The minimum Degree Group Steiner Problem

Guy Kortsarz[a], Zeev Nutov[b]

[a]*Rutgers University, Camden*
[b]*The Open University of Israel*

Abstract

The DB-GST problem is given an undirected graph $G(V, E)$, and a collection of groups $\mathcal{S} = \{S_i\}_{i=1}^{q}$, $S_i \subseteq V$, find a tree that contains at least one vertex from every group S_i, so that the maximum degree is minimal. This problem was motivated by On-Line algorithms [8], and has applications in VLSI design and fast Broadcasting. In the WDB-GST problem, every vertex v has individual degree bound d_v, and every $e \in E$ has a cost $c(e) > 0$. The goal is, to find a tree that contains at least one terminal from every group, so that for every v, $deg_T(v) \leq d_v$, and among such trees, find the one with minimum cost. We give the first approximation for this problem, an $(O(\log^2 n), O(\log^2 n))$ bicriteria approximation ratio the WDB-GST problem on trees inputs. This implies an $O(\log^2 n)$ approximation for DB-GST on tree inputs. The previously best known ratio for the WDB-GST problem on trees was a bicriteria $(O(\log^2 n), O(\log^3 n))$ (the approximation for the degrees is $O(\log^3 n)$) ratio which is folklore. Getting $O(\log^2 n)$ approximation requires careful case analysis *and was not known*.

Our result for WDB-GST *generalizes* the classic result of [5] that approximated the cost within $O(\log^2 n)$, but did not approximate the degree.

Our main result is an $O(\log^3 n)$ approximation for BD-GST on Bounded Treewidth graphs.

The DB-Steiner k-tree problem is given an undirected graph $G(V, E)$, a collection of terminals $S \subseteq V$, and a number k, find a tree $T(V', E')$ that contains at least k terminals, of minimum maximum degree. We prove that if the DB-GST problem admits a ρ ratio approximation, then the DB-Steiner

Email addresses: guyk@camden.rutgers.edu (Guy Kortsarz), nutov@openu.ac.il (Zeev Nutov)

[1]Partially supported by NSF Grant Number 1910565.

k-tree problem, admits an $O(\log^2 k \cdot \rho)$ expected approximation. We also show that if there are k groups, there exists an algorithm that is able to cover$k/4$ of the groups with minimum maximal degree, then there is a deterministic $O(\log n \cdot \rho)$ approximation for DB-Steiner k-tree problem. Using the work of [7] we derive an an $O(\log^3 n)$ approximation for DB-Steiner k-tree problem on general graphs, that runs in quasi-polynomial time.

1. Introduction

1.1. Degree Bounded Network Design

Degree Bounded Network design problems are central to combinatorial optimization and computer science. In [11], the authors present a very general algorithm for Degree Bounded Network Design problems, based on the iterative rounding techniques [10]. See the book [12] that contains many such results.

1.2. Our problems and results

In [8], namely in the 8'th Workshop on Flexible Network Design, M. Hajiaghayi presented the following open question: *Can we approximate the Min Cost Group Steiner (MC-GST) problem with Degree Bounds?* The problem is open even on Bounded Treewidth Graphs and even if edges have cost 0.

We formalize the open problem of [8]. The WDB-GST problem is:
Input: A graph $G(V, E)$ and a collection of groups $\mathcal{S} = \{S_i\}$, so that $S_i \subset V$, degree bound d_v for every v, and cost $c(e)$ for every $e \in E$.
Required: A tree $T'(V', E')$ containing at least one vertex of every group, so that $deg(v) \leq d_v$ for every v, and among all such trees, find the one of minimum cost.

The DB-GST problem is a special case with edges of zero cost and uniform bound on the degrees:
Input: A graph $G(V, E)$ and a collection of groups $\mathcal{S} = \{S_i\}$, so that $S_i \subset V$.
Required: A tree $T(V', E')$ containing at least one vertex of every group with minimum maximal degree.

The DB-Steiner k-tree problem is:
Input: A graph $G(V, E)$ and a collection of terminals $S \subseteq V$, and a number k.
Required: A tree $T(V, E')$ that containing at least k terminals with minimum maximal degree.

Our main theorem is:

Theorem 1.1. *The DB-GST problem on* Bounded Treewidth Graphs *admits a polynomial time $O(\log^3 n)$ approximation.*

This is the **only known** open question posed by M. Hajiaghayi that is known to have a polynomial time polylog approximation.

The MC-GST is a classic problem which does not have degree bounds but edges have costs. The goal is to find a tree T that contains at least one vertex from every group, with minimum cost. In the elegant paper [5], the authors give an $O(\log^2 n)$ ratio approximation for this problem on tree inputs.

We give the first generalization of [5]. We study the WDB-GST problem on tree inputs.

Theorem 1.2. *There exists a bicriteria $(O(\log^2 n), O(\log^2 n))$ approximation algorithm for the WDB-GST problem on tree inputs. Namely the cost is at most $O(\log^2 n)$ times the optimum cost for degree bounds $\{d_v\}$, so that for every v, $deg_T(v) \leq O(\log^2 n) \cdot d_v$.*

An $(O(\log^2 n), O(\log^3 n))$ bicriteria ratio, namely $deg(v) \leq O(\log^3 n) \cdot d_v$, was known (folklore). Improving the ratio for the degrees requires careful case analysis and *was not known*.

