3-27-2009

A Constant Factor Approximation for the Single Sink Edge Installation Problem

Sudipto Guha
University of Pennsylvania, sudipto@central.cis.upenn.edu

Adam Meyerson
University of California - Los Angeles

Kamesh Munagala
Duke University

A Constant Factor Approximation for the Single Sink Edge Installation Problem

Abstract

We present the first constant approximation to the single sink buy-at-bulk network design problem, where we have to design a network by buying pipes of different costs and capacities per unit length to route demands at a set of sources to a single sink. The distances in the underlying network form a metric. This result improves the previous bound of O(log |R|), where R is the set of sources. We also present a better constant approximation to the related Access Network Design problem. Our algorithms are randomized and combinatorial. As a subroutine in our algorithm, we use an interesting variant of facility location with lower bounds on the amount of demand an open facility needs to serve. We call this variant load balanced facility location and present a constant factor approximation for it, while relaxing the lower bounds by a constant factor.

Keywords

approximation algorithms, network design, Steiner trees, facility location

Comments

A CONSTANT FACTOR APPROXIMATION FOR THE SINGLE SINK EDGE INSTALLATION PROBLEM[*]

SUDIPTO GUHA[†], ADAM MEYERSON[‡], AND KAMESH MUNAGALA[§]

Abstract. We present the first constant approximation to the single sink *buy-at-bulk* network design problem, where we have to design a network by buying pipes of different costs and capacities per unit length to route demands at a set of sources to a single sink. The distances in the underlying network form a metric. This result improves the previous bound of $O(\log |R|)$, where R is the set of sources. We also present a better constant approximation to the related Access Network Design problem. Our algorithms are randomized and combinatorial. As a subroutine in our algorithm, we use an interesting variant of facility location with lower bounds on the amount of demand an open facility needs to serve. We call this variant *load balanced facility location* and present a constant factor approximation for it, while relaxing the lower bounds by a constant factor.

Key words. approximation algorithms, network design, Steiner trees, facility location

AMS subject classifications. 68W25, 68W20

DOI. 10.1137/050643635

1. Introduction. Network design problems require laying cables on an underlying metric in order to connect a set of demand points. The network must support each demand point operating at a known peak (or average) rate, and we would like the cheapest possible network supporting these demands. In a metric scenario (which is standard), if the cost of cables is linear in the amount of bandwidth they provide, this problem is polynomial-time solvable using multicommodity flow techniques. However, in several real applications, the costs of cables obey economies of scale; the cost-per-unit-bandwidth is less for a high-capacity cable. Similar problems arise outside the data networks community, e.g., in location theory, where we consider transporting products for sale where the cost is a concave function of the amount of demand transported. The concavity also may arise implicitly, e.g., in clustering data, where we are willing to tolerate a less dense cluster if it contains a larger number of points.

The problem of buy-at-bulk network design, with a single sink node to which all the demand has to be routed, was first introduced by Salman et al. in [21]. They gave an $O(\min\{\log n, \log D\})$ approximation, where the number of nodes is n and D is the maximum demand. They also showed that the problem is NP-hard. Awerbuch and Azar [3] gave an $O(\log^2 n)$ approximation for the multiple sink case, where different demand points may communicate with different sinks. This result may be improved

to $O(\log n)$ using subsequent results of [4, 5, 9] on approximation of metrics using trees. Andrews and Zhang [2] considered the Access Network Design problem, where all demands need to connect to a single conceptual entity, the "core" of the network which represents the internet backbone, a set of file servers, or the factories where a product is produced. Even though there may exist multiple sinks, the sinks are symmetric in that demand points do not care to which sink they are connected. The Access Network Design problem is a special case of a single sink buy-at-bulk problem with the added twist that cables have to be utilized up to a minimum capacity.

All of the above problems can be termed as *uniform* variant in the sense that any cable type can be used between any pair of vertices. For the more general *nonuniform* problem where cable types may have limited availability (but the network still has a single sink), Meyerson, Munagala, and Plotkin provided an $O(\log n)$ approximation in [19].

In this paper we focus on the uniform single sink case. Our main contributions are as follows:

- Our algorithm provides the first constant factor approximation for the single sink buy-at-bulk problem. This subsumes the Access Network Design problem [2] as well. The approximation ratio we obtain for the buy-at-bulk problem is 292.
- We provide a Structure Theorem that allows us to identify the key regions in the concave function. This allows one to write a linear program (LP) with an $O(1)$ integrality gap. All subsequent analysis of this problem showing an $O(1)$ approximation (see below and section 7) depends on this theorem as well.
- We define and provide the first constant factor bicriteria approximation algorithm for a natural variant of facility location, the *Load Balanced Facility Location* problem, where there is a lower bound specified on the demand an open facility needs to serve. This problem arises in several clustering scenarios as well [14].

We provide randomized combinatorial algorithms, but they can be derandomized using standard techniques. The paper [12] contained several other results on multi-level network design problems which were based on similar techniques. We omit their discussion in the interest of keeping the presentation focused and simple.

Solution technique. There are several important novel ideas in our solution methodology. First, the optimal solution could use many cables with slightly differing costs in succession, obtaining marginal benefit from each. Such a solution is hard to characterize. We show that to a constant factor approximation, we can replace such similar cable types with the *same* cable type. This leads to a solution with simpler layered characterization (*Structure Theorem*). In particular, we show that a near-optimal solution is composed of alternating layers of Steiner and shortest path forests, each layer using only a single type of cable. The next layer gets used only when sufficient demand has been accumulated at the roots of the previous layer to make the next cable type cost-effective. Such a solution effectively hierarchically aggregates demand in order to exploit the economies of scale in cable costs.

The challenge now is to approximate the layered solution. We show that the shortest path instance corresponds to a variant of the well-studied facility location problem, where a certain minimum demand (corresponding to making the next cable type feasible) must be collected at the facilities. We present a constant factor approximation for this variant, which we term the *load balanced facility location*. This

is an independent contribution of this work. We present a simple constant factor approximation for the Steiner forest variant as well. We finally show that an iterative bottom-up aggregation of demand using these Steiner and shortest path forests yields a constant factor approximation.

Independent results. Independent of this work, Karger and Minkoff [16] defined the load balanced facility location problem as a subroutine for solving the single commodity *Rent-or-Buy* problem. They obtained the same algorithm as the one we present in section 6. We note that the single commodity Rent-or-Buy problem is a special case of the single sink buy-at-bulk problem considered in this paper.

Garg et al. [10] obtained an $O(K)$ approximation (where K is the number of different cable types) to the single sink buy-at-bulk problem by rounding the natural LP formulation. Our paper improves the approximation ratio via a fully combinatorial algorithm. Since the natural LP relaxations for the Steiner forest and shortest path problems have a constant integrality gap, it is natural to expect that an LP *based on the structure theorem* should have an $O(1)$ integrality gap as well, and we provide this LP for completeness. However, the authors of [10] observe that the Structure Theorem in our paper can be used on their stronger, natural LP as well. This connection has also been made explicit by Talwar [23] subsequently.

