Designing Networks Incrementally

Adam Meyerson* Kamesh Munagala† Serge Plotkin ‡

Abstract

We consider the problem of incrementally designing a network to route demand to a single sink on an underlying metric space. We are given cables whose costs per unit length scale in a concave fashion with capacity. Under certain natural restrictions on the costs (called the Access Network Design *constraints), we present a simple and efficient randomized algorithm that is competitive to the minimum cost solution when the demand points arrive* online. *In particular, if the order of arrival is a random permutation, we can prove a $O(1)$ competitive ratio. For the fully adverserial case, the algorithm is $O(K)$-competitive, where K is the number of different pipe types. Since the value of K is typically small, this improves the previous $O(\log n \log\log n)$-competitive algorithm which was based on probabilistically approximating the underlying metric by a tree metric. Our algorithm also improves the best known approximation ratio and running time for the offline version of this problem.*

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* [14, 3, 2, 13], 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 [2] 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 there are some natural restrictions on the costs of the pipes. They show applications of this problem in designing telephone and data networks. We will focus on this problem in this paper.

An important consideration in provisioning networks is that it must be done *incrementally*. New demand constantly arrives, and the existing infrastructure needs to be upgraded to cope with the larger demand. The goal now is to construct a network and provision cables so that each existing demand point has a valid path to the sink at all instants of time. Additinally, the path of an existing demand point should not change because of new arrivals. We are allowed to buy more cables on existing paths to route new demand, but we are not allowed to remove cables we already laid, or re-route demand paths. In other words, the algorithm must be *online*, and competitive against the optimal solution on the existing subset of demand points at every time instant.

We present a simple randomized algorithm for the online version of the Access Network Design problem. This algorithm is constant-competitive if the demand points arrive in random order (implying any permutation is equally likely), and $O(K)$ competitive for adverserial arrival of demand points, where K is the number of pipe types (this number is typically small). This algorithm improves the best known online algorithm for this problem both in terms of approximation ratio and running time.

In addition, the offline version of this algorithm has a better approximation ratio and running time than the best known algorithm for Access Network Design due to Guha et al [10], which was based on facility location techniques. The most complicated sub-routine in our algorithm is finding the nearest neighbor.

Background: The problem of *buy-at-bulk* network design was first defined in [14]. Awerbuch and Azar [3] obtain an $O(\log^2 n)$ approximation to this problem, where n is the number of demand points. Their work is based on techniques to approximate any metric by tree metrics [5]. The approximation factor can be improved to $O(\log n \log\log n)$ using Bartal's result in [6] and derandomized using the results of [7, 8].

For *single sink* buy-at-bulk (*i.e.*, all demands need to be

*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.

‡Supported by ARO Grants DAAG55-98-1-0170 and ONR Grant N00014-98-1-0589. Department of Computer Science, Stanford University CA 94305. Email: plotkin@cs.stanford.edu.

routed to a single "sink" node), Salman et al [14] 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 [4, 12]. For arbitrary pipe types, Guha et al [10] obtained a $O(1)$ combinatorial approximation. Meyerson et al [13] provide an $O(\log n)$ approximation to single sink network design without the assumption that all pipe types are available on all edges.

In this paper, we consider a restricted version of the single sink buy-at-bulk problem, called the *Access Network Design* problem, defined by Andrews and Zhang [2]. They provided a $O(K)$ approximation using LP rounding, where K is the number of pipe types (which typically is small number). Guha et al [9] give a 95 approximation to this problem using facility location techniques, with a running time of $O(Kn^2 \log n)$. The approximation ratio was improved to 81 in [10], while keeping the running time and basic technique the same.

The only previously known result for the *online* buy-at-bulk problem is $O(\log n \log\log n)$ competitive, and works as follows. We embed the underlying metric into a tree using the results in [5, 6, 7, 8]. Whenever a new source-sink pair arrives, we route it along the tree, and map the path onto the graph. We note that this method requires a computationally expensive tree embedding to be performed. The online single-sink buy-at-bulk problem with just one pipe type has a lower bound on the competitive ratio of $\Omega(\log n)$, which follows easily from the lower bound for online Steiner trees [1, 11]. Therefore the online algorithm via tree embeddings is almost optimal in terms of competitive ratio. However, the lower bound does not apply to Access Network Design, and we will actually show an algorithm that does better for this case.

Our Results: We present a much simpler randomized algorithm for the Access Network Design problem using an entirely different approach. The algorithm simply assigns each arriving demand point a pipe type according to a certain probability distribution. We then use this assigned pipe type to send demand from this node to the closest existing point with a higher assigned pipe type.

For the offline case, our algorithm runs in time $O(n^2)$ and achieves an expected approximation ratio of 68. The most interesting feature of our algorithm is that it is competitive when the demand points arrive online. The competitive ratio we can show is different depending on whether the order of arrival is *random* or *adverserial*. For random order of arrivals, we show an expected competitive ratio of $O(1)$, with $O(n)$ processing time per demand point. Against adverserial arrival of the demand points, we show an expected competitive ratio of $O(K)$. These results improve the best known competitive ratio of $O(\log n \log\log n)$ mentioned above for small values of K.

It is interesting to note that our algorithm is the same for both the offline and online cases – it is just the analysis which gets weaker as we move from the offline case to the online cases. The key technique in the analysis is to construct a routing scheme of low cost on the optimal tree using our assignment of pipe types. We then bound the cost of our routing in terms of the cost of the new routing on the optimal tree. The precise routing scheme on the optimal tree depends on the version of the problem – we can construct better schemes for the offline version compared to the online version.

