Improved Approximation Algorithms for Buy-at-bulk Network Design

Naveen Garg, Rohit Khandekar
Department of Computer Science and Engineering,
Indian Institute of Technology, New Delhi, India.

Goran Konjevod
Department of Mathematical Sciences,
Carnegie Mellon University, Pittsburgh, PA 15213-3890.

R. Ravi, F. S. Salman, Amitabh Sinha
GSIA, Carnegie Mellon University, Pittsburgh, PA 15213-3890.

July 2000

1 Introduction

Problem. Consider a network through which each unit of traffic can be routed at a given cost per unit length. As new technology emerges, it is often possible to reduce the per unit cost of routing traffic by upgrading the links of the network. Upgrading a link requires a one-time equipment purchasing and installation cost but in return provides a *discounted* routing cost on the link. Then, it is important to find out which links of the network should be upgraded so that the total cost of the upgrading and the routing will be minimum. Most often the resources that facilitate the routing on the network can be chosen from a set of available models. A model with a higher upgrading price provides a deeper discount. This "deep discount" problem becomes more interesting when a dollar spent on upgrading can buy more discount rate if we are willing to spend more on upgrading.

The deep discount problem models the design of telecommunication networks where a set of cables types can be used to install bandwidth on the links. In fact, the deep discount problem is closely related to the *buy-at-bulk* network design problem studied in [AA97, SCR+97]. In the buy-at-bulk problem, each cable type has a cost and a capacity per unit length. The costs of the cables obey the economies of scale, that is cables with higher capacity have a lower cost per unit demand. On every edge of the network, multiple copies of different cable types can be bought to support the flow on the edge. The objective is to minimize the total cost of cables installed to support the traffic. The deep discount and the buy-at-bulk problems differ slightly in their cost structures. It is easy to show that any approximation for one of them is an approximation to the other with a factor 2 loss in the performance ratio.

In this paper we focus on the single-sink version of the deep discount problem, where every demand point sends its traffic to a sink node. This model is useful especially in the design of access networks, where traffic from endusers is collected at a switching center. Also, in the transportation of goods and supply chains, we can model the design of a network through which commodities from the suppliers are collected at a warehouse.

Previous Work. Designing networks using facilities that provide economies of scale has attracted interest in recent years. Awerbuch and Azar [AA97] gave a randomized $O(\log^2 n)$ approximation algorithm for the buy-at-bulk problem with many cable types and many sources and sinks, where n is the number of nodes in the input graph. Mansour and Peleg [MP 94] gave an $O(\log n)$ approximation for the single cable type case. For the case with a single-sink, Salman et.

al [SCR+97] gave an $O(\log D)$ approximation algorithm for Euclidean metric spaces, where D is the total demand. They also gave a constant approximation in [SCR+97] for the single cable type case, which was later improved by Hassin, Ravi and Salman [HRS00].

Andrews and Zhang [AZ98] studied a special case of the single-sink buy-at-bulk problem which they call the *access network design* problem and gave an $O(k^2)$ approximation, where k is the number of cable types. As in the deep discount problem, they use a cost structure where each cable type has a buying and a routing cost, but they assume that if a cable type is used, the routing cost is at least a constant times the buying cost. This assumption enables charging the building cost to the routing cost for a demand path and therefore approximate solutions that send traffic through shortest paths can be constructed. On the other hand, in the deep discount problem we do not have this assumption and need to balance the building and routing costs.

Our Results. We give an $O(k)$ approximation algorithm to the deep discount problem, where k is the number of discount types. Our algorithm is based on LP rounding. The LP formulation that we use is similar to the one used by Andrews and Zhang, but our rounding algorithm differs in the construction of an integral solution. We construct a tree network in stages. Starting with the largest discount type, we use only one discount type in each stage. We first select the nodes to be included in a stage, and then build a LAST (light approximate shortest paths tree) on the selected nodes.

In the next section we define the deep discount problem formally and show its relation to the k-cable buy-at-bulk problem. In section 3, we show that there exists an optimal tree solution with monotonic discount types. In sections 4 through 7, we present the LP rounding algorithm and the analysis of the performance ratio.