The DB-Steiner k-tree problem is defined as follows:
Input: A graph $G(V, E)$ and a collection S of terminals and an integer k.
Required: A tree $T(V', E')$ containing at least k vertices of S with minimum maximal degree.

We prove the following lemma:

Lemma 1.3. *If the DB-GST problem admits a ρ approximation, the DB-Steiner k-tree problem admits an $O(\log^2 n) \cdot \rho$ expected approximation. If there is ρ ratio algorithm for the problem of covering $k/4$ of the groups with minimum maximum degree, then there is a* deterministic *$O(\log n \cdot \rho)$ approximation for DB-Steiner k-tree problem.*

Building on the work of [7] we obtain the following theorem:

Theorem 1.4. *The DB-Steiner k-tree problem admits an $O(\log^3 n)$ approximation that runs in quasi-polynomial time.*

1.3. Related work

In [7], the authors give a bicriteria polylog$n$ approximation for the WDB-GST problem, however, the algorithm runs in quasi-polynomial time namely in time $n^{\text{polylog}(n)}$.

Theorem 1.5. [7] *The WDB-GST problem admits an $(O(\log^2 n), O(\log^2 n))$ ratio approximation algorithm that runs in quasi-polynomial time.*

In [9] it is proven that unless $P = Quasi(P)$ the MC-GST problem on trees cannot be approximated within $\Omega(\log^{2-\epsilon} n)$ for any constant ϵ. Hence the ratio for GST on tree inputs [5] is almost optimal.

Since we do not know how to approximate WDB-GST even on Bounded Treewidth Graphs, one might consider either removing the costs (as we did) or remove the degree bounds. In [1], the authors study the MC-GST problem on Bounded Treewidth Graphs (namely they remove the degree bounds). They use a novel approach without a reduction to a tree [4], but nevertheless give an $O(\log^2 n)$ ratio for MC-GST on Bounded Treewidth graphs. We too give an $O(\log^3 n)$ ratio for DB-GST without using [4]. Our algorithm is combinatorial and reduces the graph into a tree in a novel way.

1.4. Motivation for problems with no cost but with degree bounds

The motivation of Hajiaghayi was solving the WDB-GST problem in the On-Line setting. Dehghani et al. [1] gave a negative result for WDB-GST: There exists an input demand sequence that forces any On-Line algorithms to pay a factor of $\Omega(n)$ either in the cost or in the degree violation. Hence, in On-Line algorithm you cannot deal with both costs and degree bounds. To date, there are no non-trivial approximation algorithm for WDB-GST either in the online or offline setting, even on Bounded Treewidth Graphs, and even when all the edges have zero-cost. The zero cost case is very similar to DB-GST.

The main motivation for the MC-GST problem comes from VLSI design [5], The goal is to connect a collection $S \subseteq V$ of terminals to a designated root r. Any terminal has multiple ports it can be placed at. The collection of different ports in which a terminal may be placed at, is called a *group*. The different possible location may be due to, say, rotating or mirroring or both. Note that the ports of two different terminals may intersect. In the classic MC-GST problem [5], each link has a cost. The goal is to find a minimum cost subtree, which contains at least one vertex of any group. While low cost is highly desirable, the cost is payed once, and later the VLSI circuit is applied constantly. Low degrees are crucial in VLSI design. Each circuit has several input wires. Low degree implies that the computation of the value of the circuit can be done fast. In [19], a natural VLSI problem is reduced to DB-Steiner k-Tree. This paper iteratively builds trees with low degrees in order to bound from above the latency of the VLSI computation. Low degree are also important for efficient layout of the VLSI circuit (see [14]). It seems to the authors, that for VLSI design, the DB-GST problem is not less important than the MC-GST problem.

The motivation for the DB-Steiner k-tree problem came from the Telephone k-Multicast problem. We are given an unweighted graph $G(V, E)$ a collection S of terminals and a number $k < |S|$. Say that a vertex r knows a message and the goal is to send the message *in the Telephone Model* to k vertices of S. In the Telephone Model at each round a matching between the vertices P which know the message, and the vertices of $S - P$ is chosen. After a round, the vertices in $S - P$ which participated in the matching, join P. The process stops after k terminals know the message, and the goal is to minimize the number of rounds. Note that every multicast scheme defines a tree. The parent of vertex v is the (unique) vertex which sent the message to v. Conversely, given the optimal tree, the best multicast scheme can be found by dynamic programming [18].

Say that we use a tree with maximum degree Δ containing k terminals. The parameter Δ bounds from below the number of rounds required for the multicast. Using trees with low maximum degrees can be a good heuristic for the Telephone k-Multicast problem.

The DB-Steiner k-tree problem is *the Minimum Degree variant* of two classical problems. The k-MST and k-Steiner tree problems (see [6]). As the above problems are important, so are their minimum degree variants.

In Multicommodity Facility Location problem under Group Steiner Access [15], facilities need serve clients. Each facility belongs to a Group Steiner Tree. Short service times require that the Group Steiner trees have low degrees.