Organization of the paper. In section 2, we state the single sink buy-at-bulk problem formally and discuss the structural properties of the optimal solution. In section 3.2, we discuss a scaling idea to remove similar pipe types and show how it improves the structure of the optimum solution. We then present the HIERARCHY algorithm in section 4 and show a constant approximation ratio. We show in section 5 how to improve the approximation ratio for Access Network Design. We present the algorithm for load balanced facility location in section 6 and conclude by surveying results that appeared subsequently to the publication of a preliminary version of this paper [13].

2. Definitions and preliminaries. The single sink buy-at-bulk problem as defined in [21] is as follows: Given a graph $G(V, E)$ with a distance (length) function c_e on the edges, the goal is to construct a network routing a set $S_1 \subseteq V$ of demand nodes to a single sink s. We are given K types of connections (pipes) where pipe type i has a fixed cost σ_i per-unit-length and a capacity u_i. Each demand node $v \in S_1$ needs to transport some amount of demand d_v to the sink. The objective is to optimize the cost of buying pipes along the edges to route all demands to the sink. We are allowed to buy multiple copies of a pipe along the same link. The above can be termed as a capacitated version; we will use an alternate incremental cost formulation of the above problem, which easily models arbitrary piecewise linear concave costs.

DEFINITION 2.1. *In the single sink buy-at-bulk problem we are given a set of pipes, where pipe type i has fixed and incremental costs σ_i and δ_i, respectively. If we transport d units of demand along a path of length L using pipe type i, we will pay a total of $L(\sigma_i + \delta_i d)$. The goal is to construct a network routing a set $S_1 \subseteq V$ of demand nodes to a single sink s, while minimizing the cost of the network.*

Let $f_i(Y) = \sigma_i + \delta_i Y$ be the per-unit-distance cost of routing demand Y along a pipe of type i. Define $h(Y) = \min_i f_i(Y)$. The function $h()$ is piecewise linear and concave. Note that a pipe type which does not affect $h()$ (that is, does not define the envelope) will not affect the solution at all. Therefore, if we focus only on the pipe types which are useful and number the pipes in decreasing order of σ_i, we observe that $\delta_1 > \delta_2 > \cdots > \delta_K$. In this formulation the "capacity" of pipe k is $u_i = \sigma_i / \delta_i$.

It is not hard to see that a solution under this incremental cost formulation costs

at least as much as the same solution under the capacitated model, and at most twice as much as the solution under the capacitated model. Furthermore, we have the following.

LEMMA 2.2 (see [2]). *In the incremental cost model the optimum solution naturally defines a tree.*

The above is straightforward since the costs are subadditive under the above assumptions of $\{\sigma_i\}, \{\delta_i\}$; after the links in the optimum are bought and the fixed cost paid, the entire demand from each node can be routed along the path with the lowest incremental cost (ties broken arbitrarily) from each node to the sink. This would define a tree, and we can eliminate the edges we do not use. Note that as a consequence, every node would use a unique pipe type for its outgoing flow. Also due to the subadditivity property of the costs, along every flow path the pipe types will increase in number.

The authors of [2] also introduced the following problem.

DEFINITION 2.2. *The Access Network Design problem is defined as follows: It is the same as the single sink buy-at-bulk problem with the following added restrictions ($c = 1/2$ is used in [2]):*
  1. *For $2 \le k \le K$, if $d < c\frac{\sigma_k}{\delta_k}$, then $d\delta_{k-1} + \sigma_{k-1} < d\delta_k + \sigma_k$.*
  2. *The smallest demand looks like the smallest pipe capacity, or, more precisely, $\delta_1 > c\sigma_1$.*
  3. *$\sum_{\kappa < k} \sigma_\kappa = O(\sigma_k)$.*

As mentioned earlier, our solutions for the above problems will use solutions to the following variant of the Facility Location problem.

DEFINITION 2.3. *The load balanced Facility Location problem is defined as follows: We are given a network $G(V, E)$ with a distance function $c(\cdot)$ on the edges and a set of demand points, with demands d_j. The cost of opening a facility at location i is f_i. In addition, there is a lower bound of L_i on the demand that a facility opened at i must satisfy. We are required to open facilities and allocate the demands to the open facilities so that an open facility at i has at least L_i demand routed to it. The cost of our solution is the sum of the distances traveled by the demands and the cost of the open facilities. The goal is to minimize this cost.*

3. Single sink buy-at-bulk.

3.1. Roadmap.

Intuition. Assume that we consider only the pipe types which are not dominated by others. Thus the cost per-unit-length is a piecewise concave function of the demand. Observe that as we increase demand along an edge, there are break-points at which it becomes cheaper to use the next larger pipe type. Let g_k be the demand for which it becomes cheaper to use a pipe of type $k+1$ compared to a pipe of type k. Suppose that we are in a scenario $0 = g_0 < u_1 < g_1 < u_2 < g_2 < \cdots < u_K < g_K = \infty$ (we will show how to achieve a similar scenario later).

Observe now that if the demand amount is in the range $[g_{i-1}, u_i]$, we can ignore the incremental cost with a factor 2 loss in cost, and the cost of the edge will just be σ_i times the length of the edge, independent of the demand. If, on the other hand, the demand is in the range $[u_i, g_i]$, we can ignore the fixed cost with a factor 2 loss in cost, and the cost of the edge per-unit-length is δ_i times the demand. This implies that the optimum solution can be converted with a factor 2 loss in cost to a layered solution. Layer i has a Steiner forest using pipes of type i followed by a forest of shortest path trees using pipes of the same type. Each pipe in the Steiner forest has at least g_{i-1} demand, and each pipe in the shortest path forest has at least u_i amount of demand.

The shortest path forest should ensure that we collect at least a demand of g_i, such that we can use the Steiner forest corresponding to a larger pipe type. This gives us a clustering problem where each cluster is supposed to have a minimum number of (weighted by demand) points—this is the reason for using the load balanced Facility Location problem.

For completeness, we first define the Facility Location problem [22].

DEFINITION (Facility Location). *We are given a set of demands D. Let d_j be the demand at $j \in D$. We are given a set of feasible locations F, where ϕ_i is the cost of opening a facility at location $i \in F$. The points D and F are embedded in a metric space where c_{ij} is the distance between points i and j. The goal is to open a subset of facilities $X \subseteq F$ and connect each demand $j \in D$ to the closest open facility $q(j) \in X$, so that the total cost of the open facilities, $\sum_{i \in X} \phi_i$, plus the sum of the routing cost, $\sum_{j \in D} c_{q(j)j} d_j$, is minimized. Let ρ_f denote the best approximation ratio for the Facility Location problem. This is 1.52 due to [18].*

The *load balanced Facility Location* problem has an added constraint: Each $i \in F$ has a lower bound L_i on the demand it needs to serve if opened. The solution X constructed must satisfy that for every $i \in X$, at least L_i demand is routed to i in the solution. In section 6 we prove the following.