2 Preliminaries

2.1 The deep discount problem

Let $G = (V, E)$ be a graph with edge-lengths $l : E \to I\!R^+$. Let $d(u, v)$ denote the length of the shortest path between vertices u and v. We are given source vertices $\{v_1, \dots, v_m\} = \mathcal{S} \subseteq V$ and a sink vertex $t \in V$. We want to simultaneously route $\texttt{dem}_j > 0$ amount of flow from v_j to t. The cost of routing flow on an edge is proportional to the length of the edge and the amount of flow. The proportionality constant is called the rate of routing. There are k different rates $r_0, r_1, \dots, r_{k-1}$ available. We say that we are using discount-i while routing with the rate r_i. For using a rate r_i on an edge e, we also have to pay a fixed cost of $p_i l_e$. This can be thought of as a cost to purchase the discount-i on this edge. Therefore, for routing f units of flow on edge e using discount-i, the total cost is $l_e(p_i + r_i f)$.

Let us order the rates as $r_0 > r_1 > \dots > r_{k-1}$. The rate $r_0 = 1$ and the price $p_0 = 0$ correspond to not using any discount. Observe that if $p_i \geq p_{i+1}$ for some i then discount-i will never be used. Therefore, without loss of generality, we can assume $p_0 < p_1 < \dots < p_{k-1}$.

2.2 The buy-at-bulk problem with k-cable types

In an edge-weighted undirected graph, suppose we are given a set of sources $\{v_1, \dots, v_m\}$ which want to transport $\{\texttt{dem}_1, \dots, \texttt{dem}_m\}$ units of flow respectively to a common sink s. We have available to us k different cables, each having capacity u_i and cost c_i. We wish to buy cables such that we have enough capacity to support the simultaneous flow requirements. There is no flow cost, our only cost incurred is the purchase price of cables. The problem of finding a minimum cost feasible network is the buy-at-bulk problem with k-cable types (or the k-cable problem/KCP for short). It is **NP**-Hard [SCR+97].

2.3 Approximate equivalence of KCP and DD

Suppose we are given a KCP instance $KCP = (G, c, u)$ on a graph G with k cable types having costs and capacities $(1,1), (c_1, u_1), \ldots, (c_{k-1}, u_{k-1})$. We transform it into an instance of DD by setting edge costs (fixed and per-unit) $(0,1), (c_1, \frac{c_1}{u_1}), \ldots, (c_{k-1}, \frac{c_{k-1}}{u_{k-1}})$, and call this $DD(KCP)$.

Conversely, given a DD instance $DD = (G, p, r)$, we transform it into a KCP instance $KCP(DD)$ with cable types having costs and capacities $(1,1), (p_1, \frac{p_1}{r_1}), \ldots, (p_{k-1}, \frac{p_{k-1}}{r_{k-1}})$.

It is easy to see that $KCP(DD(KCP)) = KCP$ and $DD(KCP(DD)) = DD$, that is, the two transformations are inverses of each other. For a problem instance X, we abuse notation to let X also denote the cost of a feasible solution to it. Let X^* denote the cost of an optimal (integer) solution to X.

Lemma 2.1 $KCP \leq DD(KCP)^*$

Proof. Consider an edge e and let the flow on e in $DD(KCP)^*$ be x_e. If the solution uses discount type 0 on e, then the KCP solution does not pay any more than the routing cost already paid. If it uses a discount type $i > 0$, we install $\lceil \frac{x_e}{u_i} \rceil$ copies of cable type i at this edge. Clearly this gives us a feasible solution to KCP. For this edge $DD^*(KCP)$ has routing cost $\frac{l_e x_e c_i}{u_i}$ and building cost $l_e c_i$, hence a total cost of $l_e c_i (1 + \frac{x_e}{u_i})$. The KCP solution has cable cost $l_e c_i \lceil \frac{x_e}{u_i} \rceil$ on edge e, which is no more than the total cost incurred by edge e in $DD(KCP)^*$. ∎

Lemma 2.2 $DD \leq 2KCP(DD)^*$

Proof. We will initially allow for multiple discount types on each edge and pay for it all. A pruned solution will satisfy all the desired properties and cost only less. Hence let x_e^i be the number of copies of cable type i used at edge e. Note that the flow on e is at most $x_e^i p_i / r_i$). We purchase a discount type i cable on edge e and route the flow through it. We pay no more than $p_i l_e + \frac{x_e^i p_i}{r_i} r_i l_e \leq (x_e^i + 1) p_i l_e$, which is no more than two times the cost $x_e^i p_i l_e$ already paid by $KCP(DD)$. ∎

Together, the above two lemmas imply that $KCP(DD)^* \leq KCP(DD) \leq DD^* \leq DD \leq 2KCP(DD)^*$, so that a ρ approximation algorithm for KCP gives a 2ρ approximation algorithm for DD. Similarly, a ρ approximation to DD is a 2ρ approximation to KCP.

