

Approximation Algorithms for a Capacitated Network Design Problem

R. Hassin^{1*} and R. Ravi^{2**} and F. S. Salman^{3***}

¹ Department of Statistics and Operations Research, Tel-Aviv University, Tel Aviv 69978, Israel.

`hassin@math.tau.ac.il`

² GSIA, Carnegie Mellon University, Pittsburgh, PA 15213-3890.

`ravi@cmu.edu`

³ GSIA, Carnegie Mellon University, Pittsburgh, PA 15213-3890.

`fs2c@andrew.cmu.edu`

Abstract. We study a network loading problem with applications in local access network design. Given a network, the problem is to route flow from several sources to a sink and to install capacity on the edges to support flows at minimum cost. Capacity can be purchased only in multiples of a fixed quantity. All the flow from a source must be routed in a single path to the sink. This NP-hard problem generalizes the Steiner tree problem and also more effectively models the applications traditionally formulated as capacitated tree problems. We present an approximation algorithm with performance ratio $(\rho_{ST} + 2)$ where ρ_{ST} is the performance ratio of any approximation algorithm for minimum Steiner tree. When all sources have the same demand value, the ratio improves to $(\rho_{ST} + 1)$ and in particular, to 2 when all nodes in the graph are sources.

1 Introduction

We consider a single-sink multiple-source routing and capacity installation problem where capacity can be purchased in multiples of a fixed quantity. In telecommunication network design this corresponds to installing transmission facilities such as fiber-optic cables on the edges of a network, and in transportation networks this applies to assigning vehicles of fixed capacity to routes. Topological design of communication networks is usually carried in stages due to the complexity of the problem. One of the fundamental stages is the design of a local access network which links the users to a switching center. The problem we study models this stage of the planning process.

Problem statement. We are given an underlying undirected graph $G = (V, E)$, $|V| = n$. A subset S of nodes is specified as sources of traffic and a single sink t is specified. Each source node $s_i \in S$ has a positive integer-valued demand

* This work was done when this author visited GSIA, Carnegie Mellon University.

** Supported by an NSF CAREER grant CCR-9625297.

*** Supported by an IBM Corporate Fellowship.

dem_i . All the traffic of each source must be routed to t via a single path, that is flow cannot be bifurcated. The edges of G have lengths $\ell : E \rightarrow \mathfrak{R}^+$. Without loss of generality, we may assume that for every pair of nodes v, w , we can use the shortest-path distance $dist(v, w)$ as the length of the edge between v and w ; Therefore, we take the metric completion of the given graph and assume all edges from the complete graph are available. Capacity must be installed on the edges of the network by purchasing one or more copies of a facility, which we refer to as the “cable” based on the telecommunication application. The cable has per unit length cost c and capacity u . Without loss of generality we can assume $c = 1$.

The problem is to specify for each source s_i , a path to t to route demand dem_i such that cables installed on each edge of the network provide sufficient capacity for the flow on the edge, and total cost of cables installed is minimized. Notice that we allow paths from different sources to share the capacity on the installed cables, the only restriction being that the capacity installed on an edge is at least as much as the *total* demand routed through this edge.

The problem is NP-hard since the problem with cable capacity large enough to hold all of the demand is equivalent to a Steiner tree problem with the sources and the sink as the terminal nodes.

Previous Work. This problem has been studied in the literature as the *network loading problem*, together with its variations such as the multicommodity and multiple facility cases. For a survey on exact solution methods see the chapter on multicommodity capacitated network design by Gendron, Crainic and Frangioni in [SS99]. In spite of the recent computational progress, the size of the instances that can be solved to optimality in reasonable time is still far from the size of real-life instances.

