Improved Combinatorial Algorithms for Single Sink Edge Installation Problems

Sudipto Guha* Adam Meyerson† Kamesh Munagala‡

June 2, 2000

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 $O(\log |S|)$, where S is the set of sources. We also present an improved constant approximation to the related Access Network Design problem. Our algorithms are randomized and fully combinatorial. They can be derandomized easily at the cost of a constant factor loss in the approximation ratio.

*Department of Computer Science, Stanford University CA 94305. Research Supported by IBM Research Fellowship, NSF Grant IIS-9811904 and NSF Award CCR-9357849, with matching funds from IBM, Mitsubishi, Schlumberger Foundation, Shell Foundation, and Xerox Corporation. Email: `sudipto@cs.stanford.edu`.

†Department of Computer Science, Stanford University CA 94305. Supported by ARO DAAG-55-97-1-0221. Email: `awm@cs.stanford.edu`.

‡Department of Computer Science, Stanford University CA 94305. Supported by ONR N00014-98-1-0589. Email: `kamesh@cs.stanford.edu`.

1 Introduction

The problem of designing networks using trunks to route demands has received considerable attention. In this problem, commonly known as *Buy-at-Bulk Network Design* [11, 2, 1, 10], we are given demands at nodes in a network which have to be routed to their respective destinations using pipes of certain capacities and costs per unit length. The costs obey economies of scale, in the sense that it is cheaper to buy a pipe of larger capacity than many pipes (which sum to the same capacity) of a smaller capacity. The goal is to optimize the total cost of the pipes we buy to route the demands. Andrews and Zhang [1] define a special case of this problem called the *Access Network Design* problem, where all demands need to be routed to a central core network and the costs and capacities of the pipes obey certain common constraints. They show applications of this problem in designing telephone networks.

The problem of *buy-at-bulk* network design was first defined in [11]. Awerbuch and Azar [2] obtain an $O(\log^2 n)$ approximation to this problem even when all demands have different sinks. Their work is based on techniques to approximate any metric by tree metrics [4]. The approximation factor can be improved to $O(\log n \log \log n)$ using Bartal's result in [5] and derandomized using the results of [6, 7]. For *single sink* buy-at-bulk, Salman et al [11] show a constant approximation when there is only a single pipe type. Their method is based on the technique of balancing Steiner and shortest path trees [3, 9]. Andrews and Zhang [1] define the Access Network Design problem, which is a special case of single sink buy-at-bulk where the pipe costs and capacities obey certain common constraints, and provide an $O(K)$ approximation, where K is the number of pipe types. Guha et al [8] give a constant approximation to this problem. Meyerson et al [10] provide an $O(\log n)$ approximation to single sink network design where the costs on the edges are arbitrary non-decreasing concave functions of demand.

We present a constant approximation to the single sink buy-at-bulk network design problem, improving the previous result of $O(\log n)$. Our algorithm is randomized and combinatorial, but can be derandomized easily. We also present an improved constant approximation to Access Network Design.

In Section 2, we state the single sink buy-at-bulk problem formally, and discuss some structural properties of the optimal solution. In Section 3, we discuss a scaling idea to remove similar pipe types, and show how it improves the structure of the optimum solution in Section 4. We then present the constant factor approximation algorithm in Section 5 and analyze it. We show how to improve the approximation ratio for Access Network Design in Section 6. We conclude by mentioning some open problems.

2 Problem Statement

We are asked to construct a network on an underlying graph of distances. We are given K types of connections (pipes) each with a fixed cost and a capacity. The cost of placing a pipe of fixed cost σ_k along a path of length L will be $\sigma_k L$. We are given a set S_1 of demand nodes and a single sink s. Each demand node $v \in S_1$ needs to transport some amount of demand d_v to the sink. We are asked to buy a set of pipes as cheaply as possible so as to route all demands to the sink. We are allowed to buy multiple copies of a pipe along the same link.