3 Structure of the optimum

Let us look at how an optimal solution allocates the discount types to edges and routes the flow to the sink. Clearly an edge will use only one type of discount. Suppose in an optimum, an edge e uses discount-i. Define a new length function $l'_e := r_i l_e$. Clearly once the price for the discount is paid, the routing cost is minimized if we route along a shortest path according to the length function l'. Therefore, there is an optimum which routes along shortest paths according to such a length function l'. As a result, we can also assume that the flow never splits. That is, if two commodities share an edge then they share all the subsequent edges on their paths to t. This is because we can choose a shortest path tree in the support graph; flow along a tree to the root will never split.

The cost of routing f units of flow on an edge e using discount-i is $l_e(p_i+r_if)$. So, the discount type corresponding to minimum cost depends only on f and is given by $type(f) := minarg_i\{p_i + r_if | 0 \leq i < k\}$.

Lemma 3.1 *The function type(f) defined above is non-decreasing in f.*

Proof. Consider any two discount types, say i and j with $i < j$. We know that $r_i > r_j$ and $p_i < p_j$. As the cost functions $p+rf$ are linear, there is a critical value f of flow such that, $p_i+r_if = p_j+r_jf$. If the flow is smaller than f, discount-i is better than discount-j. And if the flow is more than f, discount-j is better than discount-i. This proves the lemma. ∎

Suppose commodity-j flows along a path P from v_j to t. As the flow never splits, the flow along P is non-decreasing as we go from v_j to t. Hence from the above lemma, the discount type never decreases as we go from v_j to t. We call this property *path monotonicity.*

Summarizing, we get the following.

Theorem 3.2 *There exists an optimum which satisfies the following properties for all $1 \leq j \leq m$.*

1. *The commodity-j is sent along a shortest $v_j - t$ path according to some distance function l'.*
2. *If two commodities share an edge then they share all the subsequent edges on their paths to t thus the support graph is a tree.*
3. *The discount types along the path are non-decreasing as we go from v_j to t.*

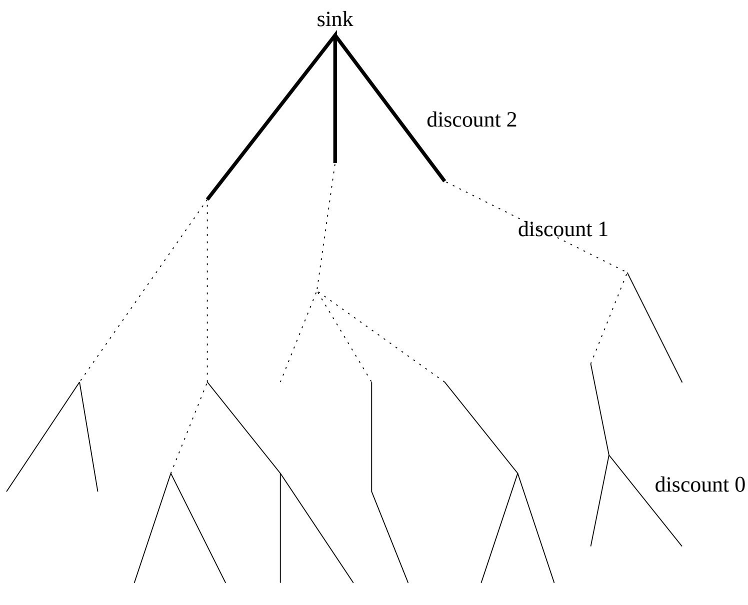

Figure 1: Structure of an integer optimum.

Figure 1 illustrates the structure of an optimum assuming 3 discount types.