2 Preliminaries

We are asked to construct a network on an underlying graph of distances. We are given a set S of demand nodes and a single sink s. Each demand node $v \in S$ 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 are given K types of pipes each with a fixed cost and a capacity. The cost of placing a pipe of fixed cost σ_k and capacity u_k along a path of length L will be $\sigma_k L$. The cost of routing demand d along this distance using pipes of type k will therefore be $L \cdot \sigma_k \cdot \lceil \frac{d}{u_k} \rceil$.

2.1 The Incremental Cost Model

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 $\delta_k = \frac{\sigma_k}{u_k}$. 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 $L(\sigma_k + \delta_k d)$. The contribution $L\sigma_k$ is termed the *Fixed Cost*, and the contribution $L\delta_k d$ is termed the *Incremental Cost*. It is 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, the concavity of the cost function gives us the following conditions: $\sigma_1 < \sigma_2 < \cdots < \sigma_K$, and $\delta_1 > \delta_2 > \cdots > \delta_K$.

2.2 The Access Network Design Problem

The Access Network Design problem is a special case 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. While the amount of demand is less than[1] u_k, it is cheaper to use a pipe of type $k-1$ compared to a pipe of type k. For $2 \leq k \leq K$, if $d < u_k$, then $d\delta_{k-1} + \sigma_{k-1} < d\delta_k + \sigma_k$.

2. The smallest demand is at least the smallest pipe capacity.

3. The fixed costs scale, that is, $\sum_{\kappa<k} \sigma_\kappa \leq \sigma_k$.

Note that the above conditions imply that $\frac{\sigma_k - \sigma_{k-1}}{\delta_{k-1} - \delta_k} \geq u_k$ amount of demand is required before it becomes cheaper to use pipe type k than type $k-1$.

We will analyse the case when all demands are unit, and $u_1 = 1$ This assumption is for simplifying the proof and can be done away with easily, without affecting either the running time or the approximation ratio. Details are omitted from this extended abstract.

2.3 Layered Structure of Optimum Solution

Andrews and Zhang [2] showed 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 implies that along any edge, the fixed cost of routing is at most the incremental cost.

We can therefore compare ourselves against the optimal solution that satisfies the above mentioned properties. The Access Network restrictions immediately show this important property about the incremental costs:

Lemma 2.1. *For any $k > 1$, $\delta_{k-1} \leq 2\delta_k$.*

Proof. $\frac{\sigma_k - \sigma_{k-1}}{\delta_{k-1} - \delta_k} \geq u_k = \frac{\sigma_k}{\delta_k}$. □

3 The Offline Algorithm

3.1 Description

The algorithm is very simple to describe – for every demand point, we assign a pipe type at random, and use that pipe type for routing demand from that node to the closest node with larger pipe type. As we mentioned in the previous section, we assume all demands are unit, and $u_1 = 1$.

[1] 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 affect only the constants in our proof, we will simplify our notation by assuming $\beta = 1$.

1. We first eliminate some pipe types so that the incremental costs scale by at exactly $\frac{1}{3}$. By the Lemma 2.1, this can be done at a factor 3 increase in incremental cost. The details are similar to the scaling in [9] and are therefore omitted.

2. For every node v, we choose a pipe of type i with probability $p(i) = \min\{1, \frac{\gamma}{u_i}\}$. We will compute the exact value of γ later. Note that we may be choosing multiple pipes at a location. We will retain the largest pipe type, and discard the rest.

3. We consider nodes in increasing order of pipe types we assigned. For every vertex v with pipe type i, we send its demand (and the demand routed to it) to the closest node w with a pipe of larger type. We will use pipes of type i to route from v to w. Note that the pipes along any path to the sink will be of increasing types.

3.2 Analysis

The basic technique in the analysis is to take the optimal tree and assign the same pipe types to the nodes as we assigned in our solution. We then construct a routing on this tree using these pipe type so that every demand point is routed to a point with larger type. We then argue that the expected cost of this scheme is not much worse than the cost of the optimal solution. Finally, we show that our algorithm cannot be more than constant factor away from the cost of our routing scheme on the optimal solution.

Let the fixed cost of the optimal solution be F^*, the incremental cost be I^*, and the total cost be C^*. Note that $F^* \leq I^*$. After scaling the pipe types in Step (1), our incremental cost increases to $3I^*$.

A subtree of type i is a subtree of the optimum solution such that no pipes of type $i+1$ appear within the subtree, but the pipe exiting the root of the subtree has type $i+1$. The total demand in any such subtree of type i is at least u_{i+1}. Let T_j^i represent the j^{th} subtree of type i.

3.2.1 Routing Scheme on Optimal Solution

We now show a routing scheme on the optimal tree which, given our pipe type assignments, always routes from a node to a node with larger pipe type, and argue that this scheme has low cost. Suppose we are routing demand from a node $v \in T_j^i$ which was assigned a pipe of type $x < i$. We do the following:

1. $T \leftarrow T_l^x$, where $v \in T_l^x$.
2. $r \leftarrow x + 1$.
3. **If** $\exists w \in T$ of type r, route v to w using pipe of type x. **Exit**.
4. **Else**

(a) If $r = K$, route v to the sink using a pipe of type x. **Exit**.
(b) Let $T \subset T_m^r$. Set $T \leftarrow T_m^r$.
(c) $r \leftarrow r + 1$.
5. **Goto** Step (3).

We first estimate the probability that there is at least one pipe of type $i+1$ in T_j^i. Let us call the lower bound on this probability α.

Lemma 3.1. $\alpha \geq 1 - e^{-\gamma}$. *Alternatively,* $\gamma \geq -\ln(1-\alpha)$.