Problems that require to cover all terminals with connectivity constraint usually admits a constant ratio [12]. But the DB-Steiner k-tree problem requires to cover k terminals out of many. The authors of [12] explicitly state that the iterative rounding techniques do not work for selecting only k terminals out of many.

1.5. What makes the DB-GST difficult to approximate?

In the approximation algorithm for MC-GST, the first step is reducing the graph into a tree [4]. Unfortunately, this reduction may considerably increases the degrees. Similarly, the reduction of graphs to a random trees, that keeps the expected size of all cuts by at most $O(\log n)$ factor [16], does not preserve the degrees.

1.6. Two trivial observations

The DB-GST problem is $(1-\epsilon)\ln n$ inapproximable on stars unless $P = NP$ [2]. A star can model the *Set Cover* (or Hitting Set) problem (see [2]), since a leaf may belong to many groups. We add a universal vertex r that is connected to every leaf. The chosen edges of r must give a Set Cover of the groups. It is known that the Set Cover problem does not admit a $(1-\epsilon)\cdot\ln n$ approximation, for any constant $\epsilon > 0$, unless $P = NP$ [2]. This implies the same inapproximability for the DB-GST even on stars.

The DB-Steiner k-tree problem admits an exact polynomial solution for the Bounded Treewidth Graphs by the standard dynamic programming algorithm.

Implication for the Covering Group Steiner problem with minimum degree: In the Covering Steiner problem each group has a number d_i and it is required to select at least d_i terminals of group i. In [3], a reduction is given, from the Covering Steiner Problem to Group Steiner problem with the goal of covering only 1/2 of the groups. This gives an immediate approximation (with the same ratio as for Group Steiner) for the Covering Steiner Problem with degree bounds. Thus our algorithm can give the same approximation for the Covering Group Steiner problem with degree bounds and costs on trees. And an $O(\log^3 n)$ ratio for Covering Group Steiner with minimum maximal degree problem on Bounded Treewidth Graphs.

2. Preliminaries

2.1. Notation

Let $E(v)$ be the edges of v. Let t be a tree rooted at r r, which contains a vertex v. We denote by T_v the subtree of T rooted at v. We denote the parent edge of v, by e_v.

Definition 2.1. *Let $Gr(v)$ be all the groups v belongs to.*

Throughout, we ignore the fact that some numbers are not integral. Correcting the claims by rounding up or down is elementary. When we write $\log n$, unless stated otherwise, we mean $\log_2 n$.

2.2. Bounded Treewidth Graphs, definition

A graph $G(V, E)$ has Treewidth k if there is a tree of sets $\{X_i\}$ with each set $X_i \subseteq V$ and so that the following holds

1. $\bigcup_i X_i = V$
2. The size of every X_i is at most $k + 1$.
3. For every edge $e = (u, v)$ there is an X_i so that $u \in X_i$ and $v \in X_i$
4. For any vertex v the set $\{X_j\}$ that contain v are a connected subtree.

2.3. The algorithm of [5]

The algorithm of [5] uses the natural LP for the Minimum Cost Group Steiner problem, which is described below. Let $\delta(S)$, for $S \subseteq V$ be all the edges with exactly one endpoint in S.

$$\text{Minimize} \quad \sum_e x_e \cdot c_e$$
$$\text{Such that}$$

$$\begin{array}{lll} \sum_{e\in\delta(S)} x_e & \geq 1 & \forall S_i \in \mathcal{S}, S \subset V, \text{ such that } r \in S \text{ and } S \cap S_i = \emptyset \\ x_e \geq 0 & & \forall e \end{array}$$

The LP sets fractional capacities on edges, so that every group can send a unit of flow to the root.

A normal randomized rounding process will fail to cover the groups. The authors give a special rounding method. Recall that e_v is the parent edge entering into v.

We assume that the root has a parent edge with capacity 1. For every edge $e \in E(v)$ the edge is added to the solution with probability

$$\frac{x_e}{x_{e_v}}.$$

Definition 2.2. *We say that a vertex* reaches the root *in such rounding, if all the edges from the vertex to the root were selected into the solution. We say that a group reaches the root if a vertex that belongs to the group reaches the root.*

We think of the root as having a parent edge with value 1. The probability that an edge e reaches the root is

$$\frac{x_e}{x_{e_v}} \cdot \frac{x_{e_v}}{x_{e_{p(v)}}} \cdots .$$

This is a telescopic multiplication that equals x_e. Thus, the expected cost of the solution is $\sum_e x_e \cdot c(e) = opt_f$ with opt_f the value of the fractional optimum.

The main lemma of [5] is:

Theorem 2.1. *The probability one of the vertices of a group reaches the root* $1/(c \cdot logn)$ *for some constant* c*.*

Thus, the expected number of iterations required to cover all groups is $O(\log^2 n)$ and therefore, this is the expected ratio.