In this paper we focus on obtaining approximation algorithms. A constant factor approximation for this problem was obtained by Salman et al. in [SCR+97] by applying the method of Mansour and Peleg [MP 94] to the case of single sink, and single cable type. The main algorithm of Mansour and Peleg applies to the multiple-source multiple-sink single cable problem with approximation ratio $O(\log n)$ in an n -node graph. By using a Light Approximate Shortest-Path Tree (LAST) [KRY 93] instead of a more general-purpose spanner in this algorithm, Salman et al. obtained a 7-approximation algorithm for the single-sink version. When all the nodes in the input network except the sink node are source nodes, the approximation ratio in [SCR+97] reduces to $(2\sqrt{2} + 2)$. Another constant factor approximation algorithm for this problem also follows from the work of Andrews and Zhang [AZ98] who gave an $O(k^2)$ -approximation algorithm for the single sink problem with k cable types, but the resulting constant factor is rather high.

Results. In this paper, we improve the approximation ratio to $(\rho_{ST} + 2)$ by routing through a network that is built on an approximate Steiner tree, with performance ratio ρ_{ST} . The idea is to utilize the Steiner tree when demand is low compared to the cable capacity and when demand accumulates to a value close to the cable capacity, it is sent directly to the sink. For the special case

when demand of each source is uniform, the approximation ratio improves to $(\rho_{ST} + 1)$. When all the nodes in the input network except the sink node are source nodes, the approximation ratio reduces to 3 with non-uniform demands, and to 2, for uniform demands.

Our study was also motivated by obtaining better approximation algorithms for the capacitated MST problem [Pap78,AG88,KB83,CL83,S83]: Given an undirected edge-weighted graph with a root node and a positive integer u , the problem is to find the minimum weight tree such that every subtree hanging off the root node has at most u nodes in it. This problem has been cited [KR98,AG88] to model the local access network design problem when every non-sink node is required to route a single unit of demand to the sink via cables each of capacity at most u . The requirement that every demand has to send its unit flow via a single path is modeled as requiring a tree as the solution. However, if routing these demands at nodes is not a concern, we can still enforce the non-bifurcating requirement for the demands without requiring that the solution be a tree. This reformulation leads exactly to our single cable problem in the uniform case with all nodes being sources. Our 2-approximation algorithm for this problem is then a better solution than the best-known 3-approximation [AG88] for the corresponding capacitated MST formulation. In the nonuniform demand case, our $(\rho_{ST} + 2)$ -approximation is better than the best known 4-approximation presented in [AG88] in addition to handling the Steiner version that does not require all non-sink nodes be source nodes.

In the next two sections, we present the algorithms for the case of uniform and non-uniform demands, respectively. We close with an extension of the local access design problem.

2 Uniform Demand

We first present an approximation algorithm for the case when every source has the same demand. Without loss of generality, we assume demand equals one for each source.

We can outline the algorithm as follows. First we construct an approximate Steiner tree with terminal set $S \cup \{t\}$ and cost $dist(e)$ on each edge e in polynomial time. Let T be the approximate Steiner tree with worst-case ratio ρ_{ST} ¹. Let the tree T be rooted at the sink node t . Next, we identify subtrees of T such that total demand in a subtree equals the cable capacity u . We then route the total demand within a subtree directly to the sink from the node of the subtree closest to the sink. The subtrees collected by the algorithm may contain common nodes but have disjoint source sets.

For a formal statement of the algorithm, we need the following definitions. Let the level of a node be the number of tree edges on its path to the root. The parent of a node v is the node adjacent to it on the path from v to t . For each node v , let T_v denote the subtree of T rooted at v and $D(T_v)$ denote the total

¹ The MST is a 2-approximate solution. Better approximation ratios are known, e.g., a 1.55-approximation was given recently in [RZ00].

unprocessed demand in T_v . Let R be the set of unprocessed source nodes. Then, $D(T_v) = \sum_{s_i \in R \cap T_v} dem_i = |R \cap T_v|$. The Algorithm Uniform below outputs a routing for the demand from each source to the sink, and the number of cables that are installed to support the routing.