Proof. Let D be the amount of demand in the subtree, where $D \geq u_{i+1}$.

$$\begin{aligned} \alpha &\geq 1 - \left(1 - \frac{\gamma}{u_{i+1}}\right)^D \\ &\geq 1 - e^{-\gamma} \end{aligned}$$

□

3.2.2 Bounding the Incremental Cost

We will now inductively bound the expected incremental cost we pay in the above routing scheme. Consider subtree $T_k^{i-1} \subset T_j^i$. Let r_k^{i-1} denote the root of T_k^{i-1} and r_j^i denote the root of T_j^i. Let e_k^{i-1} denote the edge connecting r_k^{i-1} to r_j^i. Note that the optimal solution was using an edge of type i along e_k^{i-1}, and therefore was paying an incremental cost per unit demand per unit length of δ_i.

In our routing scheme on the optimal tree, denote the incremental cost per unit demand per unit length to route along e_k^{i-1} as U_k^{i-1}. Inductively assume $\mathbf{E}[U_*^{i-1}] \leq c_1\delta_i$, for all trees T_*^{i-1}.

Lemma 3.2. $\mathbf{E}[U_*^i] \leq \frac{\alpha}{3\alpha-2}\delta_{i+1}$.

Proof. Consider the tree T_*^i and the edge e_*^i. We are interested in the expected incremental cost paid along this edge. With probability $\alpha' \geq \alpha$, there exists at least one node of type $i+1$ in T_*^i, in which case, the incremental cost per unit demand per unit length along e_*^i is δ_{i+1}.

With probability $1-\alpha'$, we will not find any such node, in which case the expected incremental cost is at most $c_1\delta_i$ by induction. Also, the expected value depends only on pipes of types $1, 2, \ldots, i$ present in the subtree, and is independent of the presence of a pipe of type $i+1$. Therefore,

$$\begin{aligned} \mathbf{E}[U_*^i] &\leq (1-\alpha')\frac{\alpha}{3\alpha-2}\delta_i + \alpha'\delta_{i+1} \\ &\leq \frac{\alpha}{3-2\alpha}\delta_{i+1} \end{aligned}$$

The final inequality uses the fact that $\delta_i = 3\delta_{i+1}$ because of scaling. □

We now compute the total incremental cost V_j^i of going from r_*^{i-1} to r_j^i. This process involves traveling from r_*^{i-1} to r_j^i along edge e_*^{i-1}, and from there, possibly going to a node of type $i+1$ in T_j^i, and back up to r_j^i along a pipe of type $i+1$. Let l_i denote the average length of the subtrees T_*^{i-1} (including the edge e_*^{i-1}), and $d(T_j^i)$ denote the average length of the tree T_j^i (including the edge e_j^i).

Lemma 3.3. $\sum_i \sum_j \mathbf{E}[V_j^i] \leq \frac{9\alpha^2}{3\alpha-2} I^*$.

Proof. With probability $\alpha' \geq \alpha$, we route from r_*^{i-1} to a randomly chosen node of type $i+1$ in T_j^i using pipes of expected incremental cost $c_1\delta_i$, and from there to r_j^i using pipes of type $i+1$. With the remaining probability, we travel along e_j^i using pipes of expected incremental cost $c_1\delta_i$. Let D_j^i denote the demand in T_j^i.

First, note that $\sum_i \sum_j d(T_j^i)\delta_{i+1} \leq \frac{3}{2} \cdot 3I^*$. This follows from the analysis in [9]. The additional factor of 3 is because of scaling the pipe types in Step (1). Therefore, using $\delta_{i-1} = 3\delta_i$, we obtain:

$$\begin{aligned} \mathbf{E}[\frac{V_j^i}{D_j^i}] &\leq \alpha' \cdot c_1\delta_i \cdot l_i + \alpha' \cdot \delta_{i+1} \cdot d(T_i) \\ &\quad + (1-\alpha') \cdot c_1\delta_i \cdot (d(T_i) - l_i) \\ &\leq \alpha' \cdot d(T_j^i)(c_1\delta_i + \delta_{i+1}) \\ &\quad - (2\alpha' - 1) \cdot c_1\delta_i \cdot (d(T_j^i) - l_i) \end{aligned}$$

We now sum this expression up, and note that the maximum value is attained when $\alpha' = \alpha$, and $c_1 = \frac{\alpha}{3\alpha-2}$.

$$\begin{aligned} \sum_i \sum_j \mathbf{E}[V_j^i] &\leq 3I^* \cdot (1.5\alpha + 3c_1(1.5\alpha + 1 - 2\alpha)) \\ &\leq \frac{9\alpha^2}{3\alpha-2} I^* \end{aligned}$$

□

3.2.3 Bounding the Fixed Cost

We now compute the expected fixed cost per unit demand of the routing scheme. We denote the fixed cost per unit demand to route along e_j^i by the variable W_j^i. Inductively assume that $\mathbf{E}[W_*^{i-1}] \leq c_2\delta_i$.

Lemma 3.4. $\mathbf{E}[W_*^i] \leq \frac{\alpha(1+\gamma)}{3\alpha-2}\delta_{i+1}$.

Proof. Assume that the amount of demand in T_j^i is $D \geq u_{i+1}$. First, we observe that if there is at least one pipe of type $i+1$ within the subtree, the expected fixed cost per unit demand will be proportional to the expected number of

points of type $i+1$, and this cost is $(1+\frac{\gamma(D-1)}{u_{i+1}}) \cdot \frac{\sigma_{i+1}}{D} \leq (1+\gamma)\delta_{i+1}$.