3. A simple reduction from DB-Steiner k-tree to DB-GST

3.1. Preliminaries

We show that if the DB-GST problem admits a ρ ratio, then the DB-Steiner k-tree problem admits an $O(\log k \cdot \rho)$ ratio. In this section we present a short and simple a $O(\log^2 n) \cdot \rho$ expected approximation ratio for the Minimum Degree k-Tree problem. In the next section we present a more complex algorithm that approximated the maximum degree in the BD-Steiner k-tree within $O(\log k \cdot \rho)$. This follows under a slightly stronger assumption.

3.2. A simple randomized reduction

Theorem 3.1. *If BD-GST admits a* ρ *approximation, then the BD-Steiner* k*-tree admits an* $O(\log^2 k \cdot \rho)$ *expected approximation.*

Proof. We use the following Chernoff inequality (see [13]).

$$Pr\left(\mu \leq (1-\delta)\mu\right) \leq exp(-\delta^2 \mu/2).$$

Create $k/(5 \log k)$ empty bins. Throw uniformly at random, each terminal to a bin. Group S_i are the elements that arrived to bin i.

Notation 3.2. *Let* T^* *be the optimum solution for our BDS* k*-tree instance, and let* d^* *be its maximum degree.*

Definition 3.1. *A terminal is called a* true terminal *if it one of the k which belong to T^*. We say that a group is* full *if it contains a true terminal.*

Claim 3.3. *With probability at least $1 - 1/k$, each group is full*

Proof. The number of true terminals that arrive to a group (bin) is a binomial variable with probability $5 \log k/k$ and k trials. Thus the expected number of true terminals per group is $\mu = 5 \log k$. By the Chernoff bound (see [13])

$$Pr\left(\mu \leq (1-\delta)\mu\right) \leq exp(-\delta^2\mu/2).$$

We plug in the δ so that $(1-\delta)\mu < 1$. This gives δ very close to 1. By the Chernoff bound the probability that a bin is not full is at most $1/k^2$. By the union bound we get that with probability $1 - 1/k$ each bin is full. □

We now relate the value d^*, the max degree in the optimum for the DB-GST problem to the max degree required for covering all $k/(5 \log k)$ groups.

Claim 3.4. *If the $k/(5 \log k)$ of the bins are full, the maximum degree for covering all $k/(5 \log k)$ of the groups is at most d^**

Proof. Let T' be the subtree of T^* induced by the $k/(5 \log k)$ true terminals of the full bins. Tree T' has degree no larger than d^*. We can use T' as a solution for covering $k/(5 \log k)$ groups with maximum degree d^* since the true terminals belong to $k/(5 \log k)$ different groups. Hence the maximum degree for covering at least $k/(5 \log k)$ groups is at most d^*. These may not be true terminals but since every terminal will do, this makes no difference. □

With probability at least $1 - 1/k$, every group is full. By the assumption in Claim 3.4 there is a solution for covering *all* the $k/(5 \cdot \log k)$ groups with maximum degree d^*. Hence we apply the assumed ρ approximation for the DB-GST. After $O(\log^2 k)$ applications of this approximation we get with high probability a tree with k terminals, and maximum degree $O(\log^2 k \cdot \rho \cdot d^*)$ is derived. This proves the theorem. □

4. A deterministic reduction under a slightly stronger assumption

4.1. The new assumption

In this section we need to assume a slightly stronger assumption. That there is a ρ approximation algorithm that finds the best $k/4$ groups to cover, of all the k groups, so that the maximum degree is minimal. Note that the maximum degree for the best $k/4$ groups may be much smaller than the degree for covering all groups, but you need to choose which groups to include, wisely. While the assumption seems stronger it is highly likely that an algorithm that cover all groups, can be used for covering the best $k/4$ groups, with essentially the same ratio. For example if the algorithm for covering all groups uses LP methods, and has ratio ρ, is is immediate to see that the maximum degree for covering $k/4$ groups admits $4 \cdot \rho$ approximation. For example for tree inputs, we can cover the best $k/4$ groups with minimum maximum degree with the same ratio, up to constants. Also the algorithm of [7] is LP based, and easily adapted to cover $k/4$ terminals with the same ratio up to constants.

4.2. A deterministic algorithm

In this section we assume the following slightly stronger assumption.

Theorem 4.1. *If there exists a ρ ratio polynomial time approximation algorithm to cover the best $k/4$ out of k groups, with minimum maximal degree, then there is a deterministic algorithm with ratio $O(\rho \cdot \log n)$ for covering the BD Steiner k-tree problem.*

Proof. Let T^* be the optimum tree for the DB-Steiner k-tree instance. There may be many terminals, but only k of them are true terminals, namely belong to T^*.

For simplicity, we denote the true terminals by $\{0, 1, \ldots, k-1\}$. Our method does not depend on which numbers are used in the representation. We use the bins $0, 1, \ldots, k-1$. Group i is defined by the terminals which arrive to bin i.

We build k groups, $S_0, S_1, \ldots, S_{k-1}$. We use the well known two point based sampling (see [13]). This method is also used in designing universal and perfect hash functions (see [13]). Let p be a prime, such that $8k \leq p \leq 16k$.

