Improved Approximation Algorithm for Steiner k-Forest with Nearly Uniform Weights

Michael Dinitz[1]
Johns Hopkins University
mdinitz@cs.jhu.edu
and
Guy Kortsarz[2]
Rutgers University, Camden
guyk@crab.rutgers.edu
and
Zeev Nutov
The Open University of Israel
nutov@openu.ac.il

In the Steiner k-Forest problem we are given an edge weighted graph, a collection D of node pairs, and an integer $k \leq |D|$. The goal is to find a min-weight subgraph that connects at least k pairs. The best known ratio for this problem is $\min\{O(\sqrt{n}), O(\sqrt{k})\}$ [Gupta et al. 2010]. In [Gupta et al. 2010] it is also shown that ratio ρ for Steiner k-Forest implies ratio $O(\rho \cdot \log^2 n)$ for the related Dial-a-Ride problem. The only other algorithm known for Dial-a-Ride, besides the one resulting from [Gupta et al. 2010], has ratio $O(\sqrt{n})$ [Charikar and Raghavachari 1998].

We obtain approximation ratio $n^{0.448}$ for Steiner k-Forest and Dial-a-Ride with unit weights, breaking the $O(\sqrt{n})$ approximation barrier for this natural case. We also show that if the maximum edge-weight is $O(n^{\epsilon})$ then one can achieve ratio $O(n^{(1+\epsilon)\cdot 0.448})$, which is less than $\sqrt{n}$ if ϵ is small enough. The improvement for Dial-a-Ride is the first progress for this problem in 15 years. To prove our main result we consider the following generalization of the Minimum k-Edge Subgraph (Mk-ES) problem, which we call Min-Cost ℓ-Edge-Profit Subgraph (MCℓ-EPS): Given a graph $G = (V, E)$ with edge-profits $p = \{p_e : e \in E\}$ and node-costs $c = \{c_v : v \in V\}$, and a lower profit bound ℓ, find a minimum node-cost subgraph of G of edge-profit at least ℓ. The Mk-ES problem is a special case of MCℓ-EPS with unit node costs and unit edge profits. The currently best known ratio for Mk-ES is $n^{3-2\sqrt{2}+\epsilon}$ [Chlamtac et al. 2012]. We extend this ratio to MCℓ-EPS for general node costs and profits bounded by a polynomial in n, which may be of independent interest.

Categories and Subject Descriptors: F.2.2 [**Nonnumerical Algorithms and Problems**]: Computations on discrete structures; G.2.2 [**Discrete Mathematics**]: Graph Algorithms

General Terms: Network Design, Steiner k-Forest, Approximation Algorithms

1. INTRODUCTION

1.1 Problems considered and previous work

We consider the following problem, first studied by [Hajiaghayi and Jain 2006]:

Steiner k-Forest
Instance: A graph $G = (V, E)$ with integral edge-weights $w = \{w_e : e \in E\}$, a collection D of node pairs (called the *demands*), and an integer $k \leq |D|$.
Objective: Find a minimum weight subgraph of G that connects at least k pairs from D.

Let Q denote the union of the pairs in D; the nodes in Q are called *terminals*, and the graph (Q, D) is called the *demand graph*.

Steiner k-Forest generalizes several known problems, among them the following:

- When $k = |D|$, namely, if we need to connect all pairs in D, we get the Steiner Forest problem, which admits a 2-approximation algorithm [Agrawal et al. 1995].
- When the demand graph is connected and $k = |D|$, we get the Steiner Tree problem, which admits a $(\ln 4 + \epsilon)$-approximation scheme [Byrka et al. 2013].
- When the demand graph is connected and $Q = V$ we get the k-MST problem, which admits a 2-approximation algorithm [Garg 2005].
- Consider the Minimum k-Edge Subgraph (Mk-ES) problem (the minimization version of the Densest k-Subgraph problem): Given a simple graph G and an integer k, find a subgraph of G with k edges and minimum number of nodes. Mk-ES is essentially equivalent to a particular case of Steiner k-Forest with unit weights when the input graph is a star. Given an instance $G = (V, E), k$ of Mk-ES, obtain an instance $G' = (V', E'), D, k$ of Steiner k-Forest with unit weights as follows. $G' = (V', E')$ is a star with center s and leaf set V, where s is a new node; the demands are $D = \{\{u, v\} : uv \in E\}$, namely, we create a demand for every edge in E. Then Steiner k-Forest on G' is exactly Mk-ES on G, see [Hajiaghayi and Jain 2006]. For simple graphs, Mk-ES admits an $\tilde{O}(n^{3-2\sqrt{2}+\epsilon})$-approximation scheme [Chlamtac et al. 2012] (note that $3 - 2\sqrt{2} < 0.1716$).

Another problem closely related to Steiner k-Forest is the following.

Dial-a-Ride
Instance: A graph $G = (V, E)$ with integral edge-lengths $w = \{w_e : e \in E\}$, a collection of items with a source and a destination each, and an integer $k \leq |D|$.
Objective: Move every item from its source to its destination using a vehicle that can carry at most k items, minimizing total travel length.

This version is called the *non preemptive* Dial-a-Ride, in the sense that we can not leave an item in a node that is not its destination for picking it up later.

Steiner k-Forest hardness of approximation. We note that a strong lower bound on approximating Steiner k-Forest is not known. Indeed, it is only known that the problem is APX-hard (as the Steiner Tree problem is, see [Agrawal et al. 1995]).

The Densest k-Subgraph problem is as follows: Given a graph and an integer k, find a subgraph of G with k nodes and maximum number of edges. It is known that if Mk-ES admits approximation ratio ρ then Densest k-Subgraph admits ratio ρ^2. The first approximation for Densest k-Subgraph was $O(n^{2/5})$ given in [Kortsarz and Peleg 1993], and it was improved to $O(n^{1/3})$ in [Feige et al. 2001]. Today, the best known ratio for the problem is essentially $O(n^{1/4})$ [Bhaskara et al. 2010]. Since

for 22 years the best known ratio for Densest k-Subgraph remained polynomial, and because of the large effort invested in improving the ratio for the problem, it seems likely that Densest k-Subgraph admits no better than polynomial ratio. This indicates that it is unlikely that Mk-ES admits a better than polynomial approximation ratio (see above). In turn, this implies that it is unlikely that Steiner k-Forest admits better than polynomial ratio.

Known approximation for our problems. The best known ratio for Steiner k-Forest is $\min\{O(\sqrt{n}), O(\sqrt{k})\}$ [Gupta et al. 2010], even for the case of unit weights. For $k = O(n)$ this ratio almost coincides with the best known ratio $k^{1/2+\epsilon}$ for the directed version of the problem [Feldman et al. 2012], even though undirected network design problems usually have much better approximation ratios.

The Dial-a-Ride problem admits an $O(\sqrt{n})$ ratio by [Charikar and Raghavachari 1998]. In [Gupta et al. 2010] it is also shown that ratio ρ for Steiner k-Forest implies ratio $O(\rho \cdot \log^2 n)$ for the related Dial-a-Ride problem.

1.2 Our results

We prove the following.

THEOREM 1.1. Steiner k-Forest *with unit weights admits approximation ratio* $n^{0.448}$.

To prove Theorem 1.1 we consider the following generalization of the Mk-ES problem, which we call Min-Cost ℓ-Edge-Profit Subgraph, or MCℓ-EPS for short.

