\tau_v$ . Note that if  $\deg(v) \leq (1 + \rho) \cdot \tau_v$  then  $\deg(v) = O(\log n)$ . As  $b_v \geq 1$  the ratio is  $O(\log n)$ . We now bound

$$\Pr[\deg(v) > (1 + \rho) \cdot \tau_v]$$

Consider the term:

$$\frac{e^\rho}{(1 + \rho)^{(1+\rho)}} \leq \frac{e^{\log n/\tau_v}}{(\log n/\tau_v)^{\log n/\tau_v}}.$$

To get the Chernoff bound we should raise to above to the power  $\tau_v$ . Raising this term to  $\tau_v$ , the  $\tau_v$  factor cancels in both exponents. Thus:

$$\Pr[\deg(v) \leq (1 + \rho)\tau_v] \leq \frac{e^{\log n}}{(\log n / \tau_v)^{\log n}}.$$

Since  $\tau_v < 1$  the above is bounded by

$$\frac{e^{\log n}}{(\log n)^{\log n}}.$$

and the above term is bounded by  $1/n^2$  for large enough  $n$ .  $\square$

We got that with probability  $1 - 1/n^2$ , for a given  $v$ ,  $\deg(v) = O(\log^2 n) \cdot b_v$ . By the union bound with probability  $1 - 1/n$  for every  $v$ ,  $\deg(v) = O(\log^2 n) \cdot b_v$ .

### 3 A Relation Between Min-Degree Steiner $k$ -Tree and Min-Degree Group Steiner Tree (Theorem 2)

Assume that MIN-DEGREE GROUP STEINER TREE admits ratio  $\rho$ . We will show that then MIN-DEGREE STEINER  $k$ -TREE admits ratio  $\rho \cdot O(\log k)$ . We first give a simple randomized algorithm with expected ratio  $\rho \cdot O(\log^2 k)$ . Given a MIN-DEGREE STEINER  $k$ -TREE instance  $G, R, k$ , create  $k/(5 \log k)$  bins; the MIN-DEGREE GROUP STEINER TREE instance groups collection is formed by putting uniformly at random, each terminal to a random bin.

**Definition 1.** Fix some optimum solution  $F$  for the MIN-DEGREE STEINER  $k$ -TREE instance with maximum degree  $d^*$  and terminal set  $R^*$ . Terminals in  $R^*$  are called **true terminals**, and a bin is **full** if it contains a true terminal.

**Lemma 1.** With probability at least  $1 - 1/k$  each bin is full.

*Proof.* Consider (only) the  $k$  true terminals in  $R^*$ . For each group  $S$ ,  $|S \cap R^*|$  is a binomial variable with probability  $5 \log k/k$  and  $k$  trials. Thus the expected size of  $|S \cap R^*|$  is  $\mu = 5 \log k$ . By the Chernoff bound:

$$\Pr[|S \cap R^*| \leq (1 - \rho)\mu] \leq \exp(-\rho^2 \mu/2).$$

We plug the right  $\rho$  so that  $(1 - \rho)\mu \leq 1$ . This gives  $\rho$  very close to 1. By the Chernoff bound  $\Pr[S \cap R^* = \emptyset] \leq 1/k^2$ . By the union bound we get that with probability at least  $1 - 1/k$  each bin is full.  $\square$

If we think of a bin as a group, since each group contains a true terminal, the optimum solution  $F$  (restricted to the true terminals) is a solution for the Group Steiner instance, with maximum degree  $d^*$ . Note that we need to cover only  $k/(5 \log k)$  groups which is *not* the MIN-DEGREE GROUP STEINER TREE problem. However, here is a trivial reduction to the MIN-DEGREE GROUP STEINER

TREE problem. Attach a complete binary tree to the root, with  $k - k/5 \log k$  new leaves (we may need to trim the tree to get exactly  $k - k/(5 \log k)$  leaves). Every new leaf belongs to all groups. Thus  $k - k/(5 \log k)$  groups are covered for free with maximum degree 3. This still requires covering  $k/(5 \log k)$  new terminals completing the reduction. The assumed algorithm will find a tree containing at least  $k/(5 \log k)$  terminals, with maximum degree bounded by  $\rho \cdot d^*$ . Taking  $O(\log^2 k)$  iterations gives expected ratio  $O(\log^2 k \cdot \rho)$ .

We now describe a more complicated deterministic reduction with factor loss  $O(\log k)$  in the ratio. Let the terminals be  $0, 1, \dots, q - 1$ ,  $q > k$ , and assume that the above order of the terminals is random. We build  $k$  bins to serve as groups using two point based sampling (see [19]). Let  $p$  be a prime such that  $4k \leq p \leq 8k$ .