We will use an alternate formulation of this problem, introduced by Andrews and Zhang. Instead of each pipe having a capacity u_k, the pipes will have incremental cost defined by $\delta_k = \sigma_k/u_k$. This represents the per-unit-flow cost of the pipe. If we transport d units of demand along a path of length L using pipe k, we will pay a total of $L(\sigma_k + \delta_k d)$. It's not hard to see that a solution under this 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.

If we number the pipes in increasing order of capacity, we observe the following conditions: $\sigma_1 < \sigma_2 < \cdots < \sigma_K$, and $\delta_1 > \delta_2 > \cdots > \delta_K$. We will define $f_k(D) = \sigma_k + \delta_k D$ to be the per-unit-distance cost of routing demand D along a pipe of type k. The capacity of pipe k is $u_k = \sigma_k/\delta_k$.

The Access Network Design problem is a special case of Single Sink buy-at-bulk with additional restrictions on the costs of the pipe types. The main restriction is that a type k pipe is cheaper only when it routes significant demand. Formally, the restrictions can be stated as follows:

1. For $2 \leq k \leq K$, if $d < \frac{\sigma_k}{\delta_k}$, then $d\delta_{k-1} + \sigma_{k-1} < d\delta_k + \sigma_k$. For this to make any physical sense, we would actually require $d < \beta\frac{\sigma_k}{\delta_k}$ for some $\beta < 1$. Since it will not effect our proofs, we will simplify our notation by assuming $\beta = 1$.
2. The smallest demand looks like the smallest pipe capacity, or more precisely, $d \cdot \delta_1 > \sigma_1$.
3. $\sum_{\kappa<k} \sigma_\kappa = O(\sigma_k)$.

The reason for enforcing these properties will become clear when we discuss the structure of the optimum solution in Section 6.

2.1 Layered Structure of Optimum Solution

First, 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 γ_k be the demand for which it becomes cheaper to use a pipe of type $k+1$ compared to a pipe of type k. Assume without loss of generality that $0 = \gamma_0 < u_1 < \gamma_1 < u_2 < \gamma_2 < \cdots < u_K < \gamma_K = \infty$.

Observe now that if the demand amount is in the range $[\gamma_{i-1}, u_i]$, 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 $[u_i, \gamma_i]$, 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 γ_{i-1} demand and each pipe in the shortest path forest has at least u_i amount of demand.

3 Removing Similar Pipes

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. We will eliminate various pipe types in order to guarantee the following conditions for some positive $\alpha < 1/2$.

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 3.1 *We can 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.*

Proof: 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 higher pipe type than k is 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. ■

A similar process allows us to guarantee the second condition.

Lemma 3.2 *We can eliminate pipes in order 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 lemmas, we can guarantee the two conditions with only a constant increase in the cost of the solution.

Theorem 3.1 *There exists a solution which uses only the remaining pipes after elimination, and which has cost at most $1/\alpha$ times the cost of the original optimum solution.*

4 Properties of the Near-Optimum Solution

We will define b_k to be such that $f_{k+1}(b_k) = 2\alpha f_k(b_k)$. In essence, b_k is sufficient demand that it becomes considerably cheaper to use a pipe of type $k+1$ rather than a pipe of type k. We first show that $u_k \leq b_k \leq u_{k+1}$.

Lemma 4.1 $b_k \leq u_{k+1}$.

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:

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

. ■

Lemma 4.2 $b_k \geq u_k$

Proof: 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 $b_k > u_k$. ■

Lemma 4.3 *For 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 $f_{k+1}(D) \leq 2\alpha f_k(D)$. ■

There exists a near-optimum solution which uses pipe type $k+1$ only if at least b_k demand is being routed. This solution also routes all demand which enters a node using pipes of type k out of that node using pipes of types k or $k+1$. This structural observation about a nearly (within constant factor) optimum solution will be important in our proof of the approximation ratio.