Min-Cost ℓ-Edge-Profit Subgraph (MCℓ-EPS)
Instance: A multigraph $G = (V, E)$ with edge-profits $p = \{p_e : e \in E\}$ and node-costs $c = \{c_v : v \in V\}$, and a profit lower bound ℓ.
Objective: Find a minimum node-cost subgraph of G of profit at least ℓ.

MCℓ-EPS with a simple graph G, and with unit node costs and unit edge profits (and $\ell = k$), is the Mk-ES problem. As was mentioned, the currently best known ratio for Mk-ES is $n^{3-2\sqrt{2}+\epsilon}$ [Chlamtac et al. 2012]. We extend this ratio to MCℓ-EPS by modifying the algorithm of [Chlamtac et al. 2012] (for simple graphs with unit node costs and unit edge profits) to handle multigraphs with general node costs and profits bounded by a polynomial in n. The node costs can be exponential in n or beyond, but when the edge profits are exponential in n we can only give a bicriteria approximation: the algorithm will find a subgraph of node cost at most $n^{3-2\sqrt{2}+\epsilon}$ times the optimum and edge profit at least $\ell(1 - 1/\text{poly}(n))$ (rather than the desired profit of ℓ), where poly(n) is any polynomial in n. However, in the application to Steiner k-Forest, edge profits are at most n^2, and hence we just need to consider multigraphs with node costs and unit edge profits, and do not have to resort to the bicriteria approximation.

THEOREM 1.2. MCℓ-EPS *with edge profits that are at most polynomial in n (but with arbitrary node costs) admits an $n^{3-2\sqrt{2}+\epsilon}$-approximation scheme.*

The following theorem establishes a relationship between Steiner k-Forest and MCℓ-EPS, and it implies Theorem 1.1 by substituting the value of $\gamma = 3 - 2\sqrt{2} + \epsilon$ from Theorem 1.2; note that then $\frac{1}{3}(1 + 2\gamma) = \frac{1}{3}(7 - 4\sqrt{2} + \epsilon) < 0.4478 + \epsilon$.

THEOREM 1.3. *If* MCℓ-EPS *(with edge profits and node costs at most polynomial in n) admits approximation ratio $\rho = n^{\gamma}$ with $0 \leq \gamma \leq 1/4$, then* Steiner k-Forest *with unit weights admits approximation ratio $\tilde{O}\left(n^{1/3+2\gamma/3}\right)$.*

This theorem is our main contribution.

In [Gupta et al. 2010], the Dial-a-Ride problem is approximated using the approximation for Steiner k-Forest as a *black box*. Thus if the Dial-a-Ride problem has unit (or uniform) edge lengths, the black box can be replaced by our approximation for Steiner k-Forest. This implies the same approximation (up to polylog(n) factors) for Dial-a-Ride with unit edge lengths.

COROLLARY 1.4. Dial-a-Ride *with unit edge lengths admits approximation ratio $n^{0.448}$.*

Breaking the $O(\sqrt{n})$ ratio when the maximum weight is small. Many times in practice, the largest weight is not arbitrarily large. Studying problems for low weights is a well established paradigm in approximation algorithms. For example, there are several papers that deal with the TSP problem with weights in $\{1, 2\}$ (c.f. [Berman and Karpinski 2006]). In several applications its reasonable to assume that the weights are polylogarithmic in n. Our algorithm can deal with these cases, and more generally, when the maximum weight is n^{ϵ} (and the weights are integral) our ratio is $n^{1(+\epsilon)\cdot 0.448}$ (which is less than $\sqrt{n}$ for small ϵ). In several applications the maximum weight and the minimum weight are not so far apart. Given a graph with arbitrary weight so that the maximum cost over minimum cost is at most $n^{1-\epsilon}$ by a standard argument the weights can be transformed so that they are integral and belong to $[1, \lceil n^{\epsilon} \rceil]$ with negligible loss.

Hence, we now show that if the maximum weight of an edge is n^{ϵ} for a small enough ϵ, then improving the $O(\sqrt{n})$ ratio for Steiner k-Forest and Dial-a-Ride is still possible. As a first step, we can replace the input graph by an $O(\log n)$-stretch graph spanner H containing $O(n)$ edges (see [Althöfer et al. 1993]). Then any solution to the original input can be changed into a solution on H with weight at most $O(\log n)$ times the original weight (any edge e of the original solution which is not in H can be replaced by a path of total weight at most $O(\log n)$ times the weight of e). Now let E' be the edge set of H, and suppose that the average weight of the edges $w(E')/O(n)$ in the spanner H is at most n^{ϵ}. Then if we replace each edge e by a path of $w(e)$ edges each of weight 1, we only increase the size of the graph by an n^{ϵ} factor, and hence we get ratio $O(n^{(1+\epsilon)0.448})$. Clearly, the average weight in the spanner is at most n^{ϵ} if *every* weight is at most n^{ϵ}, and thus for small enough ϵ we get a bound of $o(\sqrt{n})$ in the case of the maximum weight is n^{ϵ}.

2. THE ALGORITHM (PROOF OF THEOREM 1.3)

2.1 Preliminaries

Some of our intermediate statements will be valid for the weighted version of the problem; we will mention which formal statements are valid for unit weights only. We will sometimes treat weights as lengths. Namely, the distance between a pair of nodes u and v in a weighted graph is the minimum weight of a uv-path in the graph. We denote by $\text{dist}_G(u, v)$ the minimum weight/length of a uv-path in G.

Fix some optimal solution J and a set D_J of k demands connected by J. We will use the following notation.

- τ is the optimum solution value, namely, the size (or weight) of J.
- $q = |Q_J|$ is the number of nodes in the union Q_J of the pairs in D_J.
- $\rho = n^\gamma$ denotes the best known ratio for MCℓ-EPS.

In what follows, we may "guess" the right values of τ and q, by applying any of our algorithms for all possible values of $\tau = 1, \ldots, |E|$ and $q = 1, \ldots, n$, and among the edge sets computed return the best one (a similar method works for the weighted case as well, with $\tau \in \{2^i : i = 0, \ldots, \lfloor \log_2 w(E) \rfloor\}$ being an estimate for an optimal solution value up to a factor of 2). While we can not know a priori what the correct values are, we know that since we apply exhaustive search there is a run of the algorithm with the correct two values. We prove that when the algorithm runs with the correct values, the claimed ratio holds. Hence from now on we use τ and q in the analysis, namely, we only analyze the run of the algorithm with the correct value for the parameters. Consequently, we may assume that $\mathrm{dist}_G(u, v) \leq \tau$ for every $\{u, v\} \in D$; pairs $\{u, v\} \in D$ with $\mathrm{dist}_G(u, v) > \tau$ are not connected by J and can be discarded in advance.

In what follows, note that $k \leq q(q-1)/2 < q^2$ ($k = q(q-1)/2$ may hold if D_J is a clique on Q_J) and that in the case of unit weights $q = |Q_J| \leq 2|J| \leq 2\tau$, since every node in Q_J is an endnode of some edge in J ($|Q_J| = 2|J|$ may hold if J is a matching on Q_J).

We have one very easy case $\tau\sqrt{q} > n/\rho$, which is resolved in the following statement.