4 Integer Linear Program Formulation

We introduce a variable z_e^i for each $e \in E$ and for each $0 \le i < k$, such that, $z_e^i = 1$ if we are using discount-i on edge e and 0 otherwise. The variable $f_{e;i}^j$ is the flow of commodity j on edge e using discount-i.

$$\begin{array}{llll}
\min & \sum_{e\in E}\sum_{i=0}^{k-1} p_i z_e^i l_e & + \sum_{v_j\in\mathcal{S}}\sum_{e\in E}\sum_{i=0}^{k-1} \mathtt{dem}_j f_{e;i}^j r_i l_e & \\
\multicolumn{4}{l}{\text{subject to:}} \\
(i) & \sum_{e\in\delta^+(v_j)}\sum_{i=0}^{k-1} f_{e;i}^j & \ge 1 & \forall v_j \in \mathcal{S} \\
(ii) & \sum_{e\in\delta^-(v)}\sum_{i=0}^{k-1} f_{e;i}^j & = \sum_{e\in\delta^+(v)}\sum_{i=0}^{k-1} f_{e;i}^j & \forall v \in V \setminus \{v_j, t\}, 1 \le j \le m \\
(iii) & \sum_{e\in\delta^-(v)}\sum_{i=q}^{k-1} f_{e;i}^j & \le \sum_{e\in\delta^+(v)}\sum_{i=q}^{k-1} f_{e;i}^j & 0 \le q < k, \forall v \in V \setminus \{v_j, t\}, 1 \le j \le m \\
(iv) & f_{e;i}^j & \le z_e^i & \forall e \in E, 0 \le i < k \\
(v) & \sum_{i=0}^{k-1} z_e^i & \ge 1 & \forall e \in E \\
(vi) & z, f & \text{non-negative integers} &
\end{array}$$

The first term in the objective function is the cost of purchasing the various discount types at each edge; we call this the *building* cost. The second term is the cost of sending $\mathtt{dem}_j$ amount of flow from vertex v_j to the sink; this is termed the *routing* cost of the solution.

The first constraint ensures that every source has an outflow of one unit which is routed to the sink. The second is the standard flow conservation constraint, treating each commodity separately. The third constraint enforces the path monotonicity discussed in the preceding section. The fourth simply builds enough capacity, and the fifth ensures that we install at least one cable type on each link. Note that this is valid and does not add to our cost since we have the default cable available for installation at zero fixed cost.

Relaxing the integrality constraints (vi) to allow the variables to take real non-negative values, we obtain the LP relaxation. This LP has a polynomial number of variables and constraints, and can be therefore solved in polynomial time. The LP relaxation gives us a lower bound which we use in our approximation algorithm.

5 Initial setup

The following lemma shows that the cost of solution does not increase by a large factor if we restrict ourselves to rates that are sufficiently different, that is, they decrease by a constant factor.

Let OPT be the optimum value with rates $r_0, r_1, \dots, r_{k-1}$ and corresponding prices $p_0, p_1, \dots, p_{k-1}$. Let $0 < \epsilon < 1$ be a real number. Assume that $\epsilon^{l-1} \ge r_{k-1} > \epsilon^l$. Now, let us create a new instance as follows. Let the new rates be $1, \epsilon, \dots, \epsilon^{l-1}$. And for each i, the price corresponding to ϵ^i be p_j, where r_j is the largest rate not bigger than ϵ^i. Let OPT' be the optimum value of this problem.

Lemma 5.1 $OPT' \leq \frac{1}{\epsilon} OPT$

Proof. Consider an edge e which uses discount-j in OPT. In the solution of the new problem, change its discount type to ϵ^i such that $\epsilon^i \geq r_j > \epsilon^{i+1}$. Thus, for this edge, the price does not increase and the routing cost increases by a factor smaller than $1/\epsilon$. Hence the claim. ∎

Hereafter we assume that the rates $r_0, r_1, \ldots, r_{k-1}$ decrease by a factor at least ϵ for some $0 < \epsilon < 1$, thereby incurring a factor of $1/\epsilon$.

We next filter the available cables some more, incurring another constant factor in the process. We now require that the fixed costs of cables are also sufficiently separated, that is, for all i, $p_i < \epsilon p_{i+1}$. Let OPT'' be the new optimum.