Algorithm Uniform:

Initialize: $R = S$

Main step:

- Pick a node v such that $D(T_v) \geq u$ and level of v is maximum.
- If $v = t$ or $D(T_t) < u$, then go to the final step.
- Pick a node, say w in $R \cap T_v$ such that $dist(w, t)$ is minimum, as a “hub” node.
- Let $C = \{w\}$.
- Collect source nodes in C (Details given below).
- Add edge (w, t) to the network and install one copy of the cable on (w, t) .
- Route demand of each source in C to the hub node w via the unique paths in T .
- Route demand of C at the hub directly to the sink on (w, t) .
- Remove C from R , and set $C = \emptyset$.
- If R is not empty, repeat the main step.
- If R is empty, go to the final step.

Final step:

- If $R \neq \emptyset$, then route all the demand in R to t via their path in T .
- For all edges e of T ,
 - Cancel the maximal possible amount of flow of equal value in opposite directions such that total flow will not exceed u .
- Install one copy of cable on the edges of T which have positive flow.

Collect source nodes:

Add v to C , if $v \in R$.

Let v_1, \dots, v_k be the children of v .

If $w \neq v$, then

- Let v_p be the child of v such that the hub node w is in T_{v_p} .
- Add $T_{v_p} \cap R$ to C .

While $|C| < u$,

- Pick an unprocessed child of v , say v_i .
- If $D(T_{v_i}) + |C| \leq u$, then
- Add $T_{v_i} \cap R$ to C .
- Else, (T_{v_i} is collected partially)
- Scan T_{v_i} depth-first.
- Add sources in $R \cap T_{v_i}$ to C until $|C| = u$.

Return C .

Lemma 1. *The algorithm routes demand such that flow on any edge of the tree T is at most the cable capacity u .*

Proof. Consider an edge e of T . Let v be the incident node on e with higher level (see Figure 1). Flow on e is determined by the total flow coming out of T_v and

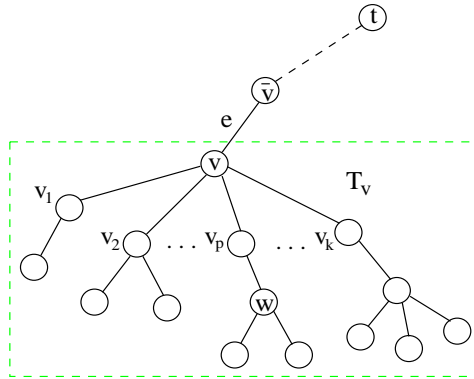


Fig. 1. Subtree T_v and its children.

going into T_v . Our proof is based on these two claims:

Claim 1: Total flow going out of T_v is at most $u - 1$.

Claim 2: Total flow coming into T_v is at most $u - 1$.

To prove claims 1 and 2, we consider two cases based on how the sources in T_v are assigned to hub nodes by the algorithm. A *partially assigned subtree* has at least one of its source nodes collected in a set C and has at least one source node not in C .

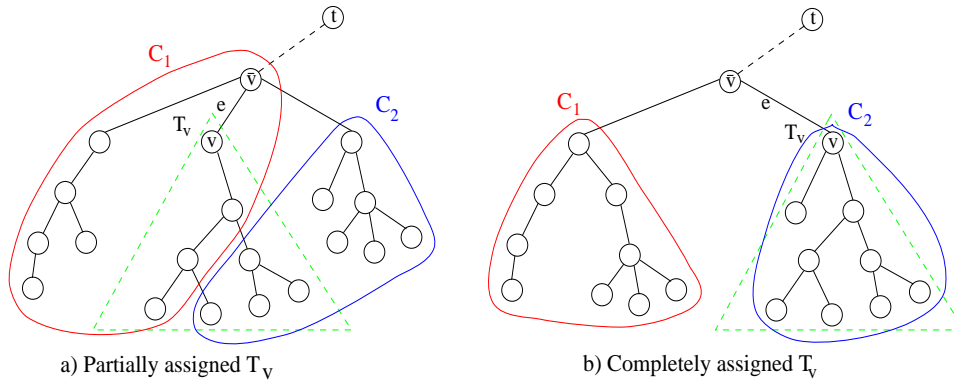


Fig. 2. Examples of partially and completely assigned subtrees.