THEOREM 3.1. *We can compute a solution for load balanced Facility Location whose cost is $2\rho_f$ times that of the optimal solution, such that our solution relaxes the lower bounds by a factor of $1/3$, so that for $i \in X$, at least $L_i/3$ demand is routed to i.*

However, if we apply the above directly and compare our solution with the optimal solution that satisfies the above-mentioned structural properties, the analysis does not immediately go through. So we use the idea that we will use a larger pipe type only when it is *significantly,* i.e., *by a constant factor,* cheaper. This will allow us to set up a geometric series that accounts for the cost of the shortest path forests. But to bound the cost of the Steiner forests we will need a different idea; namely, we *restrict ourselves to pipes where the σ_i decrease by a constant factor as well*. But this now implies that we should show that even after ruling out pipes according to the above two ideas, there is a feasible solution which is not too expensive. This is the Structure Theorem we prove.

There is, however, one remaining issue regarding how to correlate the costs of the different layers—we introduce a novel strategy where the entire demand of a layer is sent to random node in S_1. Note that it is important for our analysis that the demands be sent to S_1, because the structured solution we derive from modifying the optimum solution is defined with these demand nodes as the ground truth.

In the remainder of the section, we first describe the structured feasible solution. We then present the algorithm in section 4.

3.2. Constructing a layered solution. We now formalize the intuition described above to obtain a layered solution with a cost close to the optimal cost. Our algorithm will progressively construct partial solutions using each pipe type in turn. In order to bound the total cost, we must guarantee that pipes are very different from one another in terms of fixed and incremental costs.

DEFINITION 3.2. *Define a set of pipe types to be* good *if for some $\alpha \in (0, 1/2)$ we have the following:*

1. *For any $k < K$, we have $\sigma_k < \alpha\sigma_{k+1}$.*
2. *For any $k < K$, we have $\alpha\delta_k > \delta_{k+1}$.*

We need to prove that we can guarantee these conditions without increasing the cost of the optimum solution by too much.

LEMMA 3.3. *There exist a set of good pipes and a solution that uses only these types, such that the cost of this solution is at most $1/\alpha$ times the cost of the original optimum solution.*

Proof. We first eliminate pipes in order to guarantee that among the remaining pipes we have $\sigma_k < \alpha\sigma_{k+1}$ while increasing the fixed cost of the optimum solution by at most $1/\alpha$. The incremental cost of the optimum solution can only decrease during this phase.

We find the largest pipe k such that $\sigma_k \geq \alpha\sigma_{k+1}$. We eliminate this pipe, replacing it in the optimum solution with pipe $k+1$. We renumber the pipes and repeat. Notice that if at some point some pipe type is replaced by pipes of type k, then we will always keep pipes of type k in the final solution (since every pipe type higher than k has at least α higher fixed cost). When this finishes, we will have the desired property. The original optimum solution with pipe replacements has fixed cost at most $1/\alpha$ larger since any pipe which was replaced was replaced by a pipe with at most $1/\alpha$ bigger fixed cost. The incremental cost can only decrease, since higher fixed cost implies smaller incremental cost.

We now eliminate pipes in reverse order, where an eliminated pipe is replaced by a pipe with larger incremental cost, to guarantee that among the remaining pipes we have $\alpha\delta_k > \delta_{k+1}$, while increasing the incremental cost of the optimum solution by at most $1/\alpha$. The fixed cost of the optimum solution can only decrease.

Combining these two phases gives the solution claimed by the lemma. $\square$

DEFINITION 3.4. *Assuming that we have a good set of pipes, define b_k to be such that $f_{k+1}(b_k) = 2\alpha f_k(b_k)$. In essence, b_k is a sufficient demand that it becomes considerably cheaper to use a pipe of type $k+1$ rather than a pipe of type k.*

LEMMA 3.5. *For all k, $u_k \leq b_k \leq u_{k+1}$.*

Proof. From the definition of b_k, we can write $\sigma_{k+1} + \delta_{k+1}b_k = 2\alpha(\sigma_k + \delta_k b_k)$. Solving this equation for b_k yields

$$b_k = \frac{\sigma_{k+1} - 2\alpha\sigma_k}{2\alpha\delta_k - \delta_{k+1}} \leq \frac{\sigma_{k+1}}{2\alpha\delta_k - \delta_{k+1}} \leq \frac{\sigma_{k+1}}{\delta_{k+1}} = u_{k+1}.$$

The above shows that $b_k \leq u_{k+1}$; to see the other bound observe that when we have b_k flow, it is cheaper to use a pipe of type $k+1$ rather than a pipe of type k. It follows that $\sigma_{k+1} + \delta_{k+1}b_k < \sigma_k + \delta_k b_k$. Solving this for b_k, we can see that

$$b_k > \frac{\sigma_{k+1} - \sigma_k}{\delta_k - \delta_{k+1}}.$$

Since $\alpha < 1/2$, it follows that $\sigma_{k+1} > 2\sigma_k$ and we can conclude that $b_k > u_k$. $\square$

LEMMA 3.6. *For all k and any demand $D \geq b_k$, $f_{k+1}(D) \leq 2\alpha f_k(D)$.*

Proof. Suppose $D = b_k + x$ for some $x \geq 0$. Then, $f_{k+1}(D) = \sigma_{k+1} + \delta_{k+1}(b_k + x) = 2\alpha(\sigma_k + \delta_k b_k) + \delta_{k+1}x$. Noting that $\delta_{k+1} \leq \alpha\delta_k$, it immediately follows that $f_{k+1}(D) \leq 2\alpha f_k(D)$. $\square$

LEMMA 3.7. *For all k and any demand $D \leq u_k$, $f_{k+1}(D) \geq f_k(D)$.*

Proof. Note that $f_{k+1}(D), f_k(D)$ are nondecreasing linear functions in D, and further, $\delta_{k+1} \leq \delta_k$. Therefore, to prove the lemma, it suffices to observe that $f_{k+1}(u_k) \geq f_k(u_k)$.

But $f_k(u_k) = \sigma_k + \delta_k\frac{\sigma_k}{\delta_k} = 2\sigma_k < \sigma_k/\alpha \leq \sigma_{k+1} \leq f_{k+1}(u_k)$. Note that we require $\alpha \in (0, \frac{1}{2})$. $\square$

We now show that there exists a structured near-optimum solution. Subsequently we will search for solutions which obey this structure and are within a constant factor of the best structured solution.

THEOREM 3.8 (Structure Theorem). *There exists a tree solution that uses pipes of type k on a link iff the demand x on the link satisfies $x \in [b_{k-1}, b_k)$. Further, the tree routes all demands which entered a node using pipe k out of that node using a pipe of type k or $k+1$. This solution pays at most $\frac{1}{2\alpha^2}$ times the optimum solution.*

Proof. First, as noted in section 2, the optimal solution defines a tree due to the subadditive nature of the costs. Further, as noted there, the incremental cost model forces this solution to use only one pipe type per edge. Both these properties will be preserved by the transformations described below.