1. Choose a number  $a$ , at random, from  $1, 2, \dots, p - 1$ .
2. Choose a number  $b$ , at random, from  $0, 1, \dots, p - 1$ .
3. Terminal  $0 \leq i \leq q - 1$  is assigned bin  $((ai + b) \bmod p) \bmod k$ .

The above construction defines the  $k$  groups. Group  $j$  contains all terminals that reached bin  $j$ .

Any true terminal  $i$  is first matched to a *random number* in  $0, 1, \dots, p$ . The values that will cause item  $i$  to reach bin  $j$  are  $j, j + k, \dots, j + \alpha \cdot k$  for the maximum integer  $\alpha$  such that  $\alpha \cdot k \leq p - 1$ . In the worst case  $j = k - 1$ . Thus the question is how large is  $\alpha$  in the inequality  $(k - 1) + \alpha \cdot k \leq p - 1$ . Choosing  $\alpha = (p - k)/k$  achieves the desired bound. Since  $\alpha$  is an integer, clearly,  $p/k - 2 \leq \alpha < p/k$ . Dividing by  $p$ , implies that the probability that the true terminal  $i$  reaches bin  $j$  is at least  $1/k - 2/p$  and less than  $1/k$ .

Let  $X_{ij}$  be the event that *a true terminal  $i$  reaches bin  $j$* . By the above,  $\Pr(X_{ij}) \geq 1/k - 2/p$ . The events “ $i$  arrived to bin  $j$ ” and “ $i'$  arrived to bin  $j$ ” for  $i \neq i'$  are pairwise independent and so  $\Pr(i \text{ and } i' \text{ arrive to bin } j) \leq 1/k^2$ . We lower bound the probability that  $j$  is full, namely contains a true terminal, using the first two terms of the inclusion exclusion formula

$$\Pr \left[ \bigcup_{i=0}^{k-1} X_{ij} \right] \geq k \cdot \left( \frac{1}{k} - \frac{2}{p} \right) - \frac{\binom{k}{2}}{k^2} \geq \frac{1}{2} - \frac{2}{p}.$$

Thus for every bin, the probability that it's full is at least  $1/3$ . The expected number of full bins is at least  $k/3$ . This gives a solution to the MIN-DEGREE GROUP STEINER TREE problem as follow. Select from every appropriate group (full bin) the true terminal, and connect them using the optimum tree  $F$  (restricted to the  $k/3$  true terminals). Hence there exists a pair  $a, b$  in the sample space for which at least  $k/3$  bins are full and this can be found via the assumed  $\rho$  ratio approximation for the MIN-DEGREE GROUP STEINER TREE. Our sample space of all  $a, b$  pairs has size bounded by  $O(p^2) = O(k^2)$ . Thus we try all  $a, b$  pairs with the goal of covering at least  $k/3$  groups. For every pair  $a, b$ , we apply the assumed  $\rho$  ratio algorithm. For at least one of the  $a, b$  we get (with probability 1) a tree with maximum degree at most  $\rho \cdot d^*$  that contains at least  $k/3$  true terminals. Thus outputting the minimum maximal degree tree over all  $a, b$  choices



guaranties (with probability 1) that the maximum degree in the tree is at most  $\rho \cdot d^*$ , and at least  $k/3$  groups are covered. The penalty is an additional  $O(\log k)$  factor (on top of the  $\rho$  factor).

In [12] the BOUNDED DEGREES GROUP STEINER TREE problem is given a polylogarithmic approximation that runs in quasi polynomial time. The best approximation ratio known is  $O(\log^2 n)$  (this is slightly better than what appears in [12]). The better bound was reported to us by Bundit Laekhanukit, in a private communication. Thus we get:

**Corollary 2.** *The MIN-DEGREE STEINER  $k$ -TREE problem (on general graphs) admits an  $O(\log^3 n)$  approximation in quasi-polynomial time.*

### 4 An $O(\log^3 n)$ Approximation for Min-Degrees Group Steiner Tree on Bounded Treewidth Graphs (Theorem 3)

The high level idea of the algorithm is as follows. We show a new method to reduce the graph into a tree with a loss of an  $O(\log n)$  factor in the degrees. This process is similar to the one often applied on min-cost problems, that pay  $O(\log n)$  penalty for transforming a general graph into a tree, c.f. [10]. Degree problems are often harder, and in our case we also need to pay an additional additive term of  $O(\log n)$  (on the degrees) to get back to a graph solution.