Suppose T_v is partially assigned (see Figure 2). The first time flow goes out of T_v , a subtree $T_{\bar{v}}$ with \bar{v} at a smaller level than v is being processed by the algorithm. Due to the subtree selection rule, we can conclude that T_v has remaining demand strictly less than u . Therefore, total outflow from T_v will be at most $u - 1$. Hence, Claim 1 holds in this case.

The reason Claim 2 holds is as follows. When there exists an inflow into T_v , flow is accumulated at a hub node in T_v . Since the algorithm accumulates a flow of exactly u at any hub node, a flow of at most $u - 1$ will go into T_v . The algorithm first picks a subtree and a hub node in it, and collects demand starting with the subtrees of T_v . Therefore, the algorithm will not collect sources out of T_v , unless all the sources in T_v have already been collected. This implies that once flow enters T_v , none of the nodes in T_v will become a hub node again and hence flow will not enter T_v again.

Now let us assume that T_v is not partially assigned. Then all the sources in T_v are collected in the same set by the algorithm. If these sources are routed to a hub node out of the subtree, then outflow is at most $u - 1$. If the sources are routed to a hub node in the subtree, then inflow is at most $u - 1$. Inflow or outflow occurs only once. Thus, Claims 1 and 2 hold in this case, too.

For any edge of T , flow in one direction does not exceed u , by Claims 1 and 2. When there exists flow in both directions in an edge with total value greater than u , we cancel flow of equal value in opposite directions such that total flow will not exceed u . Cancelling flow will lead to reassigning some of the source nodes to hubs. See Figure 3 for an example.

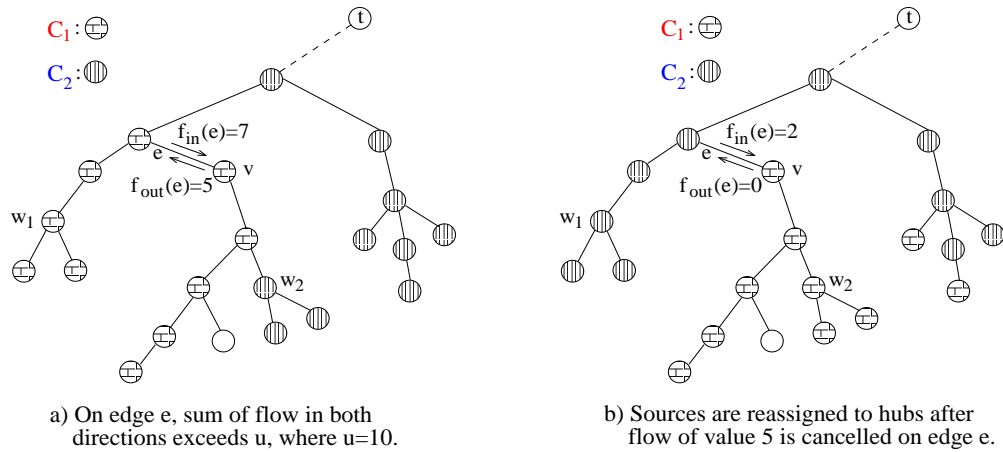


Fig. 3. An example of cancelling flow and reassigning sources to hub nodes. Here w_1 and w_2 are hub nodes chosen in the order of their indices.