We first modify the solution to use only the set of good pipes (according to Lemma 3.3). Therefore, we need to show that transformations in the the rest of the proof increase the cost of the solution by at most a factor of $\frac{1}{2\alpha}$.

Consider any edge where x units of flow are routed by the optimum solution. Let $k_0 = \operatorname{argmin}_i f_i(x)$. This is the pipe type used by the optimum solution.

Suppose a pipe has flow $b_{k-1} \le x < u_k$. We know by (repeated) application of Lemmas 3.6 and 3.5 that a pipe of type k would reduce the cost compared to any smaller pipe type. Likewise, by (repeated) application of Lemmas 3.7 and 3.5 we can conclude that a pipe of type k would reduce the cost compared to any larger pipe type. Thus $k = k_0$.

Therefore, what remains to be shown is that if $u_k \le x < b_k$ and we use a pipe of type k, then our cost does not increase significantly. First, due to the discussion above, since $b_k \le u_{k+1}$, we know that it is cheaper to use pipe type $k+1$ compared to any larger pipe type. Thus $k_0 = k$ or $k_0 = k+1$, and we need to compare $f_k(x)$ and $f_{k+1}(x)$ only. Now, by Definition 3.4 and the fact that $f_k(x)$ and $f_{k+1}(x)$ are linear nondecreasing with $\delta_{k+1} \le \delta_k$, it is immediate that $2\alpha f_k(x) \ge f_{k+1}(x)$.

Thus, if we modify the optimum solution (already restricted to good pipes) to use pipe type k in the range $[b_{k-1}, b_k)$, then the cost of the solution goes up by at most a factor of $1/(2\alpha)$; combined with Lemma 3.3 the total cost is at most $1/(2\alpha^2)$ times the optimal cost. To achieve the second part of the lemma, we observe that since we are considering a tree solution, the flow does not decrease as we proceed toward the root. We simply introduce dummy nodes or pipes of length 0 if the largest incoming pipe type is k and the outgoing pipe type is larger than $k+1$.   ☐

An LP formulation. We can encode the structural observation above into an integer program formulation. We modify the graph to include K self-loops of length 0 at every vertex in order to accomodate pipe types required by the Structure Theorem.

Denote by x_{vek} whether the demand at node v uses a pipe of type k on edge e. By y_{ek} we denote whether there exists a pipe of type k on edge e. The integer program can then be formulated as follows. Here $\operatorname{In}(v)$ denotes the set of edges coming into on node v, and $\operatorname{Out}(v)$ the set going out of v. Recall that c_e is the length of edge e, and d_v is the demand at node $v \in S_1$.

$$\text{Minimize} \sum_{e \in E} c_e \left(\sum_k \sigma_k \cdot y_{ek} + \sum_e \sum_{v \in S_1} \sum_k \delta_k \cdot d_v \cdot x_{vek} \right),$$

$$\begin{aligned}
\sum_{e \in \text{In}(w)} x_{vek} &= \sum_{e \in \text{Out}(w)} (x_{vek} + x_{vek+1}) && \forall v \in S_1, w \in V \setminus \{s\}, k, \\
x_{vek} &\leq y_{ek} && \forall v \in S_1, e \in E, k, \\
\sum_{e \in \text{Out}(v)} x_{ve1} &= 1 && \forall v \in S_1, \\
x_{vek}, y_{ek} &\in \{0,1\}.
\end{aligned}$$

In the above integer program (IP), the first constraint encodes the structural condition that if a demand enters node v using a pipe of type k, then it leaves the node using a pipe of type k or $k+1$. The second constraint encodes that a demand can use a pipe of type k on an edge only if a pipe of that type exists on that edge. The third constraint encodes that the demand leaves its source using a pipe of type 1.

The LP is obtained by relaxing the final integrality constraints. It can be shown using Theorem 4.7 that if the LP is written on the set of good pipe types, it has an $O(1)$ integrality gap. Direct LP rounding techniques exist as well [10, 23]. Since the LP is not the main focus of the paper, we omit a proof of the integrality gap.

4. The HIERARCHY algorithm. We will now present the HIERARCHY algorithm for single sink buy-at-bulk based on the structural observations we made above. The scaling idea from the previous section measures that we can compare the cost of our solution in each layer with the respective costs of the optimum solution. The algorithm is presented below.

Our solution will route the demands through the forests of increasing pipe types. This solution need not be a tree but can easily be converted to one of no greater cost.

Let ρ_s and ρ_f denote the best approximation ratios for the Steiner tree and Facility Location problems, respectively. Note that $\rho_s = 1.55$ due to [20] and $\rho_f = 1.52$ due to [18]. Note that for the "shortest path tree" part, we use Theorem 3.1 to obtain a $2\rho_f$ approximation that routes at least $b_i/3$ demand to each open facility. For the "Steiner trees" part, we use the ρ_s approximation.

4.1. Analysis. Let Γ be the structured optimal solution constructed in Theorem 3.8. We define C_i^* to be the total cost of Γ using pipes of type i. The total cost of the structured optimal solution is therefore $\sum_{i=1}^{K} C_i^* = C^*$.

Let T_i^I be the incremental cost of the Steiner tree at layer i and T_i^F be its fixed cost. The total cost of the Steiner tree at layer i is $T_i = T_i^I + T_i^F$.

Let P_i^I be the incremental cost of the shortest path tree at layer i and P_i^F be its fixed cost. The total cost of the shortest path tree at layer i is $P_i = P_i^I + P_i^F$.

Let N_i be the total cost of the consolidation steps for layer i. The total cost of our solution is therefore $\sum_i (T_i + P_i + N_i)$.

LEMMA 4.1. *For all i, v, we have $\mathbf{E}[D_i(v)] = d_v$ and $\mathbf{E}[D_i^A(v)] = d_v$; that is, the expected demand at any node after any of the consolidation steps is the original demand of the node.*

Proof. We will prove this by induction on the steps i. Suppose that the statement is true at some step. We will show that it is true at the next step. There are two cases to consider; either we performed a Steiner tree step or a shortest path tree step.

Note $D_1(v) = d_v$. For ease of notation, define $D_0^A(v) = d_v$ for all nodes v. These two equations define the base case.

Suppose we have just performed a Steiner tree step. By the induction hypothesis we know that $\mathbf{E}[D_i(v)] = d_v$. The node v is a part of some tree t with total demand D_t. We then choose a node for consolidation and the probability that we choose node v is $D_i(v)/D_t$. If we choose v, demand D_t will be placed there; otherwise the demand is 0. Thus the expected amount of demand at v, conditioned on the previous $i - 1$

Algorithm HIERARCHY

- Let s denote the sink node and S_1 the set of original demand nodes. Assume that $s \in S_1$. Let the demand of $v \in S_1 - \{s\}$ be denoted by d_v.

- The algorithm proceeds in phases. In phase i we will use pipes of type i only. Let $D_i(v)$ denote the demand of a node v at the start of the phase. Since the algorithm is randomized, this is a random variable. Let S_i be the set of nonzero demand points (and s) we have at this stage.