Theorem 4.1 *There exists a solution which uses pipe type $k+1$ on a link only if at least b_k demand is being routed across that link, and which routes all demand which entered a node using pipe k out of that node using pipes k and $k+1$. This solution pays at most $\frac{2}{\alpha} + 1$ times the optimum.*

Proof: We consider the nodes of the optimum solution tree from the bottom up. Suppose a node has flow outgoing on a pipe of type k. We can conclude that all incoming flow was on pipes of type k or less, since otherwise we could improve the optimum solution by changing one of the pipe types. Consider the flow incoming on pipes of type i in increasing order of i. Either the total flow incoming on pipes of type i is at least b_i or it is not. If it is at least b_i, then we add a pipe of length zero from this node to itself; this pipe has type $i+1$ and carries the flow which was incoming on pipes of type i. Adding this pipe does not increase the cost of the solution, since the pipe has length zero. If there is not b_i demand incoming on pipes of type i, then we add a new pipe of type i from the node to its parent which will carry all the flow which was incoming on pipes of type i. The total flow traveling from this node to its parent has not changed. We can see that the new solution constructed will have the desired properties; we must guarantee that it is within a constant of optimum. Consider an edge in the optimum tree. The original optimum placed a pipe of type k here. We may have placed an additional pipe of each type 1 through $k-1$ along this edge. The pipe of type i routes at most b_i flow. The total cost of these additional pipes is therefore at most $\sum_{i=1}^{i=k-1} f_i(b_i)$. Using Lemma 4.1 and the definition of b_i, we can guarantee that

$f_i(b_i) = \frac{1}{2\alpha} f_{i+1}(b_i) \leq \frac{1}{2\alpha} f_{i+1}(u_{i+1}) = \frac{2\sigma_{i+1}}{2\alpha}$. Substituting this into the equation, the additional pipe cost is at most $\frac{2}{2\alpha} \sum_{i=2}^{i=k} \sigma_i$. Because each fixed cost is at most half the next higher fixed cost, we can bound this sum by $\frac{4}{2\alpha}\sigma_k$ and the total cost of the solution has increased by at most a factor of $\frac{2}{\alpha} + 1$.

This breaks down only at the sink itself, where we may have less than b_k flow arriving on pipes of type k yet there is no "parent node" to which we can route. Suppose that this is the case, but that the total flow in the network is at least b_k (if the total flow in the network is less than b_k we can simply discard pipes of type higher than k). Since not all the flow has reached the sink on pipes of type k, it follows that somewhere there exists a pipe of type $k+1$. This means some location gathered at least b_k flow. Suppose we route downward from the sink along the graph to this location using pipes of type k. If any edge now has pipes of type k traveling in both directions, we simply remove the pipe towards the sink. We stop when we reach a node where we have more than b_k flow; this must happen eventually because we're routing towards a node with sufficient flow. We maintain the property that we have introduced at most one additional pipe of type k on top of pipes of type $k+1$ and up, and that these pipes route at most b_k flow. We have already accounted for the cost of this modification. ■

We will define C_k^* to be the total cost which this near-optimum solution pays using pipes of type k. The total cost of the solution is therefore $\sum_{k=1}^{k=K} C_k^* = C^*$.

5 The Algorithm

We will now present an 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 against the respective costs of the optimum solution, similar to the analysis used in [8] for Access Network Design.

Let s denote the sink node. Our algorithm constructs forests in layers. We will illustrate the construction for layer i. Let S_i be the set of demand points we have at this stage. S_1 is the original set of demand points. Layer i will use pipe type i exclusively.

We will use the load balanced facility location problem [8] as a sub-routine below. This problem is a variant of the classical facility location problem, where we have a lower bound on the amount of demand any open facility must serve. We can approximate this to a constant factor of μr provided we relax the lower bound by factor $\beta = \frac{\mu-1}{\mu+1}$. Here, r is the best known approximation for facility location, and can be taken as 1.728.

Steiner Trees Construct a Steiner tree on S_i. The edge cost per unit length is σ_i. Transport the demands from S_i upwards 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.

Consolidate Consider any root in this forest. This has at least u_i amount of demand coming to it from nodes in S_i. Let the set of demand points sending demand to some root j be S_{ij}. Pick a node at random from S_{ij} in proportion to its demand and send all the demand at j to this node.

Shortest Path Trees We solve a load balanced facility location instance on S_1 with the facility lower bound b_i on all nodes and the edge cost per unit length δ_i. 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.