If the probability of a pipe of type $i+1$ is exactly $\alpha' \geq \alpha$, using the same technique as for the incremental cost, we have:

$$\begin{aligned}\mathbf{E}[W_*^i] &\leq (1-\alpha')\frac{\alpha(1+\gamma)}{3\alpha-2}\delta_i + \alpha'(1+\gamma)\delta_{i+1} \\ &\leq \frac{\alpha(1+\gamma)}{3-2\alpha}\delta_{i+1}\end{aligned}$$

□

We now compute the total fixed cost Z_j^i to route from r_*^{i-1} to r_j^i. This will involve going up to r_j^i along edge e_*^{i-1}, and possible map back to a node of type $i+1$ within T_j^i and from there up to r_j^i using a pipe of type $i+1$.

Lemma 3.5. $\sum_i \sum_j \mathbf{E}[Z_j^i] \leq \left(\frac{9\alpha^2(1+\gamma)}{3\alpha-2}\right) I^*$.

Proof. We use exactly the same arguments as for the incremental cost. Let α' denote the probability of a pipe of $i+1$ existing in the subtree.

$$\begin{aligned}\mathbf{E}[\frac{Z_j^i}{D_j^i}] &\leq \alpha' \cdot c_2\delta_i \cdot l_i + \alpha' \cdot \delta_{i+1}(1+\gamma) \cdot d(T_i) \\ &\quad +(1-\alpha') \cdot c_2\delta_i \cdot (d(T_i) - l_i) \\ &\leq \alpha' \cdot d(T_j^i)(c_2\delta_i + (1+\gamma)\delta_{i+1}) \\ &\quad -(2\alpha'-1) \cdot c_2\delta_i \cdot (d(T_j^i) - l_i)\end{aligned}$$

We now sum this expression up, and note that the maximum value is attained when $\alpha' = \alpha$, and $c_2 = \frac{\alpha(1+\gamma)}{3\alpha-2}$.

$$\begin{aligned}\sum_i \sum_j \mathbf{E}[Z_j^i] &\leq 3I^*(1.5(1+\gamma) + 3c_2(1.5\alpha + 1 - 2\alpha)) \\ &\leq \frac{9\alpha^2(1+\gamma)}{3\alpha-2} I^*\end{aligned}$$

□

3.2.4 Final Analysis

Denote the expected fixed and incremental costs of the routing scheme on the optimal tree described above by F_O and I_O respectively; and the same for our solution by F_S and I_S. The following lemma is based on the observation that our algorithm routes every point to a closer point than the routing scheme outlined above.

Lemma 3.6. $\mathbf{E}[F_S|F_O] \leq F_O$ *and* $\mathbf{E}[I_S|I_O] \leq 3I_O$.

Proof. Since we always route to the closest pipe of higher type, our fixed cost cannot be more than the fixed cost of the routing scheme on the optimal tree described above. Therefore, the first claim follows. To prove the second claim, we use induction starting from the largest pipe type. Clearly, the claim holds for the largest pipe type. Consider a node v of type i. Suppose we were sending it to a node w distance d_1 away, while the routing scheme described above on the optimal tree was sending it a distance $d_2 \geq d_1$ away to a node z. Let the optimal routing cost for node w be $I_O(w)$. Clearly, $I_O(z) \leq \delta_{i+1}(d_1+d_2) + I_O(w)$.

Denote the routing cost for z in our solution by $I_S(z)$. By induction, $I_S(z) \leq 3I_O(z)$. Combining this with $d_1 \geq d_2$ and $\delta_{i+1} = \frac{\delta_i}{3}$, we derive:

$$\begin{aligned}I_S(v) &= d_2 \cdot \delta_i + I_S(z) \\ &\leq d_1 \cdot \delta_i + I_S(z) \\ &\leq d_1 \cdot \delta_i + 3I_O(z) \\ &\leq d_1 \cdot \delta_i + 3(2\delta_{i+1}d_1 + I_O(w)) \\ &\leq 3(d_1 \cdot \delta_i + I_O(w)) \\ &= 3I_O(v)\end{aligned}$$

□

Therefore, we have:

1. $\mathbf{E}[F_S] \leq \left(\frac{9\alpha^2(1+\gamma)}{3\alpha-2}\right) I^*$.
2. $\mathbf{E}[I_S] \leq \left(\frac{27\alpha^2}{(3\alpha-2)}\right) I^*$.

When $\gamma = 2$, then $\alpha = 1 - e^{-\gamma} = 0.865$. Plugging this value in, we have a 67.95 approximation. We can improve this constant slightly by a more careful scaling and choice of γ. We omit the computation from this abstract.

4 Online Algorithm for Non-Adverserial Inputs

We now assume that the points arrive online, but in random order (meaning that each permutation of the points is equally likely). We wish to lay pipes in such a fashion that we never remove a pipe we already laid, and all existing points have a path to the sink. The goal is to construct a solution which is $O(1)$-competitive to the optimal solution at every step.

The algorithm is almost the same as before. When a new demand point is added, it is assigned a random pipe type. Using this pipe type, it is then connected to the closest *existing* point of greater pipe type. It is easy to see that the resulting structure is a tree. Note however that when we add a new demand point, we may have to buy extra capacity along every link on its path to the sink. This gives a running time of $O(n)$ per addition.

4.1 Routing Scheme on Optimal Solution

Consider any stage of the algorithm where we have added a set $\mathcal{P}$ of points. First, we can convert the optimal solution at any stage into a tree that satisfies the Access Network conditions at a constant factor loss in cost. Consider the optimal solution T^* on this set.