  (i) *Steiner trees.* Construct an approximately optimal Steiner tree on S_i. Root this tree at s. Transport the demands from S_i upward along the tree. If on any edge the amount of demand is larger than u_i, we "cut" the tree at that edge. This gives us a forest on S_i where each edge has at most u_i demand through it.

  (ii) *Consolidate A.* Consider any subtree t in the above forest which does not contain s. Let the set of nonzero demand nodes in t be $S_i^A(t)$. Pick a node y at random from $S_i^A(t)$ in proportion to its demand $D_i(y)$. For all nodes in $S_i^A(t)$, we send their demand (currently located at the root of t) to z using pipes of type i. Let A_i be the set of nodes y chosen corresponding to different t. Denote the demand of a node v immediately after this step to be $D_i^A(v)$.

  (iii) *Shortest path trees.* Approximately solve a load balanced Facility Location instance on S_1 with the facility lower bound b_i on all nodes (and no facility costs). If there does not exist b_i total demand, then we instead route directly to the sink. We get a forest of shortest path trees. We route our current demands along these trees to their roots. Note that we solve the load balanced Facility Location problem on the S_1 nodes and *not* on the $S_i^A(t)$ nodes.

  (iv) *Consolidate B.* Consider any facility p opened in the above forest of shortest path trees. Some set of nodes from S_1 were assigned to p, denoted by $S_i^B(p)$, and their (original) total demand is at least $b_i/3$. We choose a node z at random from $S_i^B(p)$ with probability proportional to d_z. For all $y \in S_i^B(p) \cap A_i$, we send their demand (currently at node p) to node z using pipes of type i. Let S_{i+1} be the set of nodes z that are chosen corresponding to the different facilities p. Note that the only nodes currently having nonzero demands are nodes in S_{i+1}.

---

steps, is $D_i(v)$. If we now remove the conditioning, $\mathbf{E}[D_i^A(v)] = \mathbf{E}[D_i(v)] = d_v$, as desired.

Suppose we have just performed a shortest path tree step. Note that we used the S_1 nodes for the load balanced Facility Location construction in this step. The probability that we consolidate to v is $d_v/D(p,i)$, where v is assigned to p in this stage and the total demand assigned to p is $D(p,i)$. Note that

$$D(p,i) = \sum_{u \in S_i^B(p)} d_u.$$

The demand of v is $D_{i+1}(v)$ after this step. Note that v collects the demands of the nodes in set $A_i \cap S_i^B(p)$; further, the demand of a node u in this intersection is $D_i^A(u)$.

Conditioned on the previous steps, the expected value of $D_{i+1}(v)$ is

$$\mathbf{E}[D_{i+1}(v)] \frac{d_v}{D(p,i)} \sum_{u \in S_i^B(p)} D_i^A(u).$$

Now if we remove the conditioning,

$$\mathbf{E}[D_{i+1}(v)] = \frac{d_v}{D(p,i)}\mathbf{E}\left[\sum_{u\in S_i^B(p)} D_i^A(u)\right]$$

$$= \frac{d_v}{D(p,i)}\sum_{u\in S_i^B(p)}\mathbf{E}[D_i^A(u)]$$

$$= \frac{d_v}{D(p,i)}\sum_{u\in S_i^B(p)} d_u = d_v.$$

This proves the inductive case.   □

LEMMA 4.2. $\mathbf{E}[N_i] \leq T_i + P_i$.

Proof. The proof essentially follows by observing that the cost is a concave function of demand, and the consolidation process picks a random node in proportion to the demand sent to the root.

This is obvious for the shortest path trees—let the nodes belonging to a tree rooted at facility p be $v_1,\ldots,v_r$ with distances $\ell_1,\ldots,\ell_r$ from node p. Then the expected consolidation cost is

$$\sum_p\sum_{a=1}^r (d_{v_a}/D)f_i(D)\ell_a \leq \sum_p\sum_{a=1}^r f_i(d_{v_a})\ell_a = P_i.$$

The inequality follows since f_i is a concave function.

For the other step, consider the consolidation process to recursively choose from the root a subtree with probability proportional to the total demand in the subtree. We just saw the proof for a one-level tree that the consolidation cost is less than the cost of sending the flows to the root. This argument is now repeated in each of the subtrees.   □

LEMMA 4.3. $T_i^I \leq T_i^F$ and $P_i^F \leq P_i^I$.

Proof. Since we cut the tree at any edge with more than u_k demand along it, we guarantee that the fixed cost paid on any edge we actually use exceeds the incremental cost.

By the same argument, the Steiner tree stage guarantees at least u_k demand or zero everywhere. For the shortest path tree, if an edge has zero demand flowing on it, we will pay zero for that edge. Otherwise, there is at least u_k demand on the edge, and we pay an incremental cost which exceeds the fixed cost for the shortest path trees.   □

LEMMA 4.4. $\mathbf{E}[P_i^I] \leq 2\rho_f\sum_{j=1}^{j=i}\alpha^{i-j}C_j^*$.

Proof. Suppose the demands at the sources were those from S_1. Then one possible solution would be Γ itself until pipes of type $i+1$ were used. We know that Γ must gather the desired b_i flow before using pipes of type $i+1$. Since we will always pay the incremental cost δ_i, and the incremental costs scale by α, we can guarantee a total cost of at most $\sum_{j=1}^{j=i}\alpha^{i-j}C_j^*$ for this solution. Our actual demand at each node has an expected value equal to the original demand, so the expected value of a feasible solution for P_i^I is bounded as above. The extra factor is due to the approximation of the load balanced Facility Location problem as stated in Theorem 3.1.   □

LEMMA 4.5. $\Pr[D_i(v) \neq 0] \leq 3d_v/b_{i-1}$.

Proof. Let v be the chosen node (denoted by z in step (iv) of the algorithm) corresponding to some p. We obtain the nodes S_i by solving an instance of the load

balanced Facility Location problem on S_1 with lower bounds b_{i-1}. In this solution, each node in S_i except s has demand at least $b_{i-1}/3$. In other words, $\sum_{u \in S_i^B(p)} d_u \geq b_{i-1}/3$.

Now note that v is chosen independent of $\{D_{i-1}(u)\}$, *and this is the reason why S_1 is used and not S_i.* Note that at any node u in S_1, the currently accumulated before the shortest path tree step is $D_{i-1}^A(u)$. We therefore have

$$\mathbf{E}[D_i(v)|D_i(v) > 0] = \sum_{u \in S_i^B(p)} \mathbf{E}[D_{i-1}^A(u)] = \sum_{u \in S_i^B(p)} d_u \geq b_{i-1}/3.$$

Now, combined with the fact that $\mathbf{E}[D_i(v)] = d_v$, the lemma follows.   □

LEMMA 4.6. *Recall that ρ_s is the approximation ratio that the Steiner tree approximation algorithm used; then* $\mathbf{E}[T_i^F] \leq \rho_s(\sum_{j=i}^{j=K} \alpha^{j-i} C_j^* + \sum_{j=1}^{j=i-1} 3(2\alpha)^{i-j} C_j^*)$.