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

Fix a group S_j for $0 \leq j \leq k-1$. Group S_j contains the vertices that reach bin j. Any terminal (and thus any true terminal) is first matched to a *random number* in $0, 1, \ldots, p-1$. Fix a true terminal i. Let P_{ij} be the event that a true terminal i belongs to S_j (namely, reaches bin j). The values in $\{0, 1, \ldots, p-1\}$

which will cause item i to reach bin j are $j, j+k, \ldots, j+\alpha\cdot k$, for the maximum integer α such that $j+\alpha\cdot k \leq p-1$. The worst j possible is $k-1$. Thus, the question is how large is α in the inequality $(k-1)+\alpha\cdot k \leq p-1$. Choosing $\alpha=(p-k)/k$ achieves the desired bound. Since α is integral, we get that $p/k-2 \leq \alpha \leq p/k$. Dividing by p, we get that bounded from below and from above by:

$$\frac{1}{k}-\frac{2}{p} \leq Pr(P_{ij}) \leq \frac{1}{k}. \tag{1}$$

Thus, the probability that a true terminal belongs to S_j, is bounded by Inequality (1).

By pairwise independence, if $j \neq i$, $Pr(P_{ij} \cap P_{pj}) \leq 1/k^2$. We bound from below the probability that S_j contains a true terminal. Our bound uses Inequality (1) and the first two terms of the inclusion exclusion formula:

$$Pr(S_j \text{ contains a true terminal}) \geq k\cdot\left(\frac{1}{k}-\frac{2}{p}\right)-\frac{\binom{k}{2}}{k^2} \geq \frac{3}{4}-\frac{1}{2}=\frac{1}{4}$$

We used $p \geq 8k$ and $\binom{k}{2}/k^2 = 1/2-1/(2k) < 1/2$. For every group, the probability that it contains a true terminal is at least $1/4$, and the expected number of groups containing a true terminal is at least $k/4$. at least $k/4$ group with with a true terminal.

Hence we proved:

Corollary 4.2. *The expected number of full bins is at least $k/4$*

Claim 4.3. *There is a deterministic algorithm, that covers the maximum degree within ratio $O(\rho\cdot\log n)$ that covers all groups*

Proof. We now use the fact that our sample space has small size. Namely that the number of a, b is $O(k^2)$ which is polynomial in the input since k is at most the number of terminals and thus at most the number n of vertices. We go over all a, b possible (this can be done in polynomial time). Since the maximum is at least the average, by Claim 4.2, with probability 1, for some pair a, b, $k/4$ of the bins are full. By the Assumption in Theorem 4.1, the algorithm that returns a tree with at least $k/4$ covered groups and so at least $k/4$ terminals with maximum degree $\rho\cdot d^*$. Since we cover $k/4$ terminals and not k, we need to apply this algorithm $O(\log k)$ times to cover all terminals. The ratio $O(\rho\cdot\log n)$ follows. □

This proves Theorem 4.1. □

5. An $(O(\log^2 n), O(\log^2 n))$ bicriteria approximation for WDB-GST on tree inputs

In this section we generalize the main theorem of [5]. We present an $(O(\log^2 n), O(\log^2 n))$ bicriteria approximation algorithm for WDB-GST on trees. Recall that in this problem each vertex has its own degree bound d_v, and edges have costs $\{c(e)\}$. It is required to to find a solution that obeys the degree bounds and contains at least one vertex of every group, and among such solutions find the one with minimum cost.

5.1. Adding degree bounds

We note that in the analysis of [5] every terminal must be a leaf. Otherwise, a vertex of S_i may be a descendant of another vertex of S_i which complicates the proof. Hence, if v is a non leaf that belongs to several groups, for every i so that $v \in S_i$ we attach a leaf x_i to v, that belongs to S_i (v does not belong to S_i after the change). The capacity of (v, x_i) is the same as the capacity of $(p(v), v)$ between v and its parent. This ensures that all the (v, x_i) edges are chosen. The bound on the degree of v in the LP does not contain the edges (v, x_i). If v reaches the root, the edges (x_i, v) are removed and do not contribute to the degree of v. We note that the analysis of [5] changes the x_e values if the x values are 1 over a power. This step can be done in our algorithm as well. The x_e do not grow, and thus the degrees do not grow. In [5] the authors did the natural step of contracting paths with the same x_e values. We cannot do it here since contracting paths increases the degrees. We now explain why it does not matter. Let $p(e)$ be the parent edge of e. In [5] they add the edge to the solution with probability $x_e/x_{(e)}$. If the two values are the same this does not change the probability that a group is connected to the root. The main lemma of [5] claims that their rounding covers $\Omega(/1/\log N)$ groups. This still hold and does not need the height reduction.

If e has parent edge $p(e)$ and $x_e = x_{p(e)}$ then x_e is taken to the solution with probability 1, not changing the probability that a group will be covered.

$$\text{Minimize} \quad \textstyle\sum_e x_e \cdot c_e$$
$$\text{Such that}$$

$$\begin{array}{lll} \sum_e x_e & \geq 1 & \forall S \subset \mathcal{S}, i \text{ such that } r \in S \text{ and } S \cap S_i = \emptyset \\ \sum_{e \in E(v)} x_e & \leq x_{e_v} \cdot d_v & \forall v \\ x_e & \geq 0 & \forall e \end{array}$$

Lemma 5.1. $\sum_{e \in E(v)} x_e \leq x_{e_v} \cdot d_v$ *is a valid inequality*

Proof. If $x_{e_v} = 0$ non of the edges touching v belong to the solution. The second case is $x_{e_v} = 1$. We get: $\sum_{e \in E(v)} x_e \leq d_v \cdot x_{e_v} = d_v$. Hence, all inequalities added are valid inequalities.