As before, we will construct a routing scheme on the optimal tree given the order of arrival of the points and the random assignments of pipe types, and argue that this routing has low cost. Each demand point will be routed to an existing point with larger type in this scheme. Whenever a new point v arives, we will assign a random pipe type to it. Suppose we assigned it a pipe of type x. The routing scheme, which is slightly different from the offline case, is as follows:

1. $T \leftarrow T_l^x$, where $v \in T_l^x$.
2. $r \leftarrow x + 1$.
3. **If** $\exists w \in T$ with pipe type $x < y \leq r$ which previously arrived:
   (a) Find such a w with the largest pipe type.
   (b) Route v to w using pipe of type x. **Exit**.
4. **Else**
   (a) If $r = K$, route v to the sink using a pipe of type x. **Exit**.
   (b) Let $T \subset T_m^r$. Set $T \leftarrow T_m^r$.
   (c) $r \leftarrow r + 1$.
5. **Goto** Step (3).

Consider any stage of the algorithm where we have added a set $\mathcal{P}$ of points. Consider the optimal solution T^* on this set. As before, let (F^*, I^*) be the fixed and incremental cost of this optimal solution. Note that $F^* \leq I^*$. Let $C^* = F^* + I^*$. Note that the points in set $\mathcal{P}$ have arrived in random order.

Assume as before that we can scale the pipe types so that the incremental costs scale by exactly $\beta = \frac{1}{3}$. The scaling of the incremental costs causes the incremental cost to become $3I^*$. We also note that this procedure ensures $u_{i+1} \geq 3u_i$.

We first change the optimal tree so that every tree T_j^{i-1} has between $\frac{3u_i}{4}$ and $\frac{3u_i}{2}$ demand. Furthermore, if there are at least two subtrees within a tree T_j^i, no subtree has more than half the total demand in the parent tree. This increases the fixed cost, but leaves the incremental cost unchanged. We just split the tree to ensure this condition. The split can always be performed, as the demand in tree T_j^i is at least u_{i+1}, which in turn, is at least $3u_i$.

We use the routing scheme on the optimal solution outlined in Section 4

4.2 Bounding the Fixed Cost

As before, consider an edge e_m^{i-1} of the optimal solution, which connects r_m^{i-1} to r_j^i. Let the total demand traversing this edge be D_m^{i-1}, and let the length of this edge be l_m^{i-1}. The fixed cost has three components:

1. W_m^{i-1} of pipes of type $j \leq i$ traveling e_m^{i-1} from r_m^{i-1} to r_j^i.
2. F_j^i of pipes of type $i+1$ placed in T_j^i in traveling to r_j^i.
3. Z_m^{i-1} of pipes of type $j \leq i$ in traveling from r_j^i to a point of type larger than j in T_j^i.

Lemma 4.1. $\mathbf{E}[W_m^{i-1}] \leq (6+\gamma) l_m^{i-1} \sigma_i \leq \frac{4}{3}(6+\gamma)\delta_i D_m^{i-1}$.

Proof. Let n_k^i denote the number of demand points of type k traversing e_m^{i-1}. Note that we will use a pipe of type k on this edge iff no existing point within T_m^{i-1} was assigned a pipe of type at least $k+1$. Therefore, $\mathbf{E}[n_k^i] \leq \frac{u_{k+1}}{u_k}$. The expected fixed cost paid by these points is therefore at most $\delta_k u_{k+1} = 3\sigma_{k+1}$. Therefore, the total expected cost is at most $6\sigma_i$ by the assumption that $\sum_{k<i} \sigma_k \leq \sigma_i$. The expected cost paid by points of type i is at most $\frac{\gamma}{u_i} u_i \sigma_i = \gamma \sigma_i$. The second part of the lemma follows from the observation that $D_j^{i-1} \geq \frac{3u_i}{4}$. □

Lemma 4.2. $\mathbf{E}[F_m^{i-1}] \leq \gamma \delta_i D_m^{i-1} d(T_m^{i-1})$.

Lemma 4.3. $\mathbf{E}[Z_m^{i-1}] \leq \frac{8}{3} d(T_j^i) \cdot (6+\gamma) \cdot \delta_i D_m^{i-1}$.

Proof. Consider any demand point of type $k \leq i$ that arrives at r_j^i from tree T_m^{i-1}. In the worst case, it could be sent to a point chosen uniformly at random from a different subtree from T_m^{i-1}. Since no subtree can have more than $\frac{1}{2}$ of the total demand, the expected distance travelled by this demand is at most $2d(T_j^i)$. The expected fixed cost paid per unit demand is at most $\frac{4}{3}(6+\gamma)$, and this value is independent of what points are chosen in the other subtrees. □

We now compute the total fixed cost paid by the demands. Let the total fixed cost be denoted F_O.

Lemma 4.4. $\mathbf{E}[F_O] \leq (96 + 20.5\gamma) I^*$.

Proof.

$$F_O = \sum_{i,j} \text{`} F_j^i + \sum_{i,j} W_j^i + \sum_{i,j} Z_j^i$$

We note that $\sum_i \sum_{k \geq i} \sum_j d(T_j^i)\delta_k \leq \frac{3}{2} 3 I^*$. Here, the factor of 3 is from the scaling of the incremental costs. Therefore, the total fixed cost is at most $(6+\gamma) \cdot (\frac{4}{3} * 3 + \frac{3}{2} * 3 * \frac{8}{3}) \cdot I^* + \gamma \cdot \frac{3}{2} \cdot 3I^*$. □

In our online solution, the total expected fixed cost of the edges we lay is at most F_O, as we route to the closest existing point with larger pipe type.