Lemma 5.2 $OPT'' \leq \frac{1}{\epsilon} OPT'$

Proof. Filter the cables so that we have $p_i < \epsilon p_{i+1}$ for all remaining cables i. Wherever OPT' uses a cable type that is no longer available, replace it by the next available cable. (By next available we mean the cable with the cheapest variable cost whose variable cost is at most the current variable cost.) The routing cost never increases, and for each edge the fixed cost increases by no more than a $1/\epsilon$ factor. ∎

6 Algorithm

In an optimum solution, as shown in figure 1, the flow is sent along a shortest-path tree in which there is a subtree rooted at the sink which uses discount-$(k-1)$, then a layer of subtrees which use discount-$(k-2)$, and so on. In our algorithm, we try to identify these trees successively.

We first solve the LP-relaxation described in section 4. We now process this optimum solution, increasing the cost only by a constant factor. We then use the processed optimum as our lower bound.

For any edge e, suppose we buy a cable of type i to a fraction z_e^i. We then also purchase all cables of types lower than i to the same fraction. Hence we define $\hat{z}_e^i = \sum_{j=i}^{k-1} z_e^j$, and we have bought $\hat{z}_e^i$ quantity of cable i at edge e. We call our new solution OPT, and let OPT_{LP} be the original LP optimum. Note that we have only increased the building cost of the solution.

Lemma 6.1 $OPT \leq \frac{1}{1-\epsilon} OPT_{LP}$

Proof. Consider an edge e and a cable type i purchased in the fractional LP optimum to extent z_e^i. For this, we purchase cables of type 0 through $i-1$ to extent z_e^i. The cost of this is $l_e \sum_{j=0}^{i-1} p_j \leq l_e \sum_{j=0}^{i-1} \epsilon^{i-j} p_i \leq \frac{\epsilon}{1-\epsilon} p_i$. Hence our new build cost on this edge corresponding to cable type i is $(\frac{\epsilon}{1-\epsilon} + 1) p_i l_e z_e^i = \frac{1}{1-\epsilon} p_i l_e z_e^i$. Summing over each z_e^i in the LP optimum, the lemma is proved. ∎

Let C_j^i denote the fraction of the routing cost for routing unit flow from v_j to t corresponding to discount-i, that is, $C_j^i = \sum_e f_{e;i}^j r_i l_e$. Hence $\sum_{0 \leq i < k} C_j^i$ is the total routing cost for vertex v_j.

Our algorithm is divided into k stages. In stage $k-i$, we identify edges which are going to use discount-i. The support graph of our routing is going to be a tree rooted at t. Let T_i denote the tree of edges which use discount-j for $j \geq i$. Initially T_k is a singleton node t. In stage $k-i$ we extend T_{i+1} to T_i by adding edges of discount type i, and T_0 is the final solution we output.

6.1 Stage $k-i$ for $i = k-1, k-2, \ldots, 1$

Let v be a source vertex not in the tree T_{i+1} so far constructed. Define $R_j^i := \frac{C_j^0 + \ldots + C_j^{i-1}}{r_{i-1}}$. Let $B(v, R)$ denote, as usual, a ball of radius R centered at vertex v.

For some constant $\gamma > 1$ to be specified later, we build the balls $B(v_j, \gamma R_j^i)$ and use them to select vertices which will get connected to T_{i+1} and hence will be included in T_i.

For all j such that $T_i \cap B(v_j, \gamma R_j^i) \neq \emptyset$, we ignore these balls and define $\mathtt{proxy}_i(v_j)$ to be some arbitrary vertex in $T_i \cap B(v_j, \gamma R_j^i)$. While unprocessed balls remain, do the following: Pick a ball $B(v_j, \gamma R_j^i)$ with smallest radius. Select vertex v_j. Remove all the balls $B(v_l, \gamma R_l^i)$ which intersect this ball. Define $\mathtt{proxy}_i(v_l) := v_j$ to be the proxy of vertex v_l at stage $k-i$.