□

Corollary 5.2. *For every vertex v*

$$\sum_{e \in E(v)} \frac{x_e}{x_{e_v}} \leq d_v.$$

Definition 5.1. *Let*

$$\mu_v = \sum_{e \in E(v)} \frac{x_e}{x_{e_v}}.$$

We use the same rounding as [5]. We note that the degree constrains do not change the analysis of the algorithm of [5]. Thus, the expected number of iterations is still $O(\log^2 n)$ and so is the expected approximation ratio for the cost. We have to analyze the expected degrees separately.

We use the Chernoff bound [13]. If X is a sum of n independent Bernoulli variables, with mean μ, then:

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

The expected degree of v is $\tau_v = O(\log^2 n) \cdot \mu$. 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. Children that reached the root should not appear in the sum.

For simplicity of analysis, we bound the degree by $O(\log^2 n)$ *independent* Bernoulli sums, which contain all children of v in every round. This is an upper bound on our random variable, because a children that reaches the root can contribute more than 1 to the degree. Hence, this random variable bound the degree from above. However, our random process gives a sum of independent Bernoulli variable which makes the analysis simpler. For a vertex v, we have a sum of $deg(v) \cdot O(\log^2 n)$ independent Bernoulli variables. The expected degree is τ_v (see above).

We now bound the expectation of τ_v by three claims.

Claim 5.3. *If $\tau_v \geq c \cdot \log n$ for a large enough constant c then with probability $1 - 1/n^2$, $deg(v) = O(\log^2 n) \cdot d_v$.*

Proof.

$$Pr\left(deg(v) > 2\tau_v\right) \leq \left(\frac{e}{4}\right)^{c \log n} \leq \frac{1}{n^2} \cdot d_v$$

The last inequality holds for large enough c. Note that this implies that with probability $1 - 1/n^2$, $deg(v) = O(\log^2 n)$. The last inequality, uses the valid inequality $\mu_v \leq d_v$ from the LP. The $O(\log^2 n)$ expected ratio follows,. □

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

Claim 5.4. *In this case, with probability at least $1-1/n^2$, $deg(v) \leq O(\log^2 n)$. Since $d_v \geq 1$ the expected ratio is $O(\log^2 n)$.*

Proof. We know that $\tau_v \leq c \log n$. Set $(1 + \delta) = \log n$.

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

Using the Chernoff bound we get:

$$Pr(deg(v) \geq (1 + \delta)\tau_v) \leq \left(\frac{e^{\log n}}{(\log n)^{\log n}}\right)^{\tau_v}.$$

Since $\tau_v \geq 1$ we get that

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

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

The last case is $\tau_v < 1$.

Claim 5.5. *In case $\tau_v < 1$ with probability $1 - 1/n^2$, $deg(v) = O(\log n) \cdot d_v$.*

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

$$Pr(deg(v) > (1+\delta)\cdot\tau_v)$$

The above is bounded by

$$\left(\frac{e^{\log n/\tau}}{(\log n/\tau)^{\log n/\tau}}\right)^{\tau}.$$

Canceling the τ_v we get a bound:

$$Pr\left(deg(v) \leq (1+\delta)\tau_v\right) \leq \frac{e^{\log n}}{\left(\log n/\tau_v\right)^{\log n}}$$

Since $\tau_v < 1$ we get an upper bound of

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

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. $deg(v) = O(\log n)$. By the union bound with probability $1-1/n$ for every v, $deg(v) \leq O(\log n)$. Since $d_v \geq 1$ the expected ratio is $O(\log n)$.

We now deal with bounded treewidth graphs. Let d^* denote the minimum maximal degree in the instance.

6. An $O(\log^3 n)$ approximation for BD-GST on bounded treewidth graphs

Notation 6.1. *Let d^* be the minimum maximum degree for the Bounded Treewidth instance at hand.*

Definition 6.1. *A set X is called a* separator *if after the removal of X, every connected component in $G - X$ contains at most $c\cdot n$ vertices for some universal constant $c < 1$.*

There are many lemmas for Bounded Treewidth Graphs separators. The following is proved in [17].

Lemma 6.2. *There is a linear time algorithm that finds a size k separator, so that every connected component resulting after the separator is removed is at most $4n/5$.*

Definition 6.2. *Let H be a subgraph and let S be its separators. Let $T_1, \ldots, T_p$ be the connected components in $H - S$. The extended separator of T_i is it's separator S in addition to a vertex v_i that has a neighbor u in S, namely $(u, v_i) \in E$. This makes sure that S has an edge to every one of its children separators in the tree. The extended separator is denoted by $S_E(T_i)$. It's size is $k+1$.*

We now formally present the decomposition of G into a tree of separators with backward edges. Denote the parent separator of S by $p(S)$.