Proof. To bound the cost of T_i^F, we will show that there exists a Steiner tree Γ_i which connects all S_i with a low cost. Note that this Steiner tree will have an approximation ratio at most ρ_s relative to the optimal Steiner tree on the set S_i.

The tree Γ_i will be the tree corresponding to the structured solution Γ, where (1) all the pipes of type $j \geq i$ use pipes of type i and (2) for the nodes v with $D_i(v) > 0$, the edges in the path to the root where any pipe of type $j < i$ is used, that corresponding edge now uses a pipe of type i.

The cost according to part (1) can be bounded by $\sum_{j=i}^{j=K} \alpha^{j-i} C_j^*$. This is because for any pipe of type $j \geq i$, we have $\sigma_i \leq \alpha^{j-i} \sigma_j$. Further, the fixed cost of using the pipe of type j is less than its total cost, and thus the above bound follows.

To bound the contribution of (2), we focus on the subtrees of Γ which have total demand at most b_{i-1} (and therefore use pipes of type $j < i$). Note that due to the introduction of zero length edges, these subtrees may share their (sub)roots, but they will be edge disjoint. For an edge e, let the demand flowing through it be $x_e < b_{i-1}$, and suppose that Γ uses a pipe of type $j_e < i$ for this edge. Let the set of nodes in Γ below e be denoted by Γ_e. Let the length of e be c_e.

Now the edge e is used in Γ_i if any v in the subtree below it has $D_i(v) > 0$. Therefore, the probability that e is used is bounded above by $\sum_{v \in \Gamma_e} 3d_v/b_{i-1}$ using Lemma 4.5 and the union bound. Thus we pay a cost

$$\sum_{v \in \Gamma_e} \frac{3d_v}{b_{i-1}} c_e \sigma_i = \frac{3x_e}{b_{i-1}} \sigma_i c_e \leq \frac{3x_e}{b_{i-1}} f_i(b_{i-1})$$

$$\leq \frac{3x_e}{b_{i-1}} (2\alpha)^{i-j_e} f_{j_e}(b_{i-1}) \leq 3(2\alpha)^{i-j_e} f_{j_e}\left(\frac{x_e}{b_{i-1}} b_{i-1}\right) = 3(2\alpha)^{i-j_e} f_{j_e}(x_e).$$

The second inequality follows from Lemma 3.6. Therefore, if we sum the right-hand side over all j_e, the contribution from edges in that level will be $3(2\alpha)^{i-j_e} C_{j_e}^*$. Taking the contribution of both (1) and (2) together, the total cost of Γ_i is at most

$$\sum_{j=i}^{j=K} \alpha^{j-i} C_j^* + \sum_{j=1}^{j=i-1} 3(2\alpha)^{i-j} C_j^*.$$

We can find a Steiner tree of cost at most ρ_s times the above cost, so the lemma follows.   □

THEOREM 4.7. *The* HIERARCHY *algorithm is a constant-approximation for the single sink buy-at-bulk problem.*

Proof. By Lemmas 4.2 and 4.3, the total expected cost of our solution, $\sum_i \mathbf{E}[N_i + P_i + T_i]$, is bounded by $\sum_i 2(2T_i^F + 2P_i^I)$. Using Lemmas 4.4 and 4.6, we conclude that the expected cost of our solution is bounded by the following:

$$4 \sum_i \left(\rho_s \sum_{j=i}^{j=K} \alpha^{j-i} C_j^* + \rho_s \sum_{j=1}^{j=i-1} 3(2\alpha)^{i-j} C_j^* + 2\rho_f \sum_{j=1}^{j=i} \alpha^{i-j} C_j^* \right).$$

By reversing the orders of summation, we can bound this by

$$4 \left(\frac{\rho_s}{1-\alpha} + \frac{6\alpha\rho_s}{(1-2\alpha)} + \frac{2\rho_f}{1-\alpha} \right) C^*.$$

This is our approximation relative to the near-optimum structured solution. Using Theorem 3.8 allows us to bound our overall approximation ratio by

$$\left(\frac{4}{2\alpha^2} \right) \left(\frac{\rho_s + 2\rho_f}{1-\alpha} + \frac{6\alpha\rho_s}{1-2\alpha} \right). \quad \square$$

The best known approximation ratio for Steiner trees is $\rho_s = 1.55$ due to [20], and that for Facility Location is $\rho_f = 1.52$ due to [18]. Setting $\alpha = 1/3$ the above reduces to a $54\rho_f + 135\rho_s < 292$ approximation for the incremental cost model.

5. Improved approximation algorithm for Access Network Design. Recall the definition of this problem from Definition 2.2. Andrews and Zhang [2] consider the case $c = 1/2$ and show that the optimal solution can be converted with a constant factor loss into a layered solution of shortest path forests. They show that there exists a near-optimal (within a constant multiplier on the cost) solution which is a tree satisfying the following properties:

1. Each demand is routed through pipes of consecutive types, i.e., types $1, 2, \ldots, \kappa$ ($\kappa \leq k$).
2. For all pipe types k, any pipe of that type has at least $u_k/2 = \frac{\sigma_k}{2\delta_k}$ amount of demand flowing through it.

This means that for Access Network Design, the optimal solution can be converted to a layered solution using shortest path forests of increasing pipe types.

We can improve the analysis of the above algorithm for Access Network Design. As shown in [12], for Access Network Design we have a layered shortest path forest solution with a reduction in cost at each layer. We can prove the following theorem.

THEOREM 5.1. *There exists a solution to the Access Network Design problem in which we use only pipe types $1, 2, \ldots$ satisfying the condition $\phi_i = \frac{\delta_{i+1}}{\delta_i} \leq \alpha$, and in which any pipe of type i has at least $u_i/2$ amount of demand flowing through it. The fixed and incremental costs of this solution are each within $\frac{1}{\alpha}$ of the original optimum which used all pipe types and which had at least $u_k/2$ demand in any pipe of type k.*

Proof. Note that since we are using pipes of larger types in increasing layers, the incremental cost δ per unit of traffic keeps decreasing. In fact, we can make sure that δ goes down by a constant fraction $\alpha < 1$ with a $\frac{1}{\alpha}$ increase in cost. The way we do this is the following.

Consider pipes of increasing types starting at type 1. Let $\phi_i = \frac{\delta_{i+1}}{\delta_i}$. Let k' be the largest number such that $\prod_{i=1}^{k'} \phi_i \geq \alpha$. We remove all pipe types $2, \ldots, k' + 1$ and use only pipes of type 1 instead of all these pipes. We next consider pipes starting at type $k' + 2$ (renumber this as pipe 2) and repeat this filtering process.

