On the Integrality Gap of a Natural Formulation of the Single-Sink Buy-at-bulk Network Design Problem[§]

Naveen Garg[*] Rohit Khandekar[*] Goran Konjevod[†] R. Ravi[‡]
F.S. Salman [‡] Amitabh Sinha[‡¶]

November 3, 2000

Abstract

We study two versions of the single sink *buy-at-bulk* network design problem. We are given a network and a single sink, and several sources which demand a certain amount of flow to be routed to the sink. We are also given a finite set of cable types which have different cost characteristics and obey the principle of economies of scale. We wish to construct a minimum cost network to support the demands, using our given cable types. We study a natural integer program formulation of the problem, and show that its integrality gap is $O(k)$, where k is the number of cables. As a consequence, we also provide an $O(k)$ approximation algorithm.

[*]Department of Computer Science and Engineering, Indian Institute of Technology, New Delhi, India. `email: {naveen, rohitk}@cse.iitd.ernet.in`

[†]Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213-3890. `email: konjevod@andrew.cmu.edu`

[‡]GSIA, Carnegie Mellon University, Pittsburgh PA 15213-3890. `email:{ravi, fs2c, asinha}@andrew.cmu.edu`

[¶]Corresponding author

[§]This research was supported by a faculty development grant awarded to R. Ravi by the Carnegie Bosch Institute, Carnegie Mellon University, Pittsburgh PA 15213-3890.

1 Introduction

1.1 Motivation

We study two network design problems which often arise in practice. Consider a network consisting of a single server and several clients. Each client wishes to route a certain amount of flow to the server. The cost per unit flow along an edge is proportional to the edge length. However, we can reduce the cost per unit flow of routing by paying a certain fixed cost (again proportional to the length of the edge). We call the problem of finding a minimum cost network supporting the required flow the *deep-discount* problem.

Alternatively, at each edge we might be able to pay for and install a certain capacity, and then route flow (up to the installed capacity) for free. The problem of finding a minimum cost network in this scenario is called the *buy-at-bulk* network design problem [SCR+].

1.2 Our results

The two problems are in fact equivalent up to a small loss in the value of the solution. In this paper, we focus on the deep-discount problem. We study the structure of the optimum solution, and show that an optimal solution exists which is a tree. We provide a natural IP formulation of the problem, and show that it has integrality gap of the order of the number of cables. We also provide a polynomial time approximation algorithm by rounding the LP relaxation.

1.3 Previous work

Mansour and Peleg [MP] gave an $O(\log n)$ approximation for the single cable type case of the single sink buy-at-bulk problem for a graph on n nodes. They achieved this result by using a low-weight, low-stretch spanner construction [ADD+].

Designing networks using facilities that provide economies of scale has also attracted interest in recent years. Salman *et al* [SCR+] gave an $O(\log D)$ approximation algorithm for the single sink buy-at-bulk problem in Euclidean metric spaces, where D is the total demand. Awerbuch and Azar [AA] 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. This improves to $O(\log n \log\log n)$ using the improved tree metric construction of Bartal [Bar]. Salman *et al* also gave a constant approximation in [SCR+] for the single cable type case using a LAST construction [KRY] in place of the spanner construction used in [MP]. The approximation ratio was later improved by Hassin, Ravi and Salman [HRS].

Andrews and Zhang [AZ] 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.

An improved approximation to the problem we study was obtained simultaneously but independently by Guha, Meyerson and Munagala [GMM1], who designed a constant factor approximation algorithm. Their algorithm is combinatorial and is based on their prior work on the access network design problem [GMM2], as opposed to our focus on the LP relaxation and its integrality gap.

2 Problem definitions and preliminaries

2.1 The deep-discount problem

Let $G = (V, E)$ be a graph with edge-lengths $l : E \to I\!R^+$. We are given source vertices $\{v_1, \dots, v_m\} = \mathcal{S} \subseteq V$ and a sink vertex $t \in V$. The source vertices require $\{\mathtt{dem}_1, \dots, \mathtt{dem}_m\}$ units of flow to be routed to the sink respectively. We also have a set of k cable types $\{\kappa_0, \kappa_1, \dots, \kappa_{k-1}\}$ available for us to purchase and install. Each cable κ_i has an associated fixed cost p_i and a variable cost r_i. If we install cable κ_i at edge e and route f_e flow through it, the contribution to our cost is $l_e(p_i + f_e r_i)$. We may therefore view the installation of cable κ_i at an edge as paying a fixed cost $p_i l_e$ in order to obtain a discounted rate r_i of routing along this edge. The problem is to route the flow at minimum cost.