4.3 Bounding the Incremental Cost

Consider the edge e_m^{i-1}. Let us find the expected demand originating in T_m^{i-1} and traversing this edge using a pipe of type $k \leq i$. Let $e_m^{i-1} \in T_j^i$. Let the incremental cost due to all such demand be X_m^{i-1}. Let the demand along this edge be D_m^{i-1}, and let the length of this edge be l_m^{i-1}.

Lemma 4.5. $\mathbf{E}[X_m^{i-1}] \leq (1 + \frac{6}{\gamma}) \cdot \delta_i D_m^{i-1}$.

Proof. Let n_k^i denote the amount of demand using a pipe of type $k < i$. $\mathbf{E}[n_k^i] \leq \frac{u_{k+1}}{\gamma}$, and the expected incremental cost per unit length paid by this demand is at most $\delta_k \cdot \mathbf{E}[n_k^i]$, which is at most $3\frac{\sigma_{k+1}}{\gamma}$. This summed over $1 \leq k < i$ gives an incremental cost of at most $\frac{6}{\gamma}\sigma_i \leq \frac{6}{\gamma}\delta_i D_j^{i-1}$. Even if we assume all demand uses pipes of type i, this contributes at most $\delta_i D_j^{i-1}$ to the expected incremental cost. □

We now add the worst possible incremental cost that this demand traveling upwards along e_m^{i-1} could encounter. This demand could be sent from r_j^i to a node of larger type in $T_j^i - T_m^{i-1}$ and from there back up to r_j^i along this larger pipe type. Let Y_m^{i-1} denote the incremental cost paid by this demand at r_j^i when it is mapped back and again up along a pipe of larger type. Let D_m^{i-1} denote the total demand in this subtree, and $d(T_j^i)$ denote the average distance in this subtree.

Lemma 4.6. $\mathbf{E}[Y_m^{i-1}] \leq \frac{16}{3} d(T_j^i) \cdot (\frac{6}{\gamma} + 1) \cdot \delta_i$.

Proof. The proof is exactly the same as for the fixed cost, and is therefore omitted. Note that since the demand is mapped both downward and back up along a larger type, we double the distance traveled. □

We now compute the total fixed cost I_O paid by the demands.

Lemma 4.7. $\mathbf{E}[I_O] \leq 28(\frac{6}{\gamma} + 1)I^*$.

Proof.

$$I_O = \sum_{i,j} X_j^i + \sum_{i,j} Y_j^i$$

Now, $\sum_i \sum_{k \geq i} \sum_j d(T_j^i)\delta_k \leq \frac{3}{2} 3 I^*$. Here, the factor of 3 is from the scaling of the incremental costs. Therefore, the total fixed cost is at most $(\frac{6}{\gamma}+1)\cdot(\frac{4}{3}*3+\frac{3}{2}*3*\frac{16}{3})\cdot I^* = 28(\frac{6}{\gamma} + 1)I^*$. □

4.4 Final Analysis

Let the fixed cost of our algorithm be F_S, and the incremental cost be I_S. From Lemma 3.6, since we connect to the closest point of the larger type, $\mathbf{E}[F_S|F_O] \leq F_O$, and $\mathbf{E}[I_S|I_O] \leq 3I_O$. Therefore, we have:

1. $\mathbf{E}[F_S] \leq (96 + 20.5\gamma)I^*$.
2. $\mathbf{E}[I_S] \leq 84(\frac{6}{\gamma} + 1)I^*$.

For $\gamma = 4$, we have an expected competitive ratio of 408. This gives us the following theorem:

Theorem 4.1. *There is a constant competitive algorithm for Access Network Design with the arrival pattern being a random permutation of the demand points.*

5 Online Algorithm for Adverserial Inputs

The algorithm is slightly different from the online non-adverserial case. We will construct an acyclic graph instead of a tree. However, we will not re-route any demand, and always send a node to one or more nodes of larger type. Let Type(v) denote the assigned pipe type for vertex v. Let $P(v)$ denote the current parent of v in the graph. Initially, $P(v) = s$ for all v.

1. For a new point v, assign random pipe type as before.
2. $w \leftarrow$ Closest existing point to v with Type(w) > Type(v).
3. **If** dist$(v, w) \leq \frac{2}{3}$dist$(v, P(v))$, set $P(v) \leftarrow w$.
4. Route new demand from v to $P(v)$ with pipes Type(v).
5. $v \leftarrow P(v)$.
6. **If** $v \neq s$, **goto** Step (2).

We will use a different analysis from the non-adverserial case. In fact, we will use different routing schemes to account for the fixed and incremental costs.

As before, the optimal solution on the set $\mathcal{P}$ of arrived points can be converted into a tree satisfying the Access Network constraints with a constant factor loss in cost. Let T^* denote the optimal tree on $\mathcal{P}$. We scale the pipe types so that the incremental costs scale by factors of exactly 3. We also change the optimal tree so that every tree T_j^{i-1} has between $\frac{3u_i}{4}$ and $\frac{3u_i}{2}$ demand. Furthermore, if there are at least two subtrees to a tree, no subtree has more than half the total demand of the parent tree.

5.1 Routing Scheme for Bounding the Fixed Cost

The following is routing scheme in the optimal solution so that any demand point is sent to a previously arrived point of larger pipe type. Suppose the arriving demand point v is of type x.