Consolidate Consider any root in this forest. Some set of nodes from S_1 were assigned to this root, and their (original) total demand is at least βb_i. We choose one such node at random, in proportion to their original demands. We send all the demand from the root to the chosen node We set S_{i+1} to these new demand locations.

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.

5.1 Analysis

We define d_v to be the demand of node v in the original (S_1) demands. We define D_v to be the demand at node v in the current stage of the algorithm.

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 $\sum_i (T_i + P_i + N_i)$.

At each layer we will construct an overall solution to the problem on the nodes S_i. Let $C_i(j)$ represent the total cost which this solution on the nodes S_i pays using pipes of type j.

Lemma 5.1 *At the end of any consolidation step, every node has* $\mathbf{E}[D_v] = d_v$.

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.

Suppose that the demand at node v after the previous consolidation step was x_v. By the induction hypothesis, $\mathbf{E}[x_v] = d_v$. There are two cases to consider; either we performed a Steiner Tree step or a Shortest Path Tree step.

Suppose we have just performed a Steiner Tree step. The current node is routed to some root with total demand D. We then choose a node for consolidation. The probability that we choose node v is x_v/D. If we choose v, demand D will be placed there; otherwise no demand can be placed there. Thus the expected amount of demand placed at v is x_v; by induction we can claim that $\mathbf{E}[D_v] = \mathbf{E}[x_v] = d_v$ as desired.

Suppose we have just performed a Shortest Path Tree step. The current node is routed to some root which would have demand D if the demands were as in the S_1 stage. The probability we consolidate to v is d_v/D; if we do, the total demand at v will be the total of the current demands of all the nodes routed to this root. Let V be the set of nodes routed to v. $\mathbf{E}[D_v]$ is therefore $\mathbf{E}[\frac{d_v}{D} \sum_{w \in V} x_w]$. Observe now that:

$$\mathbf{E}[\sum_{w \in V} x_w] = \sum_{w \in V} \mathbf{E}[x_w] = \sum_{w \in V} d_w = D$$

Thus $\mathbf{E}[D_v] = d_v$. ■

Lemma 5.2 $\mathbf{E}[N_i] \leq T_i + P_i$.

Proof: The consolidation step following a tree construction always has expected cost at most the cost of the tree construction. ■

Lemma 5.3 $\mathbf{E}[P_i^I] \leq \mu r \sum_{j=1}^{j=i} \alpha^{i-j} C_j^*$.

Proof: Suppose the demands at the sources were those from S_1. Then one possible solution would be the optimum problem solution up until pipes of type $i+1$ were used. We know that the optimum solution must gather the desired b_i flow before using pipes of type $i+1$. It follows that we can find a solution with cost at most μr times the incremental cost of the optimum using pipes of type 1 through i. 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 expected value equal to the original demand, so the expected value of P_i^I is bounded as above. ■

Lemma 5.4 $P_i^F \leq P_i^I$.

Proof: The Steiner Tree stage guarantees at least u_k demand or zero everywhere. 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. ■

Lemma 5.5 *Let D_v be the demand at $v \in S_i$. Then, $\mathbf{E}[D_v] \geq \beta b_{i-1}$.*

Proof: We obtain the nodes S_i by solving a load balanced facility location instance on S_1 with lower bounds b_{i-1}. In this solution, each node in S_i except s has demand at least βb_{i-1}. Consider any node w in S_1, and suppose that the demand our solution so far has there is x_w. Then, $\mathbf{E}[x_w] = d_w$. Let V be the set of nodes which get routed to v. Then, $\mathbf{E}[D_v] = \mathbf{E}[\sum_{w \in V} x_w] = \sum_{w \in V} \mathbf{E}[x_w] = \sum_{w \in V} d_w \geq \beta b_{i-1}$. ■

Lemma 5.6 *At stage i we can construct a solution which uses only pipes i and higher. This solution has cost $C_i(j)$ using pipes of type j, where $\mathbf{E}[C_i(j)] \leq C_j^*$ for $j > i$, $C_i(j) = 0$ for $j < i$, and $\mathbf{E}[C_i(i)] \leq \sum_{j=1}^{j=i} \frac{1}{\beta}(2\alpha)^{i-j} C_j^*$.*

Proof: For $i = 1$ we use the near-optimum solution itself and the claim follows immediately.