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. (We may scale our cost functions so that this is true in general.) Without loss of generality, $p_0 < p_1 < \dots < p_{k-1}$.

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

As before, we have sources with flow routing demands and a sink. We have available to us k different cable types, 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 (BB for short). It is **NP**-hard even when $k = 1$ [SCR+].

2.3 Approximate equivalence of BB and DD

Suppose we are given a BB instance $BB = (G, c, u)$ on a graph G with k cable types having costs and capacities $(1, 1), (c_1, u_1), \dots, (c_{k-1}, u_{k-1})$. 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 $DD(BB)$.

Conversely, given a DD instance $DD = (G,p,r)$, we transform it into a BB instance $BB(DD)$ 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 $BB(DD(BB)) = BB$ and $DD(BB(DD)) = DD$, 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. We then have the following lemmas.

Lemma 2.1 $BB \leq DD(BB)^*$.

Lemma 2.2 $DD \leq 2BB(DD)^*$.

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

2.4 Structure of an optimum solution to the deep-discount problem

Suppose in an optimal solution, an edge e uses discount-i (Clearly we will use only one discount type on an edge.). Define a new length function $l'_e := r_i l_e$. Once the fixed cost 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 assume that the support graph of the flow is a tree (in particular, a shortest path tree under l').

The cost of routing f units of flow on an edge e using discount-i is $l_e(p_i + r_i f)$. So the discount type corresponding to minimum cost depends only on f and is given by $\texttt{type}(f) := \texttt{minarg}_i\{p_i + r_i f | 0 \leq i < k\}$. As a consequence, we can prove the following.

Lemma 2.3 *The function* $\texttt{type}(f)$ *defined above is non-decreasing in* f.

Theorem 2.4 *There exists an optimum solution to the deep-discount problem which satisfies the following properties.*

1. *The support graph of the solution is a tree.*
2. *The discount types are non-decreasing along the path from a source to the root.*

Similar results were proved independently (but differently) by [AZ] and [GMM1].

3 Linear program formulation and rounding

3.1 Overview of the algorithm

First we formulate the deep-discount problem as an integer program. We then take the linear relaxation of the IP, and solve it to optimality. We now use the LP solution to construct our solution. We have already seen that the solution is a layered tree. We construct a tree in a *top-down* manner, starting from the sink. We iteratively augment the tree by adding cables of the next available lower discount type. At each stage we use an argument based on the values of the decision variables in our optimal LP solution to charge the cost of building our tree to the LP cost. We next bound the routing costs by an argument which essentially relies on the fact that the tree is layered and that distances obey the triangle inequality.

3.2 Integer program formulation

We begin by replacing each undirected edge by a pair of anti-parallel directed arcs, each having the same length as the original (undirected) edge. We introduce a variable z_e^i for each $e \in E$ and for each $0 \leq i < k$, 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. For a vertex set S (or a singleton vertex v), we define $\delta^+(S)$ to be the set of arcs leaving S. That is, $\delta^+(S) = \{(u,v) \in E : u \in S, v \notin S\}$. Analogously, $\delta^-(S) = \{(u,v) \in E : u \notin S, v \in S\}$.

$$\min \quad \sum_{e \in E} \sum_{i=0}^{k-1} p_i z_e^i l_e \quad + \sum_{v_j \in \mathcal{S}} \sum_{e \in E} \sum_{i=0}^{k-1} \texttt{dem}_j f_{e;i}^j r_i l_e$$

subject to:

$$\begin{array}{llll}
(i) & \displaystyle\sum_{e \in \delta^+(v_j)} \sum_{i=0}^{k-1} f_{e;i}^j & \geq 1 & \forall v_j \in \mathcal{S} \\
(ii) & \displaystyle\sum_{e \in \delta^-(v)} \sum_{i=0}^{k-1} f_{e;i}^j & = \displaystyle\sum_{e \in \delta^+(v)} \sum_{i=0}^{k-1} f_{e;i}^j & \forall v \in V \setminus \{v_j, t\}, 1 \leq j \leq m \\
(iii) & \displaystyle\sum_{e \in \delta^-(v)} \sum_{i=q}^{k-1} f_{e;i}^j & \leq \displaystyle\sum_{e \in \delta^+(v)} \sum_{i=q}^{k-1} f_{e;i}^j & 0 \leq q < k, \forall v \in V \setminus \{v_j, t\}, 1 \leq j \leq m \\
(iv) & f_{e;i}^j & \leq z_e^i & \forall e \in E, 0 \leq i < k \\
(v) & \displaystyle\sum_{i=0}^{k-1} z_e^i & \geq 1 & \forall e \in E \\
(vi) & z, f & \text{non-negative integers} &
\end{array}$$

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 total cost (over all vertices

v_j) of sending $\mathtt{dem}_j$ amount of flow from vertex v_j to the sink; we call this the *routing* cost of the solution. These two components of the cost of an optimal solution are referred to as OPT_{build} and OPT_{route} respectively.