LEMMA 2.1. *For any $0 \leq \gamma \leq 1/4$, the following holds: if $\tau\sqrt{q} > n^{1-\gamma}$, then* Steiner k-Forest *with unit weights admits approximation ratio $O\left(n^{1/3+2\gamma/3}\right)$.*

PROOF. Let $\theta = \frac{1}{6} - \frac{2}{3}\gamma$. Since $\gamma \leq \frac{1}{4}$, $\theta \geq 0$. Any maximal forest of G is a feasible solution that has at most $n-1$ edges. Thus if $2\tau > n^{1/2+\theta}$ then simply returning a maximal forest gives approximation ratio

$$\frac{n-1}{\tau} \leq 2\frac{n}{n^{1/2+\theta}} < 2n^{1/2-\theta} = 2n^{1/3+2\gamma/3} .$$

Otherwise, if $\tau \leq n^{1/2+\theta}$ then $q \leq \tau \leq n^{1/2+\theta}$, so $\tau\sqrt{q} \leq n^{1/2+\theta}n^{1/4+\theta/2} = n^{3/4+3\theta/2} = n^{1-\gamma}$; this contradicts the assumption $\tau\sqrt{q} > n^{1-\gamma}$. □

From now and on we will focus on the complementary "hard case" $\tau\sqrt{q} \leq n/\rho$. We use $\tilde{O}$ and $\tilde{\Omega}$ to suppress polylogarithmic terms. We first show that in order to get ratio $\tilde{O}(f)$ for Steiner k-Forest it is sufficient to be able to find a partial solution of size (or weight) $\tau \cdot \tilde{O}(f)$ that connects $\tilde{\Omega}(k)$ demands; then we can just iterate until we connect at least k demands. This type of reduction is essentially standard, but we provide a proof-sketch for completeness of exposition.

LEMMA 2.2. *Suppose that* Steiner k-Forest *admits a bicriteria approximation algorithm that returns a subgraph of weight $\leq f \cdot \tau$ that connects at least k/p demands, where $1 < p < k$. Then* Steiner k-Forest *admits approximation ratio $f \cdot (\lfloor \ln k / \ln \alpha \rfloor + 1)$, where $\alpha = 1 + \frac{1}{p-1}$. In particular, if $k = n^\epsilon$ for some $\epsilon > 0$ and $p = \mathrm{polylog}(n)$, then* Steiner k-Forest *admits a $\tilde{O}(f)$-approximation algorithm.*

PROOF. We run the bicriteria algorithm iteratively, as follows. Let k_i denote the residual demand (the number of pairs we still need to connect) at the beginning of iteration i, where $k_1 = k$. While $k_i \geq 1$, we run the bicriteria algorithm, remove from D the pairs connected in the current iteration, and set $k_{i+1} = k_i - p_i$, where $p_i \geq k_i/p$ is the number of pairs connected at iteration i. Clearly, at the beginning of each iteration i there exists a solution to the residual problem (namely, a subgraph that connected k_i pairs from the remaining pairs) of weight at most τ, where τ is the optimal solution value to the original problem. Hence the weight of the bicriteria solution computed at each iteration is at most $f \cdot \tau$.

Note that $\frac{1}{1-1/p} = 1 + \frac{1}{p-1} = \alpha$. We have

$$k_i = k_{i-1} - p_{i-1} \leq k_{i-1}(1 - 1/p) = k_{i-1}/\alpha \ .$$

Hence $k_i \leq k/\alpha^i$. The least integer i such that $\alpha^i > k$ is $i = \lfloor \ln k / \ln \alpha \rfloor + 1$, and it bounds the number of iterations. Consequently, the overall weight of the solution computed is at most $f \cdot (\lfloor \ln k / \ln \alpha \rfloor + 1) \cdot \tau$, as claimed.

To see the last statement, note that $\ln(1 + x) > x/2$ for $x \in [0, 1/2]$, and thus for $p = \text{polylog}(n)$ and n large enough we have $\ln \alpha = \ln(1 + \frac{1}{p-1}) \geq \frac{1}{2(p-1)} \geq \frac{1}{2p}$. Hence $\ln k / \ln \alpha \leq 2p \ln k = \text{polylog}(n)$. □

Suppose that for a Steiner k-Forest instance we are given a subgraph $G' = (V', E')$ (E' is a partial solution) and a set D' of demands on V' between distinct connected components of G'; we seek an augmenting edge set $F' \subseteq E \setminus E'$ that connects k' demands in D'. Then we can obtain a residual instance of the problem by contracting every connected component of G' into a single "supernode" and updating the demands accordingly. E.g., if D_{ij} is the set of demands between two connected components C_i and C_j, then after C_i and C_j are contracted into v_i and v_j, respectively, we have $|D_{ij}|$ parallel demands between v_i and v_j; equivalently, we can replace these $|D_{ij}|$ parallel demands by a single demand of profit $|D_{ij}|$, and require that F' will connect a set of demands of total profit k'.

2.2 The hard case $\tau\sqrt{q} \leq n/\rho$

We start by giving an algorithm for Steiner Forest where we bound the solution weight by the number of terminals and the maximum distance between node pairs in D.

LEMMA 2.3. Steiner Forest *admits a polynomial time algorithm that computes a solution F' of weight at most $L(|Q| - 1)$, where $L = \max_{\{u,v\} \in D} \text{dist}_G(u, v)$.*

PROOF. Let (Q, D') be a spanning forest of the demand graph (Q, D). The connected components of (Q, D') coincide with those of (Q, D). For every pair $\{u, v\} \in D'$ let P_{uv} be the edge set of a shortest uv-path, and let F' be the union of these edge sets. It is easy to see that the graph (V, F') connects every pair in D, and clearly its weight is at most $L(|Q| - 1)$. □

COROLLARY 2.4. *Suppose that for a* Steiner k-Forest *instance we are given a subgraph $G' = (V', E')$ of G that has t connected components and contains a set D' of demands such that $\max_{\{u,v\} \in D} \text{dist}_G(u, v) \leq L$. Then there exist a polynomial time algorithm that finds an augmenting edge set F' of weight $w(F') \leq L(t - 1)$ such that $E' \cup F'$ connects all demands in D'.*

PROOF. Contract every connected component of G' into a single node and update the demands accordingly. For the obtained Steiner Forest instance, compute an edge set F' as in Lemma 2.3. Note that if two connected components C_i and C_j are contracted into v_i and v_j, respectively, then joining v_i and v_j by a path connects all demands between C_i and C_j. This implies that $E' \cup F'$ connects all demands in D', and $w(F') \leq L(t-1)$ by Lemma 2.3. □

Our algorithm executes several procedures (one of them given in Section 3), and chooses the outcome of one of them. Intuitively, in each procedure, we have the following three steps.

(1) CONSTRUCT: This procedure constructs a MCℓ-EPS instance from the demand graph (Q, D) by removing some nodes, choosing some node subsets, and contracting each node subset into a supernode of a certain cost.
(2) COMPUTE: This procedure computes a ρ-approximate solution to the obtained MCℓ-EPS instance, which determines a certain set Q' of terminals.
(3) CONNECT: This procedure returns a graph obtained by connecting some pairs of chosen terminals.

The next statement illustrates this relation between Steiner k-Forest and Mk-ES. It says that if our optimal solution J connects "many" pairs by "short" paths of weight $\leq L$, then we can find an edge set that connect many pairs by total weight at most $\rho q L$.

COROLLARY 2.5. *Suppose that D_J has at least k' pairs $\{u, v\}$ with $\mathrm{dist}_J(u, v) < L$. Then there exists a polynomial time algorithm that computes an edge set F' of weight $w(F') \leq L(\rho q - 1)$ that connects k' pairs from D.*

PROOF. The algorithm is as follows.