1. $T \leftarrow T_m^x$, where $v \in T_m^x$; $r \leftarrow x + 1$.
2. **If** $\exists w \in T$ with pipe type y s.t. $r \geq y > x$ which previously arrived:
   (a) Let $T_1^{r-2}, T_2^{r-2}, \ldots, T_k^{r-2} \subset T$ be the subtrees with points of type in the range $[x + 1, r]$.
   (b) Let the most recent point of largest type in these

subtrees be $w_1, w_2, \ldots, w_k$ respectively.
   (c) Let w_l be the most recently arriving among these points.
   (d) Route v to w_l using pipe of type x. **Exit**.
3. **Else**
   (a) If $r = K$, route v to the sink using a pipe of type x. **Exit**.
   (b) Let $T \subset T_m^r$. Set $T \leftarrow T_m^r$.
   (c) $r \leftarrow r + 1$.
4. **Goto** Step (2).

As before, let the average length of subtree T_j^i be denoted $d(T_j^i)$.

Lemma 5.1. *Consider any subtree T_m^{i-1} in which at least one point of type i has appeared. The expected distance of any such point of type i to r_m^{i-1} is at most $\frac{3\gamma}{2} d(T_m^{i-1})$.*

Proof. The expected number of points which must appear in the subtree before we place a pipe of type i is at least $\frac{2}{3\gamma}$ fraction of the total number of points in the subtree, independent of the order of arrival of the points. The lemma then follows. □

We first bound the fixed cost F_m^{i-1} paid by the points of type i within T_m^{i-1} to reach r_m^{i-1}. Since we prove a $\Omega(1)$ approximation, we omit the exact constants from the proof. The proof of the following lemma is the same as in the previous section.

Lemma 5.2. $\mathbf{E}[F_m^{i-1}] \leq O(1) \cdot \delta_i d(T_m^{i-1})$.

Consider the edge e_m^{i-1}. We will first compute the expected total fixed cost per unit length, W_m^{i-1} paid by demand points of type $j \leq i$ traversing this edge in the upward direction. Let D_m^{i-1} denote the amount of demand in T_m^{i-1}. This computation is the same as for the non-adverserial case in Lemma 4.1, and therefore, we have:

Lemma 5.3. $\mathbf{E}[W_m^{i-1}] \leq O(1) \cdot \delta_i D_m^{i-1}$.

We now have to bound the expected fixed cost paid by the demands traveling in the reverse direction. We cannot use the random mapping technique used in the previous section, as it was crucially based on the point of type larger than j in T_m^{i-1} being a random point. This is no longer true. We use a different technique now.

Consider any point p where we placed a pipe of type i. Let $p \in T_l^i$. Suppose we routed this point up $r_m^{i'}$ for $i' > i$, and from $r_m^{i'}$ to a point of type $q > i$ within that subtree. Let $F_o(e,p)$ denote the fixed cost paid by p along this edge.

Lemma 5.4. $\sum_e \sum_p F_o(e,p) = \sum_i \sum_m (W_m^i + F_m^i) \leq O(1) I^*$.

Consider any edge e on the path between p and $r_m^{i'}$. We will compute the expected number of points $n_j(e,p)$ of type $j < i$ that will be routed to p along e.

Lemma 5.5. $\mathbf{E}[n_j(e,p)] \leq O(K) \cdot \frac{u_{j+1}}{u_j}$.

Proof. Let us identify the demand points of type j which will be sent to p. Let r_f^t be a node between e and $r_m^{i'}$. Consider the subtrees $T_1^{t-1}, T_2^{t-1}, \ldots, T_k^{t-1}$ which are rooted at r_f^t, and the time instant just after we added p. Let $p \in T_1^{t-1}$.

Adding points to subtrees in which the largest type is at least $j+1$ do not matter, as we cannot route a pipe of type j out of these. The adversary therefore has the choice of picking one of the other subtrees, and adding a point there. The point gets assigned a pipe of type j with probability $\frac{\gamma}{u_j}$. As soon as any of these subtrees gets a pipe of at least $j+1$, we will start routing points of type j to this subtree instead of p.

The probability of a point of type $j+1$ appearing relative to a point of type j is $\frac{u_j}{u_{j+1}}$. Therefore, the expected number of points of type j appearing before a point of type $j+1$ appears is therefore $\frac{u_{j+1}}{u_j}$. All these points get routed to p at this level. Summing this up over all nodes r_f^t between e and $r_m^{i'}$, we have the lemma. □

We now compute the expected fixed cost along edge e due to the demand points of type $j < i$ which are routed to p. Let $Z(e,p)$ denote the fixed cost paid by demands routed to this node.

Lemma 5.6. $\mathbf{E}[Z(e,p) | F_o(e,p)] \leq O(K) \cdot F_o(e,p)$.

Proof. The proof follows from the previous lemma in exactly the same fashion as bounding Z_m^{i-1} in Lemma 4.3. □

We now bound the total fixed cost F_O we pay in our routing scheme. The fixed cost has two components – that due to routing upwards and that due to routing back. $F_o(e,p)$ bounds the cost of routing forward, while $Z(e,p)$ bounds the cost of routing back. Adding these up, we have:

Lemma 5.7. $\mathbf{E}[F_O] \leq O(K) \cdot I^*$.

Let F_S be the fixed cost our algorithm pays. Note that the total fixed cost paid by edges out of v is at most three times the fixed cost of the first edge, which can be no longer than the edge used to route out of v in the routing scheme on the optimal solution. Therefore, $E[F_S | F_O] \leq 3F_O$.

Theorem 5.1. *Let F_O be the fixed cost paid in the routing scheme described above. Let F_S denote the fixed cost paid by our algorithm, and I^* denote the optimal incremental cost. Then,* $\mathbf{E}[F_O] \leq O(K) \cdot I^*$ *and* $\mathbf{E}[F_S | F_O] \leq 3F_O$.