The first set of constraints ensures that every source has an outflow of one unit which is routed to the sink. The second is the standard flow conservation constraints, treating each commodity separately. The third set of constraints enforces the path monotonicity discussed in Theorem 2.4(2), and is therefore valid for the formulation. The fourth simply builds enough capacity, and the fifth ensures that we install at least one cable type on each arc. 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.

4 The rounding algorithm

4.1 Pruning the set of available cables.

We begin by pruning our set of available cables, and we show that this does not increase the cost by more than a constant factor. This pruning is useful in the analysis.

Let OPT be the optimum value when the rates are $r_0, r_1, \ldots, r_{k-1}$ and the corresponding prices are $p_0, p_1, \ldots, p_{k-1}$. Let $\epsilon \in (0,1)$ be a real number. Assume that $\epsilon^{l-1} \geq r_{k-1} > \epsilon^l$. Now let us create a new instance as follows. Let the new rates be $1, \epsilon, \ldots, \epsilon^{l-1}$. For each i, let 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 new problem. We then have the following.

Lemma 4.1 $OPT'_{route} \leq \frac{1}{\epsilon} OPT_{route}$.

Since $OPT'_{build} \leq OPT_{build}$, we have as a consequence that $OPT' \leq \frac{1}{\epsilon} OPT$. 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 $1/\epsilon$ in the value of the solution we obtain. We may therefore prune the cable set before we set up and solve the LP.

4.2 Building the solution: Overview

Recall that G is our input graph, and k is the number of cable types. We also have a set of parameters $\{\alpha, \beta, \gamma, \delta, \epsilon\}$, the effect of which will be studied in the analysis in Section 5.

We build our tree in a top-down manner. We begin by defining T_k to be the singleton vertex $\{t\}$, the sink. We then successively augment this tree by adding cables of discount type i to obtain T_i, for i going down $k-1, k-2, \ldots, 1, 0$. Our final tree T_0 is the solution we output.

Our basic strategy for constructing the tree T_i from T_{i+1} is to first identify a subset of demand sources that are not yet included in T_{i+1} by using information from their contributions to the routing cost portion of the LP relaxation. We build a ball $B(v, R) = \{u \in V : d(u, v) \leq R\}$ of radius R (computed using the LP solution) centered at vertex v for each such vertex, where $d(u, v)$ is the length of a shortest $u-v$ path. We order these candidate nodes in non-increasing order of the radius of their balls.

We then choose a maximal set of non-overlapping balls going forward in this order. This intuitively ensures that any ball that was not chosen can be charged for its routing via the smaller radius ball that overlapped with it that is included in the current level of the tree. We contract each ball and the root component into singleton vertices.

After choosing such a subset of as yet unconnected nodes, we build an approximately minimum building cost Steiner tree with these nodes as terminals and the (contracted) tree T_{i+1} as the root. The balls used to identify this subset now also serve to relate the building cost of the Steiner tree to the fractional optimum.

Finally, in a third step, we convert the approximate Steiner tree rooted at the contracted T_{i+1} to a LAST (light approximate shortest-path tree). This ensures that all nodes in the tree are within a constant factor of their distance from the root T_{i+1} in this LAST without increasing the total length (and hence the building cost) of the tree by more than a constant factor. We use a result by Khuller, Raghavachari and Young [KRY] which converts a minimum spanning tree in a graph to a tree costing (at most) α times more, but where the path to any node is no more than β times its shortest path in the original graph, for some constants α and β. This step is essential to guarantee that the routing cost via this level of the tree does not involve long paths and thus can be charged to within a constant factor of the appropriate LP routing cost bound.

We install type$-i$ cables on the edges in this LAST. When we un-contract the root component, we break up the LAST into a forest where each subtree is rooted at some node in the un-contracted T_{i+1}. Now T_i is simply defined to be the union of T_{i+1} and our LAST.