Theorem 1. *There is a $(1 + \rho_{ST})$ -approximation algorithm for the single-sink capacity installation problem with a single cable type and uniform demand.*

Proof. Consider Algorithm Uniform. Let C_{OPT} be the cost of an optimal solution and C_{HEUR} be the cost of a solution output by the algorithm. Let C_{ST} denote the cost of the cables installed on the edges of the approximate Steiner tree

T . Let C_{DR} be the cost of cables installed on the direct edges added by the algorithm.

By Lemma 1, at most one copy of cable is sufficient to accomodate flow on the edges of the approximate Steiner tree T . The cost of a Steiner tree with terminal set $S \cup \{t\}$ is a lower bound on the optimal cost because we must connect the nodes in S to t and install at least one copy of the cable on each connecting edge. Therefore, $C_{ST} \leq \rho_{ST} C_{OPT}$.

For a source set C_k collected at iteration k , since $|C_k| = u$, the algorithm installs one copy of the cable on the shortest direct edge from the subtree T_v , which contains C_k , to t . The term $\sum_{s_i \in S} \frac{dem_i}{u} \cdot dist(s_i, t)$ is a lower bound on C_{OPT} , since dem_i must be routed a distance of at least $dist(s_i, t)$ and be charged at least at the rate $1/u$ per unit length. (In the uniform demand case, $dem_i = 1$ for all i .) Since source sets collected by the algorithm are disjoint, $\sum_k \sum_{s_i \in C_k} \frac{dem_i}{u} \cdot dist(s_i, t) = \sum_k \sum_{s_i \in C_k} \frac{dist(s_i, t)}{u}$ is a lower bound on C_{OPT} , as well. As demand of a set C_k is sent via the source in C_k that is closest to t (the hub node w_k), we get

$$dist(w_k, t) = \min_{s_i \in C_k} dist(s_i, t) \leq \frac{\sum_{s_i \in C_k} dist(s_i, t)}{\sum_{s_i \in C_k} 1} = \sum_{s_i \in C_k} \frac{dist(s_i, t)}{u}. \quad (1)$$

Thus, we finally have

$$C_{DR} = \sum_k dist(w_k, t) \leq \sum_k \sum_{s_i \in C_k} \frac{dist(s_i, t)}{u} \leq C_{OPT}. \quad (2)$$

Therefore, $C_{HEUR} = C_{ST} + C_{DR} \leq (1 + \rho_{ST})C_{OPT}$.

3 Non-uniform Demand

When source nodes have arbitrary demand, dem_i for source s_i , it is no longer possible to collect sources with total demand exactly equal to the capacity u . If we were allowed to split the (integral) demand for any source into single integral units each of which can be routed in separate paths to the sink, notice that the algorithm of the previous section can be used by expanding each source s_i to dem_i sources connected by zero-length edges in the tree. However, in the more general case, all the flow of a source must use the same path to the sink. In this case, we modify Algorithm Uniform so that we send demand directly to the sink when it accumulates to an amount between $u/2$ and u . To guarantee that we don't exceed u while collecting demand, we send all sources with demand at least $u/2$ directly at the beginning of the algorithm.

For a source set C , let $dem(C)$ be the total demand of sources in C . Recall that $D(C)$ is the total *remaining* (unprocessed) demand of C , as defined for the uniform demand case. The modified algorithm, which we call Algorithm Non-uniform, is as follows.

Algorithm Non-uniform:

Initialize: $R = S$.

Preprocessing: (send large demands directly)

For all sources s_i such that $dem_i \geq u/2$,

Route the demand on (s_i, t) .

Install $\lceil \frac{dem_i}{u} \rceil$ copies of cable on (s_i, t) .

Remove s_i from R .

Main step:

Pick a node v such that $D(T_v) \geq u/2$ and level of v is maximum.

If $v = t$, or $D(T_t) < u/2$, then go to the final step.

Pick a node, say w in $R \cap T_v$ such that $dist(w, t)$ is minimum, as a “hub” node.

Let $C = \{w\}$.

Collect source nodes in C (Details given below).