Algorithm 1: SHORT-PATHS(G, D, L, k')

1 CONSTRUCT a MCℓ-EPS instance with $\ell = k'$ from the demand graph (R, D) by removing demands $\{u, v\}$ with $\mathrm{dist}_G(u, v) \geq L$.
2 COMPUTE a ρ-approximate solution R' for the obtained Mk'-ES instance.
3 CONNECT: Return F' as in Corollary 2.4 (with $E' = \emptyset$).

Note that Q_J is a feasible solution to the obtained MCℓ-EPS instance with $q = |Q_J|$ nodes. Thus the returned ρ-approximate solution Q' has at most ρq nodes. The graph $(Q', \emptyset)$ has $|Q'| \leq \rho q$ connected components, hence the algorithm from Corollary 2.4 returns an edge set F' of weight at most $w(F') \leq L(\rho q - 1)$ that connects k' pairs. □

The next lemma is the technical heart of our paper. It says that if our optimal solution J connects "many" pairs by "long" paths, then we can find a subgraph that is relatively cheap, has few connected components, and contains $\tilde{\Omega}(k)$ demands.

LEMMA 2.6. *There exists a polynomial time algorithm that when given an instance of* Steiner k-Forest *with unit weights and integers $2 \leq d, h \leq n$ such that $\mathrm{dist}_J(u, v) \geq d$ holds for $k/2$ pairs $\{u, v\}$ in D_J, computes a subgraph $G' = (V', E')$ of G such that G' has $\tilde{O}(\rho\tau/d + n/h)$ connected components, $|E'| = \tilde{O}(\rho h\tau/d + \rho q d)$, and V' contains $\tilde{\Omega}(k)$ pairs from D.*

The proof of this lemma is rather involved, and it is given in Section 3. In the rest of this section we will use this lemma to finish the proof of Theorem 1.3.

COROLLARY 2.7. *There exists a polynomial time algorithm that under conditions of Lemma 2.6 computes a subgraph $G' = (V', E')$ that connects $\tilde{\Omega}(k)$ demands and has size at most $|E'| = \tau \cdot \tilde{O}(f(d,h))$, where*

$$f(d,h) = \rho\tau/d + n/h + \rho h/d + \rho qd/\tau$$

PROOF. We simply connect the components as in Lemma 2.6 using the algorithm from Corollary 2.4 and the obvious distance bound $L = \tau$ (recall that we assume that $\mathrm{dist}_G(u,v) \leq \tau$ for every $\{u,v\} \in D$). We write this a little more formally as Algorithm 2.

Algorithm 2: LONG-PATHS(G, D, τ, h, d)

1 Compute a graph $G' = (V', E')$ using Lemma 2.6.
2 Compute an augmenting edge set F' as in Corollary 2.4.
3 Return $E' \cup F'$.

By Lemma 2.6, G' has $\tilde{O}(\rho\tau/d + n/h)$ connected components. Thus by Corollary 2.4 $|F'| = \tau \cdot \tilde{O}(\rho\tau/d + n/h)$. We also know from Lemma 2.6 that $|E'| = \tilde{O}(\rho h\tau/d + \rho qd)$ and that the algorithm connects $\tilde{\Omega}(k)$ demand pairs. Consequently,

$$|F'| + |E'| = \tau \cdot \tilde{O}(\rho\tau/d + n/h) + \tilde{O}(\rho h\tau/d + \rho qd) = \tau \cdot \tilde{O}(\rho\tau/d + n/h + \rho h/d + \rho qd/\tau)\ ,$$

as claimed. □

We now instantiate some parameters to show that for certain ranges of values, the combination of Algorithms 1 and 2 gives a good approximation ratio.

LEMMA 2.8. *If $\tau\sqrt{q} \leq n/\rho$ then* Steiner k-Forest *with unit weights admits approximation ratio $\tilde{O}\left(\left(\frac{\rho^2 nq}{\tau}\right)^{1/3}\right)$.*

PROOF. Let $f(d,h)$ be as in Corollary 2.7 and let

$$d = \left(\frac{n\tau^2}{\rho q^2}\right)^{1/3} \qquad \text{and} \qquad h = \left(\frac{n^2\tau}{\rho^2 q}\right)^{1/3} = d \cdot \left(\frac{nq}{\rho\tau}\right)^{1/3}.$$

Note that since $\tau \geq q/2$, then for $\rho \leq n/8$ we have $d, h \geq 2$. Also note that the condition $\tau\sqrt{q} \leq n/\rho$ implies $d, h \leq n$.

We execute two different algorithms: Algorithm 1 with $L = d$ and $k' = k/2$, and Algorithm 2. Then, among the two edge sets returned we choose the one of smaller weight. If D_J has $k/2$ pairs $\{u,v\}$ with $\mathrm{dist}_J(u,v) < d$, then by Corollary 2.5 Algorithm 1 returns an edge set of weight at most $\rho qd = \tau \cdot O(f(d,h))$ that connects $k/2$ demands. Otherwise, if D_J has $k/2$ pairs $\{u,v\}$ with $\mathrm{dist}_J(u,v) \geq d$, then by Corollary 2.7 Algorithm 2 returns an edge set of weight at most $\tau \cdot \tilde{O}(f(d,h))$ that connects $\tilde{\Omega}(k)$ demands. Elementary computations show that

$$\frac{\rho\tau}{d} = \left(\frac{\rho^4 q^2 \tau}{n}\right)^{1/3} \qquad \text{and} \qquad \frac{n}{h} = \frac{\rho h}{d} = \frac{\rho qd}{\tau} = \left(\frac{\rho^2 nq}{\tau}\right)^{1/3}.$$

The statement follows, since the condition $\tau\sqrt{q} \leq n/\rho$ implies $\frac{\rho^4 q^2 \tau}{n} \leq \frac{\rho^2 nq}{\tau}$. □

COROLLARY 2.9. *For any* $0 \leq \gamma \leq 1$, *the following holds: if* $\rho = n^{\gamma}$ *and* $\tau\sqrt{q} \leq n^{1-\gamma}$, *then* Steiner k-Forest *with unit weights admits approximation ratio* $\tilde{O}\left(n^{1/3+2\gamma/3}\right)$.

PROOF. This follows from Lemma 2.8, since $\frac{\rho^2 nq}{\tau} = \frac{q}{\tau}\rho^2 n \leq 2\rho^2 n = 2n^{1+2\gamma}$. □

From Lemma 2.1 and Corollary 2.9 it follows that Steiner k-Forest with unit weights admits approximation ratio $\tilde{O}\left(n^{1/3+2\gamma/3}\right)$, as claimed in Theorem 1.3. It only remains to prove Lemma 2.6.

3. PROOF OF LEMMA 2.6

We give an overview of the proof of Lemma 2.6. Our algorithm has two phases. Roughly speaking, in the first phase we get a collection (called "cluster") of pairwise node disjoint rooted trees in G such that:

- $\tilde{\Omega}(k)$ demands in D_J have their endnodes in distinct trees.
- The radius (height) of each tree is $O(d \log n) = \tilde{O}(d)$.

A tree T is *heavy* if it has more that h edges, and T is *light* otherwise. We will use the following bounds:

- We will show that only $O(\tau/d)$ trees can contain a node from Q_J (this property depends on J, and note that we do not know these trees explicitly).
- Since the trees are pairwise disjoint and since every heavy tree has at least h nodes, we have $O(n/h)$ heavy trees (note that we know these trees explicitly).