In the last stage, 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.

5 Analysis

5.1 Building cost

We analyze the total price paid for installing discount-i cables when we augment the tree T_{i+1} to T_i.

Algorithm `Deep-discount`$(G, K, \alpha, \beta, \gamma, \delta, \epsilon)$

G: input graph

K: set of cables

$\alpha, \beta, \gamma, \delta, \epsilon$: parameters

1. Prune the set of available cables as described in 4.1.
2. Solve the LP relaxation of the IP described in 3.2 to get (z, f)
3. $T_k = \{t\}$.
4. for $i = k-1, k-2, \ldots, 1$:

    Define $S_i := \emptyset$

    $\forall v_j \notin T_{i+1}$:

    $R_j^i = \frac{C_j^0 + \ldots + C_j^{i-1}}{r_{i-1}}$, where $C_j^i = \sum_e f_{e;i}^j r_i l_e$.

    If $T_{i+1} \cap B(v_j, \gamma R_j^i) \neq \emptyset$,

    `proxy`$_i(v_j)$:= any (arbitrary) vertex in $T_{i+1} \cap B(v_j, \gamma R_j^i)$

    $S_i := S_i \cup \{v_j\}$.

    Order the remaining vertices, say L, according to nondecreasing radii.

    While $L \neq \emptyset$:

    Let $B(v_j, \gamma R_j^i)$ be the smallest radius ball in L.

    $\forall u \in L \cap B(v_j, \gamma R_j^i)$:

    `proxy`$_i(u) := v_j$

    $L := L \setminus \{u\}$

    $L := L \setminus \{v_j\}$

    $S_i := S_i \cup \{v_j\}$

    Comment: S_i is the set of sources chosen to be connected at this level.

    Contract T_{i+1} to a singleton node t_{i+1}.

    Build a Steiner tree ST_i with $S_i \cup \{t_{i+1}\}$ as terminals.

    Use cable type i for these edges.

    Convert ST_i into an (α, β)-LAST, denoted $LAST_i$.

    Define $T_i := T_{i+1} \cup LAST_i$.
5. For every source vertex $v_j \notin T_1$:

    Compute a shortest path P from v_j to T_1.

    Augment T_1 by including the edges in P.

    Use cable type 0 on the edges in P.

    $T_0 := T_1$.
6. Route along shortest paths in T_0. This is the solution we output.

Figure 1: The algorithm

Note that in building $LAST_i$ from the Steiner tree ST_i, we incur a factor of at most α in the building cost. We argue that the cost of building the tree at the current stage is $O(OPT_{build})$. Thus, putting all the k stages together, we get that the total building cost is $O(k \cdot OPT_{build})$.

For any source vertex v, the following Lemma proves that there is sufficient fractional z-value crossing a ball around v to allow us to pay for an edge crossing the ball. Since the LP optimum pays for this z, we can charge the cost of our edge to this fractional z and hence obtain our approximation guarantee.

Lemma 5.1 *Let $S \subset V$ be a set of vertices such that $t \notin S$ and $B(v_j, \delta R_j^i) \subset S$. Then,*

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

We build 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 5.2 *The cost of the LAST built is $O(OPT_{build})$.*

Proof. If we scale up the z-values in the optimum by a factor $\delta/(\delta-1)$, Lemma 5.1 indicates that we have sufficient z-value of types i or higher to build a Steiner tree connecting the balls $B(v_j, \delta R_j^i)$ to T_{i+1}. If we use the primal dual method, we incur an additional factor of 2 in the cost of the Steiner tree as against the LP solution z-values. Thus, its cost will be at most

$$2\frac{\delta}{\delta-1} p_i \sum_{q=i}^{k-1} \sum_{e} z_e^q \quad \leq \quad 2\frac{\delta}{\delta-1} OPT_{build}.$$

After un-contracting the balls, we extended the forest to centers v_l by direct edges between v_l and the vertices in $B(v_l, \delta R_l^i)$ 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 $B(v_l, \gamma R_l^i)$. Furthermore, during the selection of the vertices, we ensured that for any two selected vertices v_l and v_j, the balls $B(v_l, \gamma R_l^i)$ and $B(v_j, \gamma R_j^i)$ 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 the cost of the Steiner tree built is at most $2\frac{\gamma}{\gamma-\delta}\frac{\delta}{\delta-1} OPT_{build}$. Subsequently, the cost of the LAST built from this tree is at most $2\alpha\frac{\gamma}{\gamma-\delta}\frac{\delta}{\delta-1} OPT_{build}$. ∎

The total building cost is the sum of building costs at each of the k stages.