Separator-Tree $G(V, E)$

```
1. Set $\mathcal{C} \leftarrow \{G\}$, $\mathcal{S} \leftarrow \emptyset$
2. **While** there is a connected component $H \in \mathcal{C}$ with at least $k+1$ vertices **do**
    (a) Choose an arbitrary connected component $H \in \mathcal{C}$ so that $|H| > k$.
    (b) Compute an extended separator $S_E(H)$ of $H$ and add it to $\mathcal{S}$.
        /* $\mathcal{S}$ will contain the set of separators of the tree */
    (c) Delete $H$ from $\mathcal{C}$ and add all the connected components of $H - S_H$ into $\mathcal{C}$.
3. Return($\mathcal{S}$)
```

Definition 6.3. *We say that two separators S and S' are* neighbors *if there is a vertex $x \in S$ and a vertex $y \in S'$ that are neighbors in input graph $G(V, E)$. Define the children separators of $S_E(H)$ as all the extended separators of every connected component in $H - S_E(H)$. The edge xy are called extended edges.*

Definition 6.4. The separator graph $\hat{G}(\mathcal{S}, \mathcal{E})$ *is defined as follows.*

1. *The vertices of $\hat{G}$ are the extended separators in $\mathcal{S}$.*
2. *There is an edge in $\mathcal{E}$ between S', S'' if there is are two vertices $u \in S'$ and $v \in S''$ so that $(u, v) \in E$.*

Definition 6.5. *A backward edge is an edge from a descendant separator S' to an ancestor separator S''.*

The following is by construction.

Corollary 6.3. *Every edge in $\hat{G}(\mathcal{S},\mathcal{E})$, that does not belong to $\hat{T}$ is a backward edge.*

The graph $G(S)$ is the graph with S as vertices and with edges, both of its endpoints belonging to S. Note that the graph $G(S)$ may not be connected.

6.1. An overview of our algorithm

1. Build the graph $\hat{G}$ and the tree $\hat{T}$.
2. Add *to the solution* a collection of edges S_C, for every $S \in \hat{G}$, so that $G(S) + S_C$ becomes connected. The penalty on the degrees is an additive $O(\log n)$ term. The maximum degree in the optimum is now $d^* + O(\log n)$.
3. Contract every $S \in \hat{G}$ into a vertex s. This is possible because every separator is connected. Let the resulting graph be G' with a DFS spanning tree T'. The vertex s belongs to the groups $\bigcup_{v \in S} Gr(v)$.
4. Prove that if we replace any edge of $(u,v) \in (G' - T') \cap OPT$ (namely every backward edge in the optimum) by the the unique path between u and v in T', the solution remains feasible and the the penalty on the degree of every vertex is an *additive* $O(\log n) \cdot d^*$, and so the maximum degree in the optimum is now $O(\log n) \cdot d^* + (d^* + O(\log n)) = O(\log n) \cdot d^*$.
5. Run our approximation for DB-GST on T' as input and return the resulting subtree. Because of the $O(\log^2 n)$ ratio for the degrees the final degree is $O(\log^3 n) \cdot d^*$.

7. Connecting all the non-leaf separators with additive $O(\log n)$ ratio

Definition 7.1. *Denote by T_S, the tree of separators rooted by S. containing all descendants of S in $\hat{T}$.*

Let H be the connected graph that S separated. Hence $T_S = H$. The difference is that T_S is posed as a tree of separators. As $T_S = H$ is connected, there is a path in T_S between any two vertices of S. We note that leaves sets may not be connected. But their parents are. The leaves contain at most k vertices.

The way we make $G(S)$ connected for every S is:

Definition 7.2. *Choose an arbitrary vertex $u \in S$, and add all the shortest paths from $S - u$ to u in T_S. These edges are added to our solution.*

Note, that only vertices of T_S are used to connect S. In particular, vertices of ancestors of S are not used.

The edges used to connect S are denoted by S_C.

Lemma 7.1. *$\bigcup_S S_C$ increases the degree of every vertex, by an additive $O(k \cdot \log n) = O(\log n)$ factor.*

Proof. Because of the DFS structure of $\hat{T}$, the degree of a vertex can be affected by at most one separator S per level in $\hat{T}$. We add $|S| - 1 = k$ paths to connect S, and the paths belong to T_S, the tree that S roots. Each path increases the degree of any vertex by at most 2. Since $|S| = k + 1 = O(1)$, the number of paths per S is $O(1)$. Since a vertex is affected by at most one separator per level, the degree added to every vertex v $O(k \cdot \log n) = O(\log n)$ □

Remark: The fact that connecting a separator S uses only edges in T_S is important. Indeed level i may contain many separators. If all of them, use edges of the root separator set, S_R, the degree in S_R may be large.