While the number of heavy trees is small, we do not have a good bound on the size of a heavy tree, so taking an entire (just one) heavy tree into G' might make $|E'|$ too large. However, to connect a set Q_T of terminals that belong to the same heavy tree T, we do not have to take the entire tree; the terminals in Q_T can be connected to the root of T using $\tilde{O}(|Q_T|d)$ edges, which can be much smaller than the number of edges in T. On the other hand, the contribution of each light tree to $|E'|$ is small, but we do not have a good bound on the number of light trees. Our strategy is to choose a small amount $O(\rho\tau/d)$ of light trees and a small number $O(\rho q)$ of single nodes that belong to heavy trees, such that their union contains $\tilde{\Omega}(k)$ demands. This is achieved in the second phase, which is similar to Algorithm 1. We construct a MCℓ-EPS instance with $\ell = \tilde{\Omega}(k)$ from the demand graph (Q, D) by contracting the terminals of every light tree into a single "super-node" of cost $\alpha = qd/\tau$. We then compute a ρ-approximate solution Q' for the obtained MCℓ-EPS instance. The returned graph G' is the union of:

- The light trees that correspond to supernodes in Q'.
- Union of the shortest paths from each terminal in Q' that belongs to a heavy tree to the root of the heavy tree it belongs to.

The returned graph G' contains $\ell = \tilde{\Omega}(k)$ pairs from D. By the construction, the number of connected components in G' is bounded by the number of supernodes in Q' plus the total number of heavy trees.

Since only $O(\tau/d)$ (light) trees contain a node from Q_J, the obtained MCℓ-EPS instance admits a solution of node cost $O(\alpha\tau/d + q) = O(q)$ (this is the reason for

choosing supernode costs $\alpha = qd/\tau$). Thus the returned ρ-approximate solution Q' has node cost $O(\rho q)$, so the number of supernodes in Q' is $O(\rho q/\alpha) = O(\rho\tau/d)$. The number of heavy trees is $O(n/h)$. Thus G' has $\tilde{O}(\rho\tau/d + n/h)$ connected components, as claimed in Lemma 2.6.

On the other hand, $|E'|$ is bounded by the sum of:

- h times the number of supernodes in Q, which is $O(h\rho\tau/d)$.
- $\tilde{O}(d)$ times the the number of ordinary nodes in Q, which is (ρqd).

Consequently, $|E'| = \tilde{O}(\rho h\tau/d + \rho qd)$, so the bounds promised in Lemma 2.6 hold.

3.1 Cluster decompositions

The following type of cluster is similar to other construction that date back to [Awerbuch 1985]. But the details are always somewhat different, and we need to present the algorithm and some of its properties in full.

DEFINITION 3.1. *A (d, p)-*cluster *of a subset A of nodes in a graph G (possibly with edge weights) is a collection $\mathcal{T}_A$ of node-disjoint rooted subtrees of G such that the following holds:*

(1) Every node in A belongs to exactly one tree in $\mathcal{T}_A$.

(2) The radius (height) of each tree is at most p.

(3) $\text{dist}_G(u, v) > d$ for any two nodes $u, v \in A$ that belong to distinct trees.

*A (d, p)-*cluster-decomposition *of S is a collection of (d, p)-clusters $\{\mathcal{T}_A : A \in \mathcal{A}\}$ where $\mathcal{A}$ is a partition of S.*

Let $\lg i = \log_2 i$ denote logarithm with base 2. The purpose of this section is to prove the following theorem.

THEOREM 3.1. *There exists a polynomial time algorithm that given a graph $G = (V, E)$, a subset $S \subseteq V$ of terminals, and an integer $1 \leq d \leq n/2$ returns a $(d, d(\lg |S| + 1))$-cluster-decomposition of S with at most $\lg |S| + 1$ clusters.*

PROOF. To prove the theorem, we design a polynomial time algorithm for the following intermediate problem:
Find a $(d, d(\lg |S| + 1))$-cluster $\mathcal{T}_A$ of a subset $A \subseteq S$ with $|A| \geq |S|/2$.
Given such an algorithm, we construct the clusters in the decomposition sequentially, such that after construction of each cluster $\mathcal{T}_A$ we remove from S the corresponding set A of nodes and add A to $\mathcal{A}$. Clearly, at the end $\mathcal{A}$ is a partition of S. After each cluster construction the number of nodes in S decreases by a factor of at least 2, hence $|\mathcal{A}| \leq \lg |S| + 1$.

For a subtree T of G let $B_d(T) = \{v \in S \setminus T : \text{dist}_G(T, v) \leq d\}$ denote the set of nodes in $S \setminus T$ of distance at most d from T. Algorithm 3 below solves the above intermediate problem.

Algorithm 3: CLUSTER-CONSTRUCT(G, S, d)

```
initialize 𝒯 ← ∅, A ← ∅
while S ≠ ∅ do
    Choose root s ∈ S and set T ← ({s}, ∅)
    while |B_d(T)| ≥ |S ∩ T| do
        EXPAND(T): For each v ∈ B_d(T), add to T some shortest path from T
        to v.
UPDATE(𝒯, S, A): Add T to 𝒯, move T ∩ S from S to A, and remove B_d(T)
from S.
return 𝒯
```

In the algorithm, the lines in the loop add nodes to the trees as long as the number of terminals in the "boundary" is at least equal to the number of terminals inside the tree. When this is no longer the case, the update line removes the boundary of the new tree from the graph.

Each time we expand T, the radius of T increases by at most d while $|T \cap S|$ is at least doubled. Thus the radius of T is bounded by $d(\lg |S| + 1)$.

Note that at the update step, the set $B_d(T)$ of nodes within distance d from T is removed from S, and thus none of them will belong to A. This implies that at the end of the algorithm, $\mathrm{dist}_G(u, v) > d$ for any two nodes $u, v \in A$ that belong to distinct trees. Note also that the number of nodes moved from S to A and included in T is at least half the number of nodes removed from S (since at this point $|B_d(T)| \leq |T \cap S|$). This implies that $|A| \geq |S|/2$.

It remains to prove that the trees in $\mathcal{T}$ are pairwise node-disjoint. Suppose to the contrary that there is $v \in V$ that belongs to two trees $T_1, T_2 \in \mathcal{T}$, where T_2 was constructed after T_1. Let T_2' denote the tree stored in T_2 right before the expansion step when v was added to T_2. When v was added to T_2', this was because there was a path of length $\leq d$ that goes through v from T_2' to some $t \in S$. In particular, $\mathrm{dist}_G(v, t) \leq d$. Now let T_1' denote the tree stored in T_1 right after the expansion step when v was added to T_1. At this point, t was not added to T_1', hence we must have $\mathrm{dist}_G(v, t) > d$. This is a contradiction. □

Theorem 3.1 extends to edge-weighted graphs by an elementary construction of replacing every edge e of weight w_e by a path of length w_e.