When the above is completed, we are left with a set of pipe types $1, 2, \ldots$ satisfying the following properties. For consecutive pipe types i and $i + 1$, $\frac{\delta_{i+1}}{\delta_i} \leq \alpha$. Finally, note that in this process, if a pipe of type j was discarded and replaced by a pipe of type i, it must be the case that $\sigma_i < \sigma_j$, and $\delta_i < \frac{1}{\alpha}\delta_j$, so that the cost of using pipe i is at most $\frac{1}{\alpha}$ times the cost of using pipe j. ☐

Recall that $\phi_i = \frac{\delta_{i+1}}{\delta_i}$. From above, we can assume with a loss of $\frac{1}{\alpha}$ in the approximation ratio that all $\phi_i \leq \alpha < 1$. Our algorithm will lay pipes in increasing order of types.

Let S_i denote the demand points at stage i. We maintain the invariant that every demand point has at least $u_i/6$ demand. We solve the load balanced Facility Location instance on S_i with lower bound u_{i+1} (except on the sink s). We route the demands to the open facilities using pipes of type i. For every open facility, we choose one of the demand points sending demand to it at random in proportion to its demand, and route all the demand to this point using pipes of type $i + 1$. Let S_{i+1} be the final set of demand points to which we route the demands. Note that every demand point has at least $u_{i+1}/6$ demand.

Let P_i^I be the routing cost at stage i, and let P_i^F be the fixed cost. Note that $P_i^F \leq 6P_i^I$ because of the invariant on the demands.

We define C_i^* to be the total incremental cost incurred by the optimal solution using pipes of type i. Note that the total cost of the optimal solution is $C^* \geq \sum_i C_i^*$.

LEMMA 5.2. $\mathbf{E}[P_i^I] \leq 2\rho_f(1 + \alpha)(\sum_{j=1}^{j=i-1} \alpha^{i-j-1}C_i^*)$.

Proof. The routing cost that the optimum solution pays in routing the original demand points until stage i using pipes of type i is at most $\sum_{j=1}^{j=i-1} \alpha^{i-j-1}C_i^*$. This follows from [12] and from the analysis in section 4. This is an instance of the load balanced Facility Location problem, and we apply Theorem 3.1. ☐

It is now easy to see the following.

LEMMA 5.3. $\mathbf{E}[\sum_i (P_i^I + P_i^F)] \leq 14\rho_f \frac{1+\alpha}{1-\alpha}C^*$.

Note that we lost a factor of $\frac{1}{\alpha}$ up front in the routing cost because of scaling the pipe types. Our approximation ratio is therefore $\frac{1}{\alpha}14\rho_f\frac{1+\alpha}{1-\alpha}$. Setting $\alpha = 1/3$ and $\rho_f = 1.52$ this ratio is less than 128.

THEOREM 5.4. *We have a randomized* 128 *approximation for Access Network Design.*

This approximation factor has been subsequently improved in [15].

6. Load balanced Facility Location. Recall the definition of Facility Location and load balanced Facility Location from section 3.1. The load balanced problem differs from standard Facility Location [22] in that we must route at least L_i units of demand to each open facility i. Load balanced Facility Location has direct applications; consider a franchise which must open stores to minimize the average distance from customer to store, but which must also guarantee a minimum number of customers to each store so the individual stores remain profitable. We present a constant approximation to this problem, losing a constant factor compared to the lower bound on demand. We can write an IP for this problem.

$$\text{Minimize} \sum_i \sum_j d_j c_{ij} x_{ij} + \sum_i \phi_i y_i,$$

$$\begin{aligned}
\sum_i x_{ij} &\geq 1 & \forall j, \\
x_{ij} &\leq y_i & \forall i, j, \\
\sum_j d_j x_{ij} &\geq L_i y_i & \forall i, \\
x_{ij}, y_i &\in \{0, 1\} & \forall i, j.
\end{aligned}$$

Clearly, the general version of this problem is NP-hard, as it reduces to classical facility location when the lower bounds are set to zero. In fact, this problem is NP-hard even if all facility costs are zero, all lower bounds are equal, and all demands are unit.

THEOREM 6.1. *Suppose we are given a load balanced Facility Location instance with lower bound L on all facility locations, and facility costs being zero. Deciding if a feasible solution of cost at most C exists is NP-hard.*

Proof. We reduce the decision version of the unweighted set cover problem to an instance of this problem as follows. The sets are the facilities. The elements are the demand points with unit demand. Suppose there are n elements. In the Facility Location instance, we add edges of cost one between every element and all the sets it belongs to.

Suppose we have to decide if a cover with $s > 1$ sets exists. We add sn demand points with unit demand and connect them to all the sets (or facilities) with edges of length one. We set the lower bound on the facilities to be $n + 1$. We now ask if there is a feasible solution of cost no more than $ns + n + s$. Note that if there exists a set cover of size s, then there exists a solution of cost $ns + n + s$. The reverse also holds, and therefore this completes the reduction. ☐

DEFINITION 6.2. *An approximation algorithm for load balanced Facility Location is an (α, β) approximation for some $\alpha \geq 1$ and $\beta \leq 1$ if the cost of the solution is within α times the optimal cost and facility i, if opened, serves at least $L_i \beta$ demand.*

Let us denote by ρ_f the best known approximation ratio for classical facility location, which is $\rho_f = 1.52$ due to [18]. We present a $(2\rho_f, 1/3)$ approximation to this problem. The same result was independently obtained by [16]. Unlike classical facility location [22], the lower bound makes it hard to round the linear relaxation directly. This arises from the fact that the filtering steps of Lin and Vitter in [17] do not work. Thus fractional solutions cannot be rounded by previous approaches.

The algorithm. The algorithm proceeds in two basic steps and uses an approximation algorithm for the Facility Location problem. We note that the approximation guarantee holds relative to the LP relaxation as well (albeit with more technical details that we omit, since it is not the main focus of the paper).

Load Balanced Facility Location Algorithm

(i) *Transformation.* For facility i, add the cheapest way to route exactly L_i units of demand to i to the facility cost ϕ_i. To do this, consider demands in increasing order of distance from i, and route these demands to i until exactly L_i units have been routed. The routing cost of this process is added to ϕ_i.

(ii) *Facility location.* Next solve regular Facility Location with these facility costs using the ρ_f-approximation algorithm.

(iii) *Rounding to remove facilities.* Now consider the open facilities in arbitrary order. Consider any open facility i that serves less than $L_i/3$ amount of demand. Close the facility and route the demands it serves to their closest open facilities.

LEMMA 6.3. *Consider any feasible solution to the load balanced Facility Location problem of cost C. After the transformation in step (i), this yields a feasible instance*

of the regular Facility Location problem of cost at most $2C$.

Proof. Consider any facility i opened by the load balanced solution. Since this solution is routing at least L_i amount of demand to any open facility, the facility cost we assign in the new problem is at most the routing cost of the demand connected to that facility. Thus the total additional facility cost is at most C.   ☐

Therefore, the total cost in the solution we compute is bounded in terms of the cost of the original solution to within a factor of $2\rho_f$. Also note that facility location guarantees that each demand point goes to the closest open facility. We have to show that removing a facility does not increase the total facility plus routing cost of the solution. For this, we show a feasible way to route the demands it serves so that the cost does not increase.