Consider stage i. If we use the pipes as in the near-optimum solution, our expected cost using each pipe type j would be equal to C_j^*. For each pipe of type $j < i$, we remove the pipe if the total demand flowing across it is zero. Otherwise we replace the pipe with a pipe of type i. The cost of this replacement pipe is $f_i(D)$ where D is the demand flowing across it. Given that the demand is nonzero, the node must lie along the path from one of the chosen consolidation nodes from the previous stage. Each of these nodes has expected demand at least βb_{i-1}. It follows that $\mathbf{E}[f_i(D)] \leq \frac{1}{\beta}(2\alpha)^{i-j} f_j(D)$. We can therefore bound the cost of this modified solution using pipes of type i by an expected $\sum_{j=1}^{j=i} \frac{1}{\beta}(2\alpha)^{i-j} C_j^*$. ■

Lemma 5.7 $\mathbf{E}[T_i^F] \leq 2 \sum_{j=i+1}^{j=K} \alpha^{j-i} C_j^* + 2 \sum_{j=1}^{j=i} \frac{1}{\beta}(2\alpha)^{i-j} C_j^*$.

Proof: The solution given in Lemma 5.6 is one possible Steiner tree. The fixed cost of this Steiner tree is bounded by the following expected cost:

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

This holds because the cost on a pipe of type $i+k$ will be reduced by α^k since we pay only for a pipe of type i. We can find a Steiner Tree of at most twice this cost, so the claim follows. ∎

Lemma 5.8 $T_i^I \leq T_i^F$.

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. ∎

Theorem 5.1 *There is a constant-approximation for single-sink buy-at-bulk.*

Proof: The total cost of our solution is bounded by $\sum_i 2(2T_i^F + 2P_i^I)$. Using Lemmas 5.3 and 5.7, we conclude that the expected cost of our solution is bounded by the following:

$$4\sum_i (2\sum_{j=i+1}^{j=K} \alpha^{j-i} C_j^* + 2\sum_{j=1}^{j=i} \frac{1}{\beta}(2\alpha)^{i-j} C_j^* + \mu r \sum_{j=1}^{j=i} \alpha^{i-j} C_j^*)$$

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

$$4(\frac{2}{1-\alpha} + \frac{2}{\beta(1-2\alpha)} + \frac{\mu r}{1-\alpha})C^*$$

This is our approximation against the near-optimum solution. Using Theorem 4.1 allows us to bound our overall approximation ratio by:

$$(\frac{4}{\alpha})(1+\frac{2}{\alpha})(\frac{2}{1-\alpha} + \frac{2}{\beta(1-2\alpha)} + \frac{\mu r}{1-\alpha})$$

∎

6 Improved Algorithm for Access Network Design

For the Access Network Design problem, Andrews and Zhang [1] show a strong property. 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 $u_k = \frac{\sigma_k}{\delta_k}$ 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 will therefore compare ourselves against the optimal solution that satisfies the above mentioned structural properties.

We can improve the analysis of the algorithm in Section 5 for Access Network Design. As shown in [8], for the Access Network Design we have a layered solution with a reduction in cost at each layer. We can prove the following theorem:

Theorem 6.1 *There exists a solution to the Access Network Design problem in which we only use pipe types satisfying the condition $\alpha_i = \frac{\delta_{i+1}}{\delta_i} \leq \alpha$, and in which any pipe of type i has at least u_i 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 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 $\alpha_i = \frac{\delta_{i+1}}{\delta_i}$. Let k' be the largest number such that $\prod_{i=1}^{k'} \alpha_i \geq \alpha$. We remove all pipe types $2, \ldots, k'+1$ and use only pipe of type 1 instead of all these pipes. We next consider pipes starting at type $k'+2$ and repeat this filtering process. This is the same as in hierarchical placement problems.