We do not use the formal definition of a treewidth of a graph, but we use the fact that a bounded treewidth graph has a small balanced separator. A subset  $S$  of nodes in a graph  $G$  with  $n$  nodes is an  $\alpha$ -balanced separator (or just a **balanced separator**, if  $\alpha$  is clear from the context) if every connected component in  $H \setminus S$ , if any, has at most  $\alpha n$  nodes. It is known that any graph  $G$  has a  $2/3$ -balanced separator  $S$  of size  $\leq k$ , where  $k$  equals the treewidth of  $G$  plus 1. We can use a linear time algorithm of [21] that finds a  $4/5$ -balanced separator.

We may assume that the input graph  $G$  is connected and has at least  $k$  nodes. We construct an auxiliary rooted tree  $\hat{T}$  by repeatedly removing a balanced separator  $S$  with  $|S| \leq k$  from a large enough connected component  $H$ .

---

**Algorithm 1:** SEPARATOR-TREE( $G = (V, E)$ )

---

```

1  $\mathcal{H} \leftarrow \{G\}, \mathcal{S} \leftarrow \emptyset, \hat{\mathcal{E}} \leftarrow \emptyset$ 
2 while there is unmarked graph  $H \in \mathcal{H}$  with at least  $k + 1$  nodes do
3   | find a balanced separator  $S$  of  $H$  with  $|S| \leq k$  and add  $S$  to  $\mathcal{S}$ 
4   | mark  $H$  and add to  $\hat{\mathcal{E}}$  an edge from  $H$  to  $S$ 
5   | for every connected component  $H_i$  of  $H \setminus S$  do
6   |   | add  $H_i$  to  $\mathcal{H}$  and add to  $\hat{\mathcal{E}}$  an edge from  $S$  to  $H_i$ 
7 return  $\hat{T} = (\mathcal{H} \cup \mathcal{S}, \hat{\mathcal{E}})$ 

```

---

Let  $\mathcal{L}$  be the set of unmarked components in  $\mathcal{H}$  at the end of the algorithm. Every marked component  $H \in \mathcal{H} \setminus \mathcal{L}$  has a unique child, and thus can be shortcut

(or removed, if  $H = G$ ); we denote the resulting tree by  $\mathcal{T} = (\mathcal{S} \cup \mathcal{L}, \mathcal{E})$ . Note that  $\mathcal{S} \cup \mathcal{L}$  is a partition  $V$  into sets of size at most  $k$  each. Also note that  $\mathcal{T}$  has height  $O(\log n)$  (since we used balanced separators) and that every edge of  $G$  either connects nodes in the same part or in parts such that in  $\mathcal{T}$  one is a descendant of the other. Now we define certain trees and paths in  $G$  that are used later.

- (a) For  $S \in \mathcal{S} \cup \mathcal{L}$  the tree  $T^S$  is defined as follows. If  $S \in \mathcal{S}$  then  $T^S$  is an inclusion minimal subtree of  $H$  that contains  $S$ , where  $H$  and  $S$  are in line 3 of the algorithm; and if  $S \in \mathcal{L}$  then  $H = G[S]$  and then  $T^S$  is a spanning tree in  $H$ . Note that  $T^S$  has max-degree  $\leq k$ .
- (b) For an auxiliary edge  $SS' \in \mathcal{E}$  where  $S' = S_i$  is a child of  $S$  in  $\mathcal{T}$ , let  $P^{SS'}$  be the shortest path from  $S$  to  $S'$  in the graph induced by  $H' \cup S$ , where  $H' = H_i$  is the connected component of  $H \setminus S$  that contains  $S' = S_i$  (see lines 5, 6 in the algorithm). Clearly, the path  $P^{SS'}$  has max-degree 2 and all its nodes lie in  $S$  and descendant of  $S'$  in  $\mathcal{T}$ .

Define a new MIN-DEGREE GROUP STEINER TREE instance with input graph being the tree  $\mathcal{T}$ , where each node  $S \in \mathcal{S} \cup \mathcal{L}$  of  $\mathcal{T}$  belongs to all groups of the original instance that intersect  $S$ . The next two lemmas will enable us to finish the proof of Theorem 3. In what follows, for a node  $v \in V$  let  $S_v$  denote the node of  $\mathcal{T}$  that contains  $v$ .