Now, we are going to find a forest which connects all the selected vertices to T_{i+1}. Let $1 < \delta < \gamma$ be a constant which will also be specified later. Now, we build balls $B(v_j, \delta R_j^i)$ around each selected vertex v_j. We then build an approximately minimum Steiner tree which connects these selected balls to T_{i+1}. More formally, we contract each ball and introduce a new node for it. We also contract T_{i+1} and introduce a node for it. Then we run an approximation algorithm to find a Steiner tree connecting all the selected nodes that has cost at most twice the value of a fractional Steiner tree, i.e., within twice the cost of an LP relaxation for the Steiner tree problem on these nodes (See, e.g., [AKR]). Then, we un-contract the balls and extend the forest to the centers v_j by direct edges between v_j and the vertices in $B(v_j, \delta R_j^i)$ that are included in the forest. Thus we have a tree connecting all the selected vertices v_j to T_{i+1}.

Note that the Steiner tree constructed so far may lead to very high routing costs in the solution. To get around this, we use a construction due to Khuller, Raghavachari and Young [KRY 93] which achieves short paths from a root node while being light.

Definition 6.1 (Light approximate shortest-path tree) *Let $G = (V, E)$ be a graph with a length function $l : E \to \mathbb{R}^+$ and let $t \in V$ be a vertex. Let $\alpha, \beta > 1$ be real numbers. An (α, β)-LAST rooted at t is a tree T in G such that the total length of T is at most α times the length of an MST of G and for any vertex $v \in V$, length of the $v-t$ path along T is at most β times the length of a shortest $v-t$ path in G.*

The Steiner tree constructed can now be transformed into an (α, β)-LAST for some constants $\alpha, \beta > 1$ using the algorithm of [KRY 93]. The edges in the LAST are going to use discount-i. We then uncontract the root component, so that we have a forest of LASTs. Define T_i to be T_{i+1} union with this forest.

In the last stage k, we connect each source v_j not in T_1 to T_1 by a shortest path, using discount-0, thereby extending T_1 to T_0.

6.2 Routing

After we have decided which edges are going to use which discount types, the routing is simple. We have constructed a tree T_0 which connects every source v_j to the sink t. The routing of commodity-j is done along the unique $v_j - t$ path in T_0.

7 Analysis

We use the processed optimum OPT as a lower bound on the integer optimum. Let OPT_{build} denote the total price paid for all discount types in an optimum and let OPT_{route} be the total routing cost paid in that optimum. Thus $OPT = OPT_{build} + OPT_{route}$ is a lower bound on the total cost. In this section we prove that the total building cost is $O(OPT_{build})$ and the total routing cost is $O(k \cdot OPT_{route})$. Thus we establish $O(k)$-approximation for the problem.

7.1 Building cost

We analyze the total price paid for discount-i decided in the stage $k-i$.

Note that in building an (α, β)-LAST from the tree, we incur a factor of at most α. Define $OPT^i_{build} = \sum_e \hat{z}^i_e l_e p_i$ to be the building cost of the processed optimum corresponding to cable type i. We argue that the cost of building the tree is $O(OPT^i_{build})$. Thus, putting all the k stages together, we get that the total building cost is $O(OPT_{build})$.

The following lemma proves that we have sufficient fractional z-value crossing any set separating t and a ball around any source vertex, thus allowing us to attribute the tree cost to OPT^i_{build}.

Lemma 7.1 *Let $S \subset V$ be a set of vertices such that $t \notin S$ and $B(v_j, dR^i_j) \subset S$. Let $\delta(S)$ denote the set of edges with exactly one endpoint in S. Then,*

$$\sum_{q=i}^{k-1} \sum_{e \in \delta(S)} z^q_e \geq 1 - \frac{1}{\delta}.$$

Proof. Assume for the sake of contradiction that the sum is less than $1 - 1/\delta$. So the total flow starting from the source v_j which crosses S using discount types smaller than i is more than $1/\delta$. As it pays at least r_{i-1} per unit distance per unit flow, the total cost is more than

$$\delta \cdot R^i_j \cdot \frac{1}{\delta} \cdot r_{i-1} = \delta \cdot \frac{C^0_j + \ldots + C^{i-1}_j}{r_{i-1}} \cdot \frac{1}{\delta} \cdot r_{i-1} = C^0_j + \ldots + C^{i-1}_j.$$

This is a contradiction, as the total cost spent in discount types smaller than i is exactly $C^0_j + \ldots + C^{i-1}_j$. Hence the lemma. ∎

We now have $\sum_{q=i}^{k-1} z^q_e = \hat{z}^i_e$ by definition. Hence $\sum_{e \in \delta(S)} \hat{z}^i_e \geq 1 - \frac{1}{\delta}$. We claim that this is sufficient to pay for building the LAST at this stage.