8. An existential theorem

Theorem 8.1. *Fix an optimum OPT. Since we do not know OPT, we make an existential claim. Consider the following actions: We remove all backward edges used by the optimum and replace each backward edge from some vertex s to a vertex s' by the unique path between s and s' in T'. The number of edges added to a vertex is at most $O(\log n) \cdot d^*$. Hence the optimum has maximum degree of $O(\log n) \cdot d^* + (d^* + O(\log n) = O(\log n) \cdot d^*$.*

Proof. The solution is clearly feasible because if we removed a backward edge (s, s') we added the path between s and s' in the tree T'. Such path exists because the separators are connected and we took extended separators and so we can cross any edge of the tree. Hence we get the same connectivity as in the optimum. Therefore, the new solution is feasible. We show that the number of edges added per vertex is $O(\log n \cdot d^*)$. At every level, a vertex is influenced by only one separator S. The size of each separator S is $k + 1 = O(1)$. Each one of the $O(1)$ vertices inside a separator, is touched by at most d^* edges. Thus, a unique separator S of a vertex v in T_S may add $2 \cdot O(d^*) = O(d^*)$ edges to v. Since the number of levels is $O(\log n)$, the change in the degree of of the vertices may increase by an additive $O(\log n)\cdot d^*$ factor. This proves the existence of a feasible solution which is a subtree of T', with maximum degree $O(\log n) \cdot d^* + (d^* + O(\log n)) = O(\log n)d^*$. $\square$

The above claim implies that we can use T' as an input. We use the approximation for DB-GST on trees. This adds an $O(\log^2 n)$ multiplicative ratio on the degrees. Thus the ratio is $O(\log^3 n)$. This proves our main claim.

8.1. An application

In [7] the authors give a quasi-polynomial $(O(log^2 n), O(\log^2 n))$ approximation for degree bounded Directed Steiner tree on general graph inputs. The DB-GST problem is a special case of the Directed Steiner tree problem. Covering $k/4$ of the groups is easily dealt with, since the solution of [7] is LP based. Using the $O(\log^2 n)$ approximation for $k/4$-DB-GST on general graphs and the approximation of [7], implies a deterministic $O(\log^3 n)$ approximation for DB-Steiner k-tree, which runs in quasi-polynomial time.

9. Recent advance

In [7] the authors designed a polynomial time $O(\log n)$ approximation ratio algorithm, for the DB-GST problem one trees. Note that this ratio is optimal, even on stars. This algorithm can easily handle covering $k/4$ of the groups.

10. Open problems

Is there a polylog(n) ratio approximation, polynomial time algorithm, for WDB-GST on Bounded Treewidth Graphs? Is there a polylog(n) ratio polynomial time algorithm for DB-Steiner k-tree on planar graphs? The DB-Steiner k-tree problem on Planar graphs is NP-hard as the Hamiltonian Path problem is NP-hard on planar graphs.

Acknowledgement

Partially supported by NSF grant number 1540547, and NSF grant number 1910565.

References

[1] P. Chalermsook, S. Das, B. Laekhanukit, and D. Vaz. Beyond metric embedding: Approximating group steiner trees on bounded treewidth graphs. In *SODA 2017*, pages 737–751, 2017.

[2] Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *STOC*, pages 624–633, 2014.

[3] Guy Even, Guy Kortsarz, and Wolfgang Slany. On network design problems: fixed cost flows and the covering Steiner problem. *ACM Transactions on Algorithms*, 1(1):74–101, 2005.

[4] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. System Sci.*, 69(3):485–497, 2004.

[5] N. Garg, G. Konjevod, and R. Ravi. A polylogarithmic approximation algorithm for the group Steiner tree problem. *J. Algorithms*, 37(1):66–84, 2000.

[6] Naveen Garg. Saving an epsilon: a 2-approximation for the k-mst problem in graphs. In *STOC*, pages 396–402, 2005.

[7] Xiangyu Guo, Guy Kortsarz, Bundit Laekhanukit, Shi Li, Daniel Vaz, and Jiayi Xian. On approximation degree-bounded network design problems. *CoRR*, arXiv:1907.11404, 2020.

[8] Mohammad Taghi Hajiaghayi. A list of open problems in bounded degree network design. *The 8'th Workshop on Flexible Network Design*, 2016.

[9] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. In *STOC*, pages 585–594, 2003.

[10] K. Jain. A factor 2 approximation algorithm for the generalized Steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[11] L. Lau, J. Naor, M. Salavatipour, and M. Singh. Survivable network design with degree or order constraints. *SIAM J. Computing*, 39(3):1062–1087, 2009.

[12] Lap-Chi Lau, R. Ravi, and Mohit Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[13] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[14] Manmeet Kaur Nisha Sharma1. Survey of vlsi techniques for power optimization and estimation of optimization. *International Journal of Emerging Technology and Advanced Engineering*, 4, 2014.

[15] Laura J. Poplawski and Rajmohan Rajaraman. Multicommodity facility location under group steiner access cost. In *SODA*, pages 996–1013, 2011.

[16] Harald Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *STOC*, pages 255–264, 2008.

[17] Bruce A. Reed. Finding approximate separators and computing tree width quickly. In *STOC*, pages 221–228, 1992.

[18] Peter J. Slater, Ernest J. Cockayne, and Stephen T. Hedetniemi. Information dissemination in trees. *SIAM J. Comput.*, 10(4):692–701, 1981.

[19] Yin Wang, Xianlong Hong, Tong Jing, Yang Yang, Xiaodong Hu, and Guiying Yan. An efficient low-degree RMST algorithm for VLSI/ULSI physical design. In *PATMOS*, pages 442–452, 2004.