**Lemma 2.** *If the original instance on  $G$  has a solution  $T$  of max-degree  $d$  then the new instance on  $\mathcal{T}$  has a feasible solution  $\mathcal{T}'$  of max-degree  $d \cdot O(k \log n)$ .*

*Proof.* For an edge  $uv \in T$  let  $\mathcal{T}_{uv}$  denote the unique  $S_u S_v$ -path in  $\mathcal{T}$  (possibly  $S_u = S_v$ ). We let  $\mathcal{T}' = \bigcup_{uv \in T} \mathcal{T}_{uv}$ . Since  $T$  is connected,  $\mathcal{T}'$  is a tree; otherwise,

$\mathcal{T}'$  has a partition into two part  $\mathcal{C}$  and  $\mathcal{C}'$  each containing a node from  $\mathcal{T}'$ , such that no edge of  $T$  connects these parts. It is also not hard to verify that  $\mathcal{T}'$  is a feasible solution for the new instance, since each node  $S \in \mathcal{S} \cup \mathcal{L}$  of  $\mathcal{T}$  belongs to all groups of the original instance that intersect  $S$ .

We bound the max-degree of  $\mathcal{T}'$ . Let  $S$  be a node in  $\mathcal{T}'$ . Note that  $\deg_{\mathcal{T}'}(S)$  is at most the number  $\beta$  of branches hanging on  $S$  in  $\mathcal{T}$  that have an edge of  $T$  going from the branch to an ancestor of  $S$  (including  $S$ ) in  $\mathcal{T}$ . The number of ancestors of  $S$  is  $O(\log n)$  and the number of nodes in these ancestors is  $O(k \log n)$ . The  $T$ -degree of each node that lies in an ancestor of  $S$  is  $d$ , hence  $\beta = d \cdot O(k \log n)$ , concluding the proof. □

**Lemma 3.** *There exists a polynomial time algorithm that given a feasible solution  $\mathcal{T}' = (\mathcal{S}' \cup \mathcal{L}', \mathcal{E}')$  for the new instance on  $\mathcal{T}$  of max-degree  $d'$  constructs a feasible solution  $T'$  for the original instance of max-degree  $d' + O(k \log n)$ .*

*Proof.* Let  $G'$  be the graph formed by the trees  $\{T^S : S \in \mathcal{S}' \cup \mathcal{L}'\}$  and the paths  $\{P^{SS'} : SS' \in \mathcal{E}'\}$ . Clearly,  $G'$  is connected, and any spanning tree  $T'$  in  $G'$  is a feasible solution for the original instance on  $G$ . We bound the max-degree of  $G'$ . We view  $\mathcal{T}'$  as a rooted tree, where the root is the node of  $\mathcal{T}'$  that is closest

to the root of  $\mathcal{T}$ . Consider a node  $v$  of  $G$  and the node  $S_v$  of  $\mathcal{T}'$  that contains  $v$ . Let  $\mathcal{P}_v$  be the path from  $S_v$  to the root of  $\mathcal{T}'$ . The height of  $\mathcal{T}$  is  $O(\log n)$ , thus  $|\mathcal{P}_v| = O(\log n)$ . We count the contribution of the trees  $T^S$  and the paths  $P^{SS'}$  to the degree  $\deg_{G'}(v)$  of  $v$  in  $G'$ .

- Any tree  $T^S$  has max-degree  $k$ , and  $v$  may appear in  $T^S$  only if  $S \in \mathcal{P}_v$ ; thus the contribution of the trees  $T^S$  to  $\deg_{G'}(v)$  is  $O(k|\mathcal{P}_v|) = O(k \log n)$ .
- Paths that correspond to edges in  $\mathcal{P}_v$  may contain  $v$  and each of them may contribute  $+2$  to  $\deg_{G'}(v)$ . An edge of  $\mathcal{T}'$  that goes from  $S_v$  to its child may contribute  $+1$  to  $\deg_{G'}(v)$ . Other paths  $P^{SS'}$  have no contribution to  $\deg_{G'}(v)$ , by the construction. Thus the contribution of the paths to  $\deg_{G'}(v)$  is at most  $2|\mathcal{P}_v| + \deg_{\mathcal{T}'}(S_v) - 1 = \deg_{\mathcal{T}'}(S_v) + O(\log n)$ .

Overall, we have  $\deg_{G'}(v) = \deg_{\mathcal{T}'}(S_v) + O(k \log n)$ , and the lemma follows.  $\square$

The Theorem 3 algorithm will find an  $O(\log^2 n)$ -approximate solution  $\mathcal{T}'$  for the new instance  $\mathcal{T}$  using the algorithm from Theorem 1, and then will convert it into a solution  $T'$  for the original instance using the algorithm from Lemma 3. The overall ratio will be the product of  $O(k \log n)$  (Lemma 2) and  $O(\log^2 n)$  (Theorem 1), plus an additive term  $O(k \log n)$  (Lemma 3). Thus the overall ratio is  $O(k \log^3 n) = O(\log^3 n)$ , as claimed in Theorem 3.