COROLLARY 3.2. *Given a* Steiner k-Forest *instance, let* $\{\mathcal{T}_A : A \in \mathcal{A}\}$ *be a cluster-decomposition as in Theorem 3.1 of the set* Q *of terminals. Then for any* $D' \subseteq D$ *there exist* $A, B \in \mathcal{A}$ *(possibly* $A = B$*) such that* $|D'(A, B)| = \Omega(|D|/\lg^2 |Q|)$*, where* $D'(A, B)$ *is the set of pairs in* D *with one node in* A *and the other in* B*.*

PROOF. We have $|\mathcal{A}| \leq \lg |Q| + 1$. The statement follows by a standard averaging argument from the observations that

$$\sum_{\{A,B\} \subseteq \mathcal{A}} |D'(A, B)| \;=\; |D'|$$

$$|\{\{A, B\} : \{A, B\} \subseteq \mathcal{A}\}| \;=\; |\mathcal{A}|(|\mathcal{A}| + 1)/2 = O(\lg^2 |Q|)\ .$$

□

For simplicity of exposition, let us assume that we know the sets A, B as in the above corollary (we can try all $O(\lg^2 |Q|)$ possible choices) and that $A \neq B$ (the analysis of the case $A = B$ is similar). Furthermore, by Corollary 3.2, we lose only a polylogarithmic factor by replacing D by $D(A, B)$; hence we assume that $D = D(A, B)$, that our optimal solution J connects k pairs from $D = D(A, B)$ (namely, that $Q_J \subseteq A \cup B$), and denote by $\mathcal{T}_A, \mathcal{T}_B$ the corresponding pair of clusters.

3.2 Choosing and connecting trees

As explained before, the two parameters d and h from Lemma 2.6 are related to the cluster decomposition, and have the following meaning:

- d is the cluster decomposition parameter as in Theorem 3.1.
- h is a threshold on a tree size in a cluster; a tree T is *heavy* if it has more that h edges, and T is *light* otherwise.

Recall that Lemma 2.6 assumes that $\text{dist}_J(u, v) \geq d$ holds for at least half of the pairs $\{u, v\} \in D_J$. Thus removing from D pairs $\{u, v\}$ with $\text{dist}_J(u, v) < d$ loses only a factor of 2 in the number of pairs in D_J. For simplicity of exposition, we will assume that $\text{dist}_J(u, v) \geq d$ holds for all pairs in D_J. The following lemma shows that then the number of trees that contain a node from Q_J cannot be too large.

LEMMA 3.3. *Suppose that* $\text{dist}_J(u, v) \geq d$ *for every* $\{u, v\} \in D_J$. *Then at most* $2\tau/d$ *trees in* $\mathcal{T}_A$ *(or in* $\mathcal{T}_B$*) contain a node from* Q_J.

PROOF. Let $\mathcal{T}'_A$ be the family of those trees in $\mathcal{T}_A$ that contain a node from Q_J. For every tree $T \in \mathcal{T}'_A$ fix some pair $\{u_T, v_T\} \in D_J$, where $u_T \in T$. Let P_T be the set of the first $d/2$ edges on the $u_T v_T$-path in the graph (V, J). For any distinct $T, T' \in \mathcal{T}'_A$ the sets in $P_T, P_{T'}$ are disjoint, since the distance in G between any two terminals that belong to distinct trees is at least d. The statement follows. □

Let $\alpha = dq/\tau$. We execute the following Algorithm 4 (for illustration see Fig. 1).

Algorithm 4: CHOOSE-CONNECT-TREES$(G, D, \mathcal{T})$

1 CONSTRUCT a MCℓ-EPS instance with $\ell = k$ from the demand graph (Q, D) by contracting the terminals of every light tree into a single node of cost α.
▷ Comment: We get a multigraph with node-costs in $\{1, \alpha\}$ and unit edge-profits.
2 COMPUTE a ρ-approximate solution Q' for the obtained MCℓ-EPS instance.
3 CONNECT: $G' = (V', E')$ is the union of the light trees that correspond to supernodes in Q' and shortest paths from each terminal in Q' that belongs to a heavy tree to the root of the heavy tree it belongs to.

Note that by Lemma 3.3 at most $O(\tau/d)$ trees contain a node from Q_J, hence the total weight of the supernodes that correspond to such trees is $\alpha \cdot O(\tau/d) = O(q)$. Also note that for our choice $d = \left(\frac{n\tau^2}{\rho q^2}\right)^{1/3}$, we have $\alpha = \left(\frac{n\tau}{\rho q}\right)^{1/3} \geq \left(\frac{n}{2\rho}\right)^{1/3} \geq 1$, since $\tau \geq q/2$ and since $n >> \rho$.

To finish the proof of Lemma 2.6 it is sufficient to prove the following.

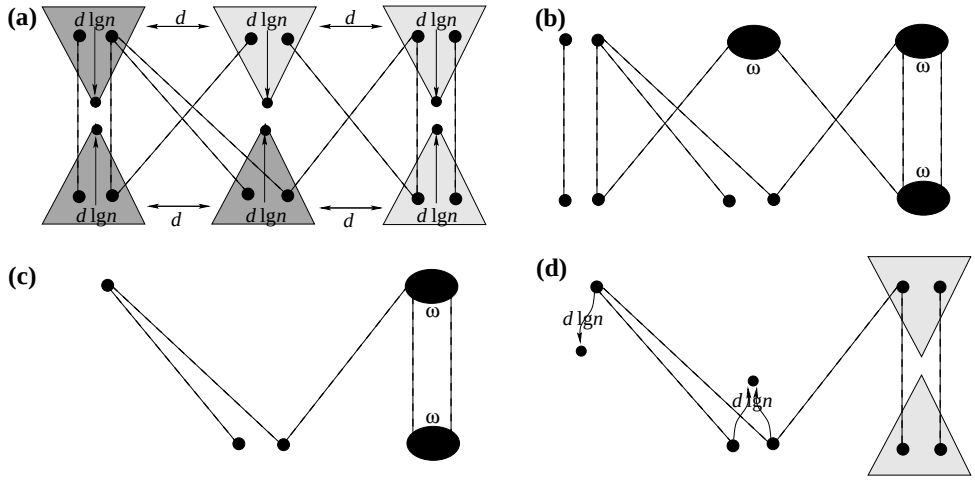

Fig. 1. Illustration to Algorithm 4. Light trees are shown by bright gray triangles, demands are shown by dashed lines. (a) The collection of trees $\mathcal{T}_A \cup \mathcal{T}_B$ (a tree in $\mathcal{T}_A$ may intersect a tree in $\mathcal{T}_B$). (b) The constructed instance of MCℓ-EPS. (c) The computed solution Q' for $k = 5$. (d) The returned graph G' has 4 connected components (two of them are light trees chosen).

LEMMA 3.4. *Suppose that* $\mathrm{dist}_J(u, v) \geq d$ *holds for all* $\{u, v\} \in D_J$. *Then Algorithm 4 computes a graph* $G' = (V', E')$ *with* $O(\rho\tau/d + n/h)$ *connected components and* $|E'| = \tilde{O}(\rho q d + h\tau/d)$.

PROOF. Since $\ell = k$, V' contains k pairs from D. The obtained MCℓ-EPS instance admits a solution with at most q nodes and $O(\tau/d)$ supernodes, since by Lemma 3.3 $O(\tau/d)$ trees contain a node from Q_J. Hence the node cost of this solution is $O(\alpha\tau/d + q) = O(q)$. Consequently, the returned ρ-approximate solution Q' has node cost $O(\rho q)$. Now, note the following bounds:

(i) $|Q'| = O(\rho q)$ and the number of supernodes in Q' is at most $|Q'|/\alpha = O(\rho\tau/d)$ (since every node has cost ≥ 1 and every super-node has cost α).

(iii) The total number of heavy trees is $O(n/h)$ (since the heavy trees in each of $\mathcal{T}_A, \mathcal{T}_B$ are pairwise disjoint and each of them has at least h nodes).