5.2 Routing Scheme for Bounding the Incremental Cost

We will use another routing scheme which does not maintain the tree property. Different demands arriving at

a node could get sent along different paths. The scheme is as follows:

1. $v \leftarrow$ Newly arriving point.
2. $x \leftarrow$ Type(v).
3. $T \leftarrow T_m^x$, where $v \in T_m^x$.
4. $r \leftarrow x + 1$.
5. **If** $\exists w \in T$ with pipe type $r \geq y > x$, which previously arrived:
   (a) Let $T_1^{r-2}, T_2^{r-2}, \ldots, T_k^{r-2} \subset T$ be the subtrees with points of type in the range $[x+1, r]$.
   (b) Let the most recent point of largest type in these subtrees be $w_1, w_2, \ldots, w_k$ respectively.
   (c) Let w_l be the most recently arriving among these points.
   (d) Route the new demand from v to w_l using pipe-type x.
   (e) $v \leftarrow w_l$.
   (f) **Goto** Step (2).
6. **Else**
   (a) **If** $r = K$, route v to the sink using pipe-type x. **Exit**.
   (b) Let $T \subset T_m^r$. Set $T \leftarrow T_m^r$.
   (c) $r \leftarrow r + 1$.
7. **Goto** Step (3).

We will now bound the incremental cost of the routing process in terms of the fixed cost of the routing scheme in Section 5.1. Consider any point p where we placed a pipe of type i. Suppose this is present in tree T_l^i and it is routed upward to $r_m^{i'}$ from where it is sent to a node with larger pipe type. This route is exactly the same as the route for the fixed cost. Later on, demands arriving at this node could get sent along different paths.

First, note that the incremental cost of routing demand upward using type i is at most the worst possible cost of routing the demands incident on p downward along their respective pipe types, as $r_m^{i'}$ is the highest possible node in the tree to which demand can ever be sent. Therefore, we can ignore the cost of routing upwards.

Let e be any edge on the path from p to $r_m^{i'}$. Let $Y(e,p)$ denote the incremental cost per unit length paid along edge e by demands sent to node p. Assume e has length $l(e)$. Let $F_o(e,p)$ denote the fixed cost per unit length paid by p along this edge in the routing scheme for the fixed cost.

Lemma 5.8. $\mathbf{E}[Y(e,p)|F_o(e,p)] \leq O(K) \cdot F_o(e,p)$.

Proof. Similar to the proof of Lemma 5.5, the expected amount of demand which can arrive at r_f^t from its subtrees before we open a pipe of type $j+1$ in one of the subtrees is at most u_{j+1}. All this demand could use a pipe of type j and get sent to p. As soon as we open a pipe of type $j+1$, this is more recent than p, and therefore, the demand will now get sent to this new node instead of p. Therefore, the expected incremetal cost paid along edge e by demands which is sent to p using pipes of type type $j < i$ is at most $O(K) \cdot u_{j+1}\delta_j = O(K)\sigma_{j+1}$. Summing this up, the lemma follows. □

This implies that the expected incremental cost is at most $O(K)\sum_{e,p} \mathbf{E}[F_o(e,p)]l(e)$, which we showed above is at most $O(K)I^*$. Let I_O denote the incremental cost of this routing scheme.

Theorem 5.2. $\mathbf{E}[I_O] \leq O(K)I^*$

Let I_S denote the incremental cost of our routing scheme.

Lemma 5.9. $\mathbf{E}[I_S|I_O] \leq 9I_O$.

Proof. Consider the instant we add a demand point p. Consider the demand added till then, and the routing scheme on these points in the optimal solution shown above. If v is being routed to w, and dist$(v,w) = d_1$, then our algorithm has assigned v to a point no more than $1.5d_1$ away. This is true even if we consider the best possible incremental cost solution on points uptill p which routes points only to nodes of larger types. Using the same idea as in the proof of Lemma 3.6, we can show that the incremental cost of routing p in our solution is not too far away from the best possible incremental cost that can be assigned by the above routing process. □

Theorem 5.3. *Let I_O be the incremental cost paid in the routing scheme described above. Let I_S denote the incremental cost paid by our algorithm, and I^* denote the optimal incremental cost. Then, $\mathbf{E}[I_O] \leq O(K) \cdot I^*$ and $\mathbf{E}[I_S|I_O] \leq 9I_O$.*

Finally, combining the analysis for fixed and incremental costs, we have:

Theorem 5.4. *The algorithm has an expected competitive ratio of $O(K)$ against adverserial arrival of demand points.*

References

[1] N. Alon and Y. Azar. On-line steiner trees in the euclidean plane. *Discrete and Computational Geometry*, 10(2):113–121, 1993.

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

[3] 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.

[4] 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.

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

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

[7] 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.

[8] 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.

[9] S. Guha, A. Meyerson, and K. Munagala. Hierarchical placement and network design problems. *Proceedings of 41st IEEE FOCS*, 2000.

[10] S. Guha, A. Meyerson, and K. Munagala. A constant factor approximation for the single-sink edge installation problem. *Proceedings of 33rd ACM STOC*, 2001.

[11] M. Imaze and B. Waxman. Dynamic steiner tree problem. *SIAM J. Discrete Math.*, 4(3):369–384, August 1991.

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

[13] A. Meyerson, K. Munagala, and S. Plotkin. COST-DISTANCE: Two metric network design. *Proceedings of 41st IEEE FOCS*, 2000.

[14] 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.