When the above is completed, we are left with a set of pipe types satisfying the following properties. For consecutive pipe types i and $i+1$, $\frac{\delta_{i+1}}{\delta_i} \leq \alpha$. If we used a pipe of type i instead of a pipe of type j, then $\delta_j > \alpha\delta_i$ and $\sigma_j > \sigma_i$. ■

Let $\alpha_k = \frac{\delta_k}{\delta_{k-1}}$. We can assume with a loss of $\frac{1}{\alpha}$ in the approximation ratio that all $\alpha_k \leq \alpha < 1$. 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 βu_i 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 where we route the demands. Note that every demand point has at least βu_{k+1} demand.

Let P_i^I be the routing cost at stage i, and let P_i^F be the fixed cost. Note that $P_i^F \leq \frac{1}{\beta} P_i^I$ 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 6.1 $\mathbf{E}[P_i^I] \leq \mu r(1+\alpha)(\sum_{j=1}^{j=i-1} \alpha^{i-j-1} C_i^*)$.

Proof: The routing cost that the optimum solution pays in routing the original demand points till 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 [8] and from the analysis in Section 5. This is an instance of the load balanced facility location problem, and so the expected cost of our solution is within μr times this solution. For routing back to randomly chosen nodes, we pay α times this cost in the expected sense, as we use a pipe of larger type. ∎

It is now easy to see the following.

Lemma 6.2 $\mathbf{E}[\sum_i (P_i^I + P_i^F)] \le (1+\frac{1}{\beta})\mu r \frac{1+\alpha}{1-\alpha} C^*$.

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 $\frac{2r\mu^2}{\mu-1}\frac{1+\alpha}{\alpha(1-\alpha)}$. Choosing $\mu = 2$ and $\alpha = \sqrt{2}-1$, we have a 80.566 approximation.

Theorem 6.2 *We have a randomized constant approximation for Access Network Design.*

7 Derandomization

The algorithms mentioned above can be derandomized as follows. Instead of constructing the trees starting from S_1, we construct it from the nodes in S_{i-1} and do not route back. The cost we pay in layer i has geometrically decreasing contribution from previous layers. We omit the details.

8 Open Problems

It would be interesting to see if we can do something better than $O(\log n \log\log n)$ on chosen pairs buy-at-bulk network design, where the sink node for each demand point could be different. Another interesting direction is to see if the techniques here work even if only a subset of pipe types are available on each edge.

References

[1] Matthew Andrews and Lisa Zhang. The access network design problem. *39th IEEE Symposium on Foundations of Computer Science*, pages 40–49, 1998.

[2] B. Awerbuch and Y. Azar. Buy-at-bulk network design. *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 542–47, 1997.

[3] B. Awerbuch, A. Baratz, and D. Peleg. Cost-sensitive analysis of communication protocols. *Proceedings of the Ninth Annual ACM Symposium on Principles of Distributed Computing*, pages 177–87, 1990.

[4] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. *37th IEEE symposium on Foundations of Computer Science*, pages 184–193, 1996.

[5] Y. Bartal. On approximating arbitrary metrics by tree metrics. *30th ACM Symposium on Theory of Computing*, 1998.

[6] M. Charikar, C. Chekuri, A. Goel, and S. Guha. Rounding via trees: Deterministic approximation algorithms for group steiner trees and k-median. *30th ACM Symposium on Theory of Computing*, 1998.

[7] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin. Approximating a finite metric by a small number of tree metrics. *39th IEEE Symposium on Foundations of Computer Science*, 1998.

[8] Sudipto Guha, Adam Meyerson, and Kamesh Munagala. Hierarchical placement and network design problems. *Stanford University Tech. Note., STAN-CS-TN-00-93*, 2000.

[9] S. Khuller, B. Raghavachari, and N. Young. Balancing minimum spanning and shortest path trees. *Algorithmica*, 14(4):305–321, 1994.

[10] Adam Meyerson, Kamesh Munagala, and Serge Plotkin. COST-DISTANCE: Two metric network design. *Stanford University Tech. Note., STAN-CS-TN-00-92*, 2000.

[11] F. S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian. Buy-at-bulk network design: Approximating the single-sink edge installation problem. *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 619–628, 1997.