We built a LAST which used discount-i. So the building cost of the LAST is p_i times the length of the LAST. The following lemma gives a bound on this cost.

Lemma 7.2 *The cost of the LAST built is $O(OPT^i_{build})$.*

Proof. If we scale up the $\hat{z}$-values in the optimum by a factor $\delta/(\delta - 1)$, lemma [7.1] indicates that we have sufficient $\hat{z}$-value to build a Steiner tree connecting the balls $B(v_j, \delta R^i_j)$ to T_{i+1}. If we use the primal dual method, we incur an additional factor of 2. Thus, its cost will be at most

$$2\frac{\delta}{\delta - 1} p_i \sum_e \hat{z}^i_e \quad \leq \quad 2\frac{\delta}{\delta - 1} OPT^i_{build}.$$

After un-contracting the balls, we extended the forest to centers v_l by direct edges between v_l and the vertices in $B(v_l, \delta R^i_l)$ that were included in the forest. We can account for this extension by using the following observation. For a center v_l, the cost of extension is at most $\frac{\delta}{\gamma - \delta}$ times the cost of the forest inside $B(v_l, \gamma R^i_l)$. Furthermore, during the selection of the vertices, we ensured that for any two selected vertices v_l and v_j, the balls $B(v_l, \gamma R^i_l)$ and $B(v_j, \gamma R^i_j)$ are disjoint. Thus the total cost of the extended tree is at most $1 + \frac{\delta}{\gamma-\delta}$ times the cost of the previous forest. Hence cost of the LAST built is at most $2\alpha\frac{\gamma}{\gamma-\delta}\frac{\delta}{\delta-1}OPT^i_{build}$. This completes the proof. ∎

The total building cost is the sum of building costs at each stage, and we have k such stages. The lower bounds at each stage being disjoint, we have the following.

Lemma 7.3 *The total building cost is $O(OPT_{build})$.*

7.2 Routing cost

After constructing the tree, for each source vertex v_j, we route the corresponding commodity along the unique (v_j, t) path on the tree. We prove that the routing cost for a source v_j is $O(k)$ times the LP lower bound for routing the commodity corresponding to v_j. Thus the total routing cost is $O(k \cdot OPT_{route})$. Let $OPT_j = \sum_i C^i_j$ denote the routing cost per unit flow for v_j in the optimum.

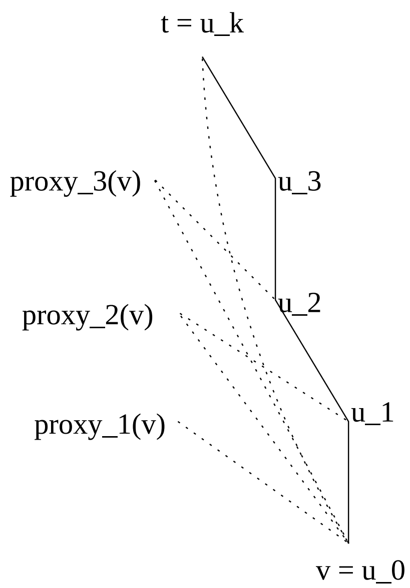

Figure 2: Analysis for routing cost.

Refer to figure 2 for the following analysis.

Lemma 7.4 *For any source vertex v_j, the cost of routing unit amount of its corresponding commodity is $O(k \cdot OPT_j)$.*

Proof. Let the (v_j, t) path along T_0 be $v_j = u_0, u_1, \ldots, u_k = t$ such that the sub-path (u_i, u_{i+1}) uses discount-i for $0 \le i < k$. The u_is need not be distinct. Let $d_T(u_i, u_{i+1})$ be the distance between u_i and u_{i+1} in the tree T_0. So that for v_j, the routing cost per unit flow is $\sum_i r_i d_T(u_i, u_{i+1})$.