(iv) The length of the paths from a terminal to the root of its tree is $\tilde{O}(d)$ (since the radius of each tree is at most $d \log |Q| = \tilde{O}(d)$).

The number of connected components in G' is bounded by the sum of:

- The number of super-nodes in Q', which is $O(\rho\tau/d)$, by (i).
- The total number of heavy trees, which is $O(n/h)$, by (ii).

Thus the total number of connected components in G' is $O(\rho\tau/d + n/h)$, as claimed.

The number of edges in $|E'|$ is bounded by the sum of:

- h times the number of supernodes in Q', which is $h \cdot O(\rho\tau/d)$, by (i).
- $|Q'| \cdot \tilde{O}(d) = \tilde{O}(\rho q d)$, by (i) and (iii).

Thus $|E'| = \tilde{O}(\rho q d + h\tau/d)$, as claimed. □

This ends the proof of Lemma 2.6, and thus also the proof of Theorems 1.3.

4. PROOF OF THEOREM 1.2

We first make the node costs bounded by a polynomial in n. We remove nodes of cost more than τ and zero the edges of cost at most τ/n^2, where τ is the optimal solution value. The cost we ignore due to the zeroing of the node costs is less than τ and is negligible in our context. Then we divide all the weights by the minimum weight and round the value down. Note that the cost of any edge over the cost of the minimum weight is at least 1. Hence the rounding down loses a negligible factor of 2: the worse case is that we may round a number that is at most 2 to 1. If the profits are exponential in n or larger, we give a bicriteria approximation in which we have the same ratio but we cover only $\ell - \ell/poly(n)$ profit where $poly(n)$ is an arbitrary polynomial function of n. Thus our generalization of [Chlamtac et al. 2012] is really for the case when node weights are arbitrary and edge profits are polynomial in n.

We call an instance of MCℓ-EPS *simple* if all the edge-profits are the same and there are at most two distinct node costs (say c_1 and c_2) such that every edge has exactly one endpoint of each cost (note that it might be the case that $c_1 = c_2$).

LEMMA 4.1. *If* MCℓ-EPS *admits an f-approximation algorithm on simple instances, then* MCℓ-EPS *admits a bicriteria approximation algorithm that returns a graph of node-cost $O(f)$ times the optimal and edge-profit $\Omega(\ell/\log^3 n)$.*

PROOF. Let $\langle G = (V, E), c, p, \ell\rangle$ be an instance of MCℓ-EPS. Recall that we may assume that the node costs are polynomial in n because of the reduction described above. Also, by assumption, the edge profits are bounded by a polynomial in n.

Partition E into $O(\log n)$ sets $E_h = \{e \in E : 2^h \le p_e < 2^{h+1}\}$. Each $e \in E_h$ is given profit 2^h. Partition the nodes similarly: $V_i = \{v \in E : 2^h \le c_v < 2^{h+1}\}$. according to powers of 2. Let E_{ijh} be the set of edges in E_h with one end in V_i and the other in V_j. The edge sets E_{ijh} partition E, and there are $O(\log^3 n)$ such sets. Each graph $G_{ijh} = (V_i \cup V_j, E_{ijh})$ gives a simple instance of MCℓ-EPS, and one of them contains $\Omega(\ell/\log^3 n)$ profit of the optimum. We run the algorithm for simple instances on each graph G_{ijh} with $\Omega(\ell/\log^3 n)$ instead of ℓ, and return the one of minimum node cost. The returned subgraph has node cost $O(f)$ times the optimal and $\Omega(\ell/\log^3 n)$ edge-profit, as required. □

By the same argument as in Lemma 2.2 we have the following.

LEMMA 4.2. *Suppose that* MCℓ-EPS *admits a bicriteria approximation algorithm that returns a graph of node-cost f times the optimal and edge-profit at least $(1 - 1/\alpha) \cdot \ell$, where $1 < \alpha < \ell$. Then* MCℓ-EPS *admits an $f \cdot \lceil \ln \ell / \ln \alpha \rceil$-approximation algorithm.*

From Lemmas 4.1 and 4.2 we have the following.

COROLLARY 4.3. *Suppose that* MCℓ-EPS *on simple instances admits a bicriteria approximation algorithm that returns a graph of node-cost f times the optimal and edge-profit $\tilde{\Omega}(\ell)$. Then* MCℓ-EPS *admits a $\tilde{O}(f)$-approximation algorithm.*

In the rest of this section we prove the following statement, which together with Corollary 4.3 implies Theorem 1.2.

LEMMA 4.4. MCℓ-EPS *on simple instances admits a bicriteria approximation algorithm that returns a graph of node-cost f times the optimal and edge-weight $\tilde{\Omega}(\ell)$, where $f = \tilde{O}\left(n^{3-2\sqrt{2}+\epsilon}\right)$ for arbitrarily small constant $\epsilon > 0$.*

We need some definitions and results from [Chlamtac et al. 2012].

DEFINITION 4.1 [CHLAMTAC ET AL. 2012]. *A bipartite graph $G = (V_1 \cup V_2, E)$ is called* (n_1, d_1, n_2, d_2)-nearly regular *if for every $i = 1, 2$ we have $|V_i| = n_i$ and the following condition on the degrees holds:*

$$d_i \geq \max_{v \in V_i} d(v) \geq \min_{v \in V_i} d(v) = \Omega(d_i / \log n).$$

LEMMA 4.5 [CHLAMTAC ET AL. 2012]. *Any graph $H = (V, E)$ contains an (n_1, d_1, n_2, d_2)-nearly regular subgraph with $\Omega(|E|/\log^2 n)$ edges, for some n_1, d_1, n_2, d_2.*

A key step in [Chlamtac et al. 2012] was the following lemma:

LEMMA 4.6 [CHLAMTAC ET AL. 2012]. *For any $\epsilon > 0$ there exists a randomized polynomial time algorithm that given a bipartite graph G on n nodes that contains an (n_1, d_1, n_2, d_2)-nearly regular subgraph, returns a subgraph $G' = (V', E')$ of G such that $|V'| \leq f \cdot (n_1 + n_2)$ (with probability 1) and $\mathbf{E}[|E'|] = \tilde{\Omega}(n_1 d_1)$, where $f = n^{3-2\sqrt{2}+\epsilon}$.*

We prove the following refinement of Lemma 4.6, which gives a more "balanced" guarantee.

LEMMA 4.7. *For any $\epsilon > 0$ there exists a randomized polynomial time algorithm that given a bipartite graph G on n nodes that contains an (n_1, d_1, n_2, d_2)-nearly regular subgraph, returns a subgraph $G' = (V', E')$ of G such that $|V' \cap V_1| \leq f n_1$ and $|V' \cap V_2| \leq f n_2$ (with high probability) and $\mathbf{E}[|E'|] = \tilde{\Omega}(n_1 d_1)$, where $f = n^{3-2\sqrt{2}+\epsilon}$.*

PROOF. We will assume that $n_1 \geq n_2$; otherwise we just switch indices. Note that if $n_1 \leq 2n_2$, then the algorithm from Lemma 4.6 produces a subgraph that satisfies the new stronger requirement on the chosen nodes. So suppose that $n_1 > 2n_2$. For simplicity, let us also assume that $p = n_1/n_2$ is an integer.