**Acknowledgment.** We thank an anonymous referee for many useful comments.

## References

1. Bartal, Y.: Probabilistic approximations of metric spaces and its algorithmic applications. In: FOCS, pp. 184–193 (1996)
2. Chalermsook, P., Das, S., Laekhanukit, B., Vaz, D.: Beyond metric embedding: approximating group Steiner trees on bounded treewidth graphs. In: SODA, pp. 737–751 (2017)
3. Charikar, M., et al.: Approximation algorithms for directed Steiner problems. *J. Algorithms* **33**(1), 73–91 (1999)
4. Chekuri, C., Even, G., Kortsarz, G.: A greedy approximation algorithm for the group Steiner problem. *Discrete Appl. Math.* **154**(1), 15–34 (2006)
5. Dehghani, S., Ehsani, S., Hajiaghayi, M.T., Liaghat, L.: Online degree-bounded Steiner network design. In: SODA, pp. 164–175 (2016)
6. Dehghani, S., Ehsani, S., Hajiaghayi, M.T., Liaghat, V., Räcke, H., Seddighin, S.: Online weighted degree-bounded Steiner networks via novel online mixed packing/covering. In: ICALP, pp. 42:1–42:14 (2016)
7. Dehghani, S., Ehsani, S., Hajiaghayi, M.T., Liaghat, V., Seddighin, S.: Greedy algorithms for online survivable network design. In: ICALP, pp. 152:1–152:14 (2018)
8. Fakcharoenphol, J., Rao, S., Talwar, K.: A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.* **69**(3), 485–497 (2004)
9. Garg, N.: Saving an epsilon: a 2-approximation for the  $k$ -MST problem in graphs. In: STOC, pp. 396–402 (2005)
10. Garg, N., Konjevod, G., Ravi, R.: A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms* **37**(1), 66–84 (2000)

11. Grandoni, F., Laekhanukit, B., Li, S.:  $o(\log^2 k / \log \log k)$ -approximation algorithm for directed Steiner tree: a tight quasi-polynomial-time algorithm. In: STOC, pp. 253–264 (2019)
12. Guo, X., Laekhanukit, B., Li, S., Xian, J.: Tight approximation for variants of directed Steiner tree via state-tree decomposition and linear programming rounding. CoRR, abs/1907.11404 (2019)
13. Hajiaghayi, M.T.: Open problems on bounded-degree network design. In: The Eighth Workshop on Flexible Network Design, Amsterdam (2016)
14. Halperin, E., Krauthgamer, R.: Polylogarithmic inapproximability. In: STOC, pp. 585–594 (2003)
15. Helvig, C.S., Robins, G., Zelikovsky, A.: An improved approximation scheme for the group Steiner problem. *Networks* **37**(1), 8–20 (2001)
16. Jain, K.: A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica* **21**(1), 39–60 (2001)
17. Kortsarz, G., Peleg, D.: Approximating the weight of shallow Steiner trees. *Discrete Appl. Math.* **93**, 265–285 (1999)
18. Lau, L.C., Ravi, R., Singh, M.: *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, New York (2011)
19. Motwani, R., Raghavan, P.: *Randomized Algorithms*. Cambridge University Press, New York (1995)
20. Poplawski, L.J., Rajaraman, R.: Multicommodity facility location under group Steiner access cost. In: SODA, pp. 996–1013 (2011)
21. Reed, B.: Finding approximate separators and computing tree width quickly. In: STOC, pp. 221–228 (1992)
22. Sharma, N., Kaur, M.: Survey of VLSI techniques for power optimization and estimation of optimization. *Int. J. Emerg. Technol. Adv. Eng.* **4**, 351–355 (2014)
23. Slater, P.J., Cockayne, E.J., Hedetniemi, S.T.: Information dissemination in trees. *SIAM J. Comput.* **10**(4), 692–701 (1981)
24. Wang, Y., Hong, X., Jing, T., Yang, Y., Hu, X., Yan, G.: An efficient low-degree RMST algorithm for VLSI/ULSI physical design. In: PATMOS, pp. 442–452 (2004)
25. Zelikovsky, A.: A series of approximation algorithms for the acyclic directed Steiner tree problem. *Algorithmica* **18**(1), 99–110 (1997)