For $1 \le i < k$, let $\texttt{proxy}_i(v_j)$ denote the proxy of v_j in stage $k - i$. Moreover, for all j, define $\texttt{proxy}_k(v_j) = t$. We have $d(v_j, \texttt{proxy}_{i+1}(v_j)) \le 2\gamma\frac{C^0_j+\ldots+C^i_j}{r_i} \le 2\gamma\frac{OPT_j}{r_i}$. We also know that $r_i d_T(u_i, u_{i+1}) \le \beta \cdot r_i d(u_i, \texttt{proxy}_{i+1}(v_j))$ because when we constructed the LAST in stage $k - i$, $d(u_i, u_{i+1})$ was at most β times the shortest path connecting u_i to T_{i+1}. Also this shortest path is shorter than $d(u_i, \texttt{proxy}_{i+1}(v_j))$, as $\texttt{proxy}_{i+1}(v_j)$ was in T_{i+1}.

By induction on i we prove that, $r_i d_T(u_i, u_{i+1}) \leq M \cdot OPT_j$ for some constant M.

For the base case when $i = 0$, v_j was connected by a shortest path to T_1. Hence $r_0 d_T(u_0, u_1) \leq r_0 d(v_j, \mathtt{proxy}_1(v_j)) \leq r_0 \cdot 2\gamma \frac{C_j^0}{r_0} \leq 2\gamma OPT_j \leq M \cdot OPT_j$ for sufficiently large M.

Now assume $r_l d_T(u_l, u_{l+1}) \leq M \cdot OPT_j$ for all $l < i$. Using triangle inequality and the induction hypothesis, we get

$$\begin{aligned}
r_i \cdot d_T(u_i, u_{i+1}) &\leq \beta \cdot r_i \cdot d(u_i, \mathtt{proxy}_{i+1}(v_j)) \\
&\leq \beta \cdot r_i \sum_{q=0}^{i-1} d(u_q, u_{q+1}) + \beta \cdot r_i \cdot l(u_0, \mathtt{proxy}_{i+1}(v_j)) \\
&= \beta \sum_{q=0}^{i-1} \frac{r_i}{r_q} \cdot r_q \cdot d(u_q, u_{q+1}) + \beta \cdot r_i \cdot l(v_j, \mathtt{proxy}_{i+1}(v_j)) \\
&\leq \beta \sum_{q=0}^{i-1} \epsilon^{i-q} \cdot M \cdot OPT_j + \beta \cdot 2\gamma OPT_j \\
&\leq (\frac{\beta\epsilon}{1-\epsilon} M + 2\beta\gamma) OPT_j \\
&\leq M \cdot OPT_j
\end{aligned}$$

for proper choices of the parameters. This completes the induction.

∎

Summing the routing cost bound over all source vertices v_j, we obtain that the total routing cost is no more than $O(k \cdot OPT_{route})$.

References

[AKR] A. Agrawal, P. Klein and R. Ravi "When trees collide: An approximation algorithm for the generalized Steiner problem on networks," In SIAM Journal of Computing,24(3):440-456, 1995.

[AZ98] M. Andrews and L. Zhang, "The access network design problem," In Proc. of the 39th Ann. IEEE Symp. on Foundations of Computer Science, 42-49, October 1998.

[AA97] B. Awerbuch and Y. Azar, "Buy at bulk network design," Proc. 38th Ann. IEEE Symposium on Foundations of Computer Science, 542-547, 1997.

[HRS00] R. Hassin, R. Ravi and F.S. Salman, "Approximation algorithms for a capacitated network design problem," To appear in the Proc. of the APPROX 2000.

[KRY 93] S. Khuller, B. Raghavachari and N. E. Young, "Balancing minimum spanning and shortest path trees," Algorithmica, **14**, 305-322, 1993.

[MP 94] Y. Mansour and D. Peleg, "An approximation algorithm for minimum-cost network design," The Weizman Institute of Science, Rehovot, 76100 Israel, Tech. Report CS94-22, 1994; Also presented at the DIMACS workshop on Robust Communication Networks, 1998.

[SCR+97] F.S. Salman, J. Cheriyan, R. Ravi and S. Subramanian, "Buy-at-bulk network design: Approximating the single-sink edge installation problem," Proc. of the 8th Ann. ACM-SIAM Symposium on Discrete Algorithms, 619-628, 1997.