Lemma 5.3 *The total building cost is $O(k \cdot OPT_{build})$.*

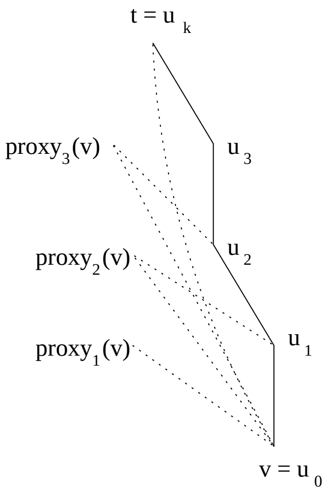

Figure 2: Analysis for routing cost.

5.2 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. Let $OPT_j = \sum_i C_j^i$ denote the routing cost per unit flow for v_j in the optimum.

Lemma 5.4 *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 $0 \leq i < k$ (refer Figure 2). The u_i's 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. Then, for v_j, the routing cost per unit flow is $\sum_i r_i d_T(u_i, u_{i+1})$.

For $1 \leq i < k$, 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 $d(v_j, \texttt{proxy}_{i+1}(v_j)) \leq 2\gamma \frac{C_j^0 + \ldots + C_j^i}{r_i} \leq 2\gamma \frac{OPT_j}{r_i}$. We also know that $r_i d_T(u_i, u_{i+1}) \leq \beta \cdot r_i d(u_i, \texttt{proxy}_{i+1}(v_j))$ 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 $r_0 d_T(u_0, u_1) \leq r_0 d(v_j, \texttt{proxy}_1(v_j)) \leq r_0 \cdot 2\gamma \frac{C_j^0}{r_0} \leq 2\gamma OPT_j \leq M \cdot OPT_j$ 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

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

where $M \geq \frac{2\beta\gamma(1-\epsilon)}{1-\epsilon(1+\beta)}$. This completes the induction. Summing over all edges in the path from v_j to t, we get the statement of the Lemma. ∎

Summing the routing cost bound over all source vertices v_j, we obtain that the total routing cost is $O(k \cdot OPT_{route})$.

6 Conclusion

The exact approximation factor of our algorithm depends on the parameters. If we set $(\alpha, \beta, \gamma, \delta, \epsilon)$ to be $(7, \frac{4}{3}, 3, 2, \frac{1}{5})$ respectively, we obtain an approximation factor of $60k$ for both components of the cost function. We contrast this with the approximation factor of around 2000 obtained by Guha, *et al.* [GMM1]. We note that we can adapt a technique used by them to cut down the approximation ratio of our building cost to a constant. The constant works out to roughly 525 if we use the same values for our parameters. The running time of our algorithm is dominated by the time to solve an LP with $O(mnk)$ constraints and variables.

Our main open question is the exact integrality gap of this problem, whether it is a constant, $O(k)$, or something in between. The question of getting even better approximation ratios for this problem of course remains open. The problem can be generalized to allow different source-sink pairs; for this problem the current state of the art is a polylogarithmic approximation [AA].

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.

[ADD+] I. Althöfer, G. Das, D. Dobkin, D. Joseph and J. Soares, "On sparse spanners of weighted graphs", In Discrete and Computational Geometry, 9:81-100, 1993.

[AZ] 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.

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

[Bar] Y. Bartal, "On approximating arbitrary metrics by tree metrics", Proc. 30th Ann. ACM Symposium on Theory of Computing, 1998.

[GMM1] S. Guha, A. Meyerson and K. Munagala, "Improved Combinatorial Algorithms for Single Sink Edge Installation Problems", Stanford University Tech Report, CS-TN-00-96, 2000.

[GMM2] S. Guha, A. Meyerson and K. Munagala, "Hierarchical Placement and Network Design Problems", Proc. 41st Ann. IEEE Symposium on Foundations of Computer Science, 2000.

[HRS] R. Hassin, R. Ravi and F.S. Salman, "Approximation algorithms for a capacitated network design problem", Proc. of the APPROX 2000, 167-176, 2000.

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

[MP] 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.

[RS] R. Ravi and F.S. Salman, "Approximation Algorithms for the Traveling Purchaser Problem and its variants in network design", Proc. of the European Symposium on Algorithms, 29-40, 1999.

[SCR+] 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.

This research was sponsored in part by National Science Foundation (NSF) grant no. CCR-0122581.