LEMMA 6.4. *Removing a facility i serving less than $\frac{L_i}{3}$ amount of demand cannot increase the cost of our solution.*

Proof. Suppose we are closing facility i. Consider the closest demand point j which does not send demand to this facility. Suppose $c_{ij} = D$. If j is being served by i', $c_{i'j} < D$, as each demand point goes to the closest open facility.

Note that at least $2L_i/3$ units of demand are at distance D or greater. Therefore, $f_i' \geq \frac{2L_i}{3}D$.

When we close the facility, we can afford to use f_i' toward rerouting the demand it serves. We send the demand to i', the facility serving j. The extra cost for doing this is at most the cost of taking the demand from i to j and from there to i'. This distance is at most $2D$ by the metric property, and the demand is at most $\frac{L_i}{3}$, and so the total rerouting cost is at most $\frac{2L_i}{3}D$.   ☐

The above can be summarized in the following theorem.

THEOREM 6.5. *The load balanced Facility Location problem has a $(2\rho_f, 1/3)$ approximation where each demand is served by its closest open facility.*

7. Conclusion. In this paper, we presented the first constant factor approximation for the single sink buy-at-bulk network design problem. We conclude by surveying the results on this and related problems that have appeared since the publication of the preliminary version of this paper [13]. First, the algorithm itself has been significantly improved and simplified. Gupta, Kumar, and Roughgarden [15] obtain a 72.8 approximation by combining the Steiner and shortest path stages into a "Rent-or-Buy" stage, and using a novel analysis. This is the current best known approximation guarantee. For the single sink case, Goel and Estrin [11] consider simultaneous (oblivious) approximation over all concave functions and obtain an $O(\log n)$ approximation.

For the multiple sink (source-sink pairs) version of this problem, as mentioned earlier, the best known approximation ratio of $O(\log n)$ follows directly from tree embeddings [3, 9]. For this version, Andrews [1] has shown an $\Omega((\log n)^{\frac{1}{4}})$ hardness of approximation assuming $NP \not\subseteq \mathrm{DTIME}(n^{\log^{O(1)} n})$.

The nonuniform version of the problem assumes different cable types are available on different edges. As mentioned earlier, the best known approximation ratio [19] for the single sink version is $O(\log n)$. Chuzhoy et al. [8] show that the single sink version is hard to approximate within $\Omega(\log \log n)$, under similar hardness assumptions as the uniform case. Charikar and Karagiozova [6] consider the nonuniform version in the presence of multiple source-sink pairs. The best result for this case is a polylogarithmic approximation ratio and is achieved by Chekuri et al. [7].

REFERENCES

[1] M. ANDREWS, *Hardness of buy-at-bulk network design*, in Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, 2004, pp. 115–124.

[2] M. ANDREWS AND L. ZHANG, *The access network design problem*, in Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, 1998, pp. 40–49.

[3] B. AWERBUCH AND Y. AZAR, *Buy-at-bulk network design*, in Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, 1997, pp. 542–47.

[4] Y. BARTAL, *On approximating arbitrary metrics by tree metrics*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, ACM, New York, 1998, pp. 161–168.

[5] M. CHARIKAR, C. CHEKURI, A. GOEL, S. GUHA, AND S. PLOTKIN, *Approximating a finite metric by a small number of tree metrics*, in Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, 1998.

[6] M. CHARIKAR AND A. KARAGIOZOVA, *On non-uniform multicommodity buy-at-bulk network design*, in Proceedings of the ACM Symposium on Theory of Computing, ACM, New York, 2005, pp. 176–182.

[7] C. CHEKURI, M. T. HAJIAGHAYI, G. KORTSARZ, AND M. R. SALAVATIPOUR, *Approximation algorithms for non-uniform buy-at-bulk network design*, in Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, 2006, pp. 677–686.

[8] J. CHUZHOY, A. GUPTA, J. NAOR, AND A. SINHA, *On the approximability of network design problems*, in Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2005, pp. 943–951.

[9] J. FAKCHAROENPHOL, S. RAO, AND K. TALWAR, *A tight bound on approximating arbitrary metrics by tree metrics*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, ACM, New York, 2003, pp. 448–455.

[10] N. GARG, R. KHANDEKAR, G. KONJEVOD, R. RAVI, F. S. SALMAN, AND A. SINHA, *On the integrality gap of a natural formulation of the single-sink buy-at-bulk network design problem*, in Proceedings of the 8th International IPCO Conference on Integer Programming and Combinatorial Optimization, Springer-Verlag, London, 2001, pp. 170–184.

[11] A. GOEL AND D. ESTRIN, *Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2003, pp. 499–505.

[12] S. GUHA, A. MEYERSON, AND K. MUNAGALA, *Hierarchical placement and network design problems*, in Proceedings of the 41st Annual IEEE Symposium on Foundation of Computer Science, IEEE Computer Society, Washington, DC, 2000, pp. 603–612.

[13] S. GUHA, A. MEYERSON, AND K. MUNAGALA, *A constant factor approximation for the single sink edge installation problems*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, ACM, New York, 2001, pp. 383–388.

[14] S. GUHA AND K. MUNAGALA, *Generalized clustering*, in Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2002, pp. 484–485.

[15] A. GUPTA, A. KUMAR, AND T. ROUGHGARDEN, *Simpler and better approximation algorithms for network design*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, ACM, New York, 2003, pp. 365–372.

[16] D. KARGER AND M. MINKOFF, *Building Steiner trees with incomplete global knowledge*, in Proceedings of the 41st Annual IEEE Syposium of Foundations of Computer Science, IEEE Computer Society, Washington, DC, 2000, pp. 613–623.

[17] J.-H. LIN AND J. S. VITTER, ϵ-*approximations with minimum packing constraint violations*, in Proceedings of the 24th Annual ACM Symposium on Theory of Computing, ACM, New York, 1992, pp. 771–782.

[18] M. MAHDIAN, Y. YE, AND J. ZHANG, *Improved approximation algorithms for metric facility location problems*, in Proceedings of the 5th International Workshop on Approximation Algorithms for Combinatorial Optimization, Spinger-Verlag, London, 2002, pp. 229–242.

[19] A. MEYERSON, K. MUNAGALA, AND S. PLOTKIN, *Cost-distance: Two metric network design*, in Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Washington, DC, 2000, pp. 624–630.

[20] G. ROBINS AND A. ZELIKOVSKY, *Improved Steiner tree approximation in graphs*, in Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 2000, pp. 770–779.

[21] F. S. SALMAN, J. CHERIYAN, R. RAVI, AND S. SUBRAMANIAN, *Buy-at-bulk network design: Approximating the single-sink edge installation problem*, in Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms, ACM, New York, SIAM, Philadelphia, 1997, pp. 619–628.

[22] D. B. SHMOYS, É. TARDOS, AND K. AARDAL, *Approximation algorithms for facility location problems*, in Proceedings of the 29th Annual ACM Symposium on Theory of Computing, ACM, New York, 1997, pp. 265–274.

[23] K. TALWAR, *The single-sink buy-at-bulk LP has constant integrality gap*, in Proceedings of IPCO, Spinger-Verlag, London, 2002, pp. 475–486.