Let $\hat{G} = (V_1 \cup \hat{V}_2, \hat{E})$, where $\hat{V}_2$ consists of p copies of V_2 and $\hat{E}$ is obtained by putting between V_1 and each copy of V_2 a copy of E. For a subgraph $G' = (V_1' \cup V_2', E')$ of G let $\hat{G}' = (V_1' \cup \hat{V}_2', \hat{E}')$ denote the corresponding subgraph of $\hat{G}$, i.e. where between each copy of V_1' and V_2' we include a copy of E'. Note that $|\hat{V}_2'| = p|V_2'|$, that $d_{\hat{G}'}(v) = p d_{G'}(v)$ if $v \in V_1'$, and that if $\hat{v} \in \hat{V}_2$ is a copy of $v \in V_2$ then $d_{\hat{G}'}(\hat{v}) = d_{G'}(v)$. This implies that if G' is (n_1, d_1, n_2, d_2)-nearly regular then $\hat{G}'$ is (n_1, d_2, n_1, d_2)-nearly regular.

We run the algorithm from Lemma 4.6 on the instance $\langle \hat{G}, (n_1, d_2, n_1, d_2) \rangle$ independently $\tilde{O}(n^2)$ times, and among the subgraphs computed take one $\hat{G}' = (V_1' \cup \hat{V}_2', \hat{E}')$ with maximum number of edges. For each $v \in V_2$, let T_v denote the number of copies of v in $\hat{V}_2'$. We build V_2' by sampling each $v \in \hat{V}_2'$ independently with probability T_v/p. Let E' be the set of edges between V_1' and V_2'. We will return the graph $G' = (V_1' \cup V_2', E')$.

We now prove the bounds on the sizes of V_1', V_2', and E'. Since we run the algorithm as in Lemma 4.6, $|V_1'| \leq 2fn_1$ and $\sum_{v \in V_2} T_v \leq |\hat{V}_2'| \leq 2fn_1$. By linearity

of expectations, we get that the expected size of V_2' is at most $\frac{n_2}{n_1}\sum_{v\in V_2} T_v \le 2fn_2$. Since each node in V_2 was chosen independently, a simple Chernoff bound implies that $|V_2'| = \tilde{O}(fn_2)$ with high probability.

To bound $|E'|$, note that a Chernoff bound implies that with high probability $|\hat{E}'| = \tilde{\Omega}(n_1 d_2)$ (since we ran Lemma 4.6 a polynomial number of times and took the best, and each run was independent). An edge $uv \in E$ with $u \in V_1'$ is included in our subgraph with probability $T_v/p = T_v n_2/n_1$. Thus

$$\begin{aligned}\mathbf{E}[|E'|] &= \sum_{u\in V_1'}\sum_{v\in V_1': uv\in E} T_v/p = \frac{n_2}{n_1}\sum_{u\in V_1'}\sum_{v\in V_2': uv\in E} T_v \\ &= \frac{n_2}{n_1}\cdot\tilde{\Omega}(n_1 d_2) = \tilde{\Omega}(n_2 d_2) = \tilde{\Omega}(n_1 d_1),\end{aligned}$$

proving the lemma. □

Now we finish the proof of Lemma 4.4. Let $\langle G = (V_1 \cup V_2, E), m, (c_1, c_2)\rangle$ be a simple MCℓ-EPS instance. Let $G^* = (V_0^* \cup V_1^*, E^*)$ be an optimal subgraph. Applying Lemma 4.5 to G^* implies that there exist values of n_1, d_1, n_2, d_2 such that there is a (n_1, d_1, n_2, d_2)-nearly regular subgraph of G of cost at most $c(V^*) = c_1|V_1^*| + c_2|V_2^*|$ that contains at least $\tilde{\Omega}(\ell) = \tilde{\Omega}|E^*|$ edges (note that up to polylogs $n_1 d_1 = n_2 d_2 = \ell = |E^*|$). So when we run the algorithm from Lemma 4.7, we get a graph $G' = (V_1' \cup V_2', E')$ with the properties that with high probability $|V_1'| = \tilde{O}(fn_1)$ and $|V_2'| = \tilde{O}(fn_2)$ and in expectation $|E'| = \tilde{\Omega}(n_1 d_1) = \tilde{\Omega}(\ell)$. The node-cost of this subgraph is $\tilde{O}(fn_1c_1 + fn_2c_2) = \tilde{O}(f)\cdot c(V^*)$. This proves Lemma 4.4, and thus also the proof of Theorem 1.2 is complete.

5. DISCUSSION AND FUTURE WORK

A central question is: Does the weighted case admits a better than $\Omega(\sqrt{n})$ ratio? It seems that our techniques are not good enough for proving such a ratio; the hard case is when $q >> \sqrt{n}$ because in the weighted case we have no solution of cost n. It may be the case that adding new techniques, on top of our techniques, will allow breaking the $O(\sqrt{n})$ ratio for both Steiner k-Forest and Dial-a-Ride with general edge weights. However, we cannot be sure of that, as there are combinatorial problems in which the approximation ratios of the weighted and the unweighted cases are drastically different For example for the Hard Capacity Vertex Cover problem, the case in which the nodes have weights, admits only a logarithmic ratio [Chuzhoy and Naor 2006], while for weights 1, the problem admits ratio 2, which is the long lasting best ratio known for the simpler Vertex Cover problem.

REFERENCES

Agrawal, A., Klein, P., and Ravi, R. 1995. When trees collide: an approximation algorithm for the generalized Steiner problem on networks. *SIAM J. Computing 24,* 3, 440–456.

Althöfer, I., Das, G., Dobkin, D. P., Joseph, D., and Soares, J. 1993. On sparse spanners of weighted graphs. *Discrete & Computational Geometry 9*, 81–100.

Awerbuch, B. 1985. Complexity of network synchronization. *Journal of the ACM 32,* 4, 804–823.

Berman, P. and Karpinski, M. 2006. 8/7-approximation algorithm for (1, 2)-tsp. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006.* 641–648.

Bhaskara, A., Charikar, M., Chlamtac, E., Feige, U., and Vijayaraghavan, A. 2010. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest k-subgraph. In *STOC.* 201–210.

Byrka, J., Grandoni, F., Rothvoss, T., and Sanità, L. 2013. Steiner tree approximation via iterative randomized rounding. *J. ACM 60,* 1, 6.

Charikar, M. and Raghavachari, B. 1998. The finite capacity dial-a-ride problem. In *FOCS*. 458–467.

Chlamtac, E., Dinitz, M., and Krauthgamer, R. 2012. Everywhere-sparse spanners via dense subgraphs. In *FOCS*. 758–767.

Chuzhoy, J. and Naor, J. 2006. Covering problems with hard capacities. *SIAM J. Computing 36,* 2, 498–515.

Feige, U., Kortsarz, G., and Peleg, D. 2001. The dense k-subgraph problem. *Algorithmica 29,* 3, 410–421.

Feldman, M., Kortsarz, G., and Nutov, Z. 2012. Improved approximation algorithms for directed steiner forest. *J. Comput. Syst. Sci. 78,* 1, 279–292.

Garg, N. 2005. Saving an epsilon: a 2-approximation for the k-MST problem in graphs. In *STOC*. 396–402.

Gupta, A., Hajiaghayi, M. T., Nagarajan, V., and Ravi, R. 2010. Dial a ride from k-forest. *ACM Transactions on Algorithms 6,* 2.

Hajiaghayi, M. T. and Jain, K. 2006. The prize-collecting generalized Steiner tree problem via a new approach of primal-dual schema. In *SODA*. 631–640.

Kortsarz, G. and Peleg, D. 1993. On choosing a dense subgraph. In *FOCS*. 692–701.