Add edge (w, t) to the network and install one copy of the cable on (w, t) .

Route demand of each source in C to the hub node via the unique path in T .

Route demand of C at the hub directly to the sink on (w, t) .

Remove C from R and set $C = \emptyset$.

If R is not empty, repeat the main step.

If R is empty, go to the final step.

Final step:

If $R \neq \emptyset$, then route all the demand in R to t via the unique paths in T .

Install one copy of cable on the edges of T which have positive flow.

Collect source nodes:

Add v to C , if $v \in R$.

Let v_1, \dots, v_k be the children of v .

If $w \neq v$, then

Let v_p be the child of v such that the hub node w is in T_{v_p} .

Add $T_{v_p} \cap R$ to C .

While $dem(C) < u/2$,

Pick an unprocessed child of v , say v_i .

Add $T_{v_i} \cap R$ to C .

Return C .

Lemma 2. *The algorithm routes demand such that:*

1) *flow on any edge of the tree T is at most u , and*

2) *flow on a direct edge added by the algorithm is at least $u/2$ and at most u .*

Proof. The proof is simpler compared to the uniform-demand case because the algorithm does not assign any subtree partially. Consider an edge e of T . Let v be incident on e such that e is not in T_v . Since all the sources in T_v are collected in the same set by the algorithm, demand of these sources is routed to a hub node either out of the subtree, or in the subtree, but not both. Thus, flow exists only in one direction. If the demand of sources is routed to a hub node out of

T_v , then outflow is at most $u - 1$. If the demand is routed to a hub node in the subtree, then inflow is at most $u - 1$. Thus, for any edge of T , flow does not exceed u .

Due to the subtree selection rule in the algorithm, if a subtree T_v is selected, then all the subtrees rooted at its children have remaining demand strictly less than $u/2$. Therefore, the first time $dem(C)$ exceeds $u/2$, it will be at most u so that total flow on the direct edges added by the algorithm is in the range $[u/2, u]$.

Theorem 2. *There is a $(2 + \rho_{ST})$ -approximation algorithm for the single-sink edge installation problem with a single cable type and non-uniform demand.*

Proof. We use the same definitions of C_{OPT} , C_{HEUR} , C_{DR} and C_{ST} as in the proof of Theorem 1.

By Lemma 2, at most one copy of the cable is sufficient to accommodate flow on the edges of the approximate Steiner tree T . Therefore, $C_{ST} \leq \rho_{ST} C_{OPT}$.

For a source set C_k collected at iteration k , the algorithm installs one copy of the cable on the shortest direct edge from the subtree T_v , which encloses C_k , to t . By Lemma 2, at most one copy of cable is sufficient to accommodate flow on the direct edges from hub nodes to t and $dem(C_k) \geq u/2$. The term $\sum_{s_i \in S} \frac{dem_i}{u} \cdot dist(s_i, t)$ is a lower bound on C_{OPT} as in the uniform demand case. Since source sets collected by the algorithm have disjoint sources and demand from a set C_k is sent via the source in C_k that is closest to t (the hub node w_k),

$$C_{OPT} \geq \sum_k \sum_{s_i \in C_k} \frac{dem_i}{u} dist(s_i, t) \geq \sum_k \sum_{s_i \in C_k} \frac{dem_i}{u} (\min_{s_i \in C_k} dist(s_i, t)). \quad (3)$$

Since $\sum_{s_i \in C_k} dem_i \geq \frac{u}{2}$ and $\min_{s_i \in C_k} dist(s_i, t) = dist(w_k, t)$, we have

$$C_{OPT} \geq \sum_k \sum_{s_i \in C_k} \frac{1}{2} dist(w_k, t) = \frac{1}{2} C_{DR}. \quad (4)$$

Therefore, $C_{HEUR} = C_{ST} + C_{DR} \leq (2 + \rho_{ST}) C_{OPT}$.

4 Extensions

Our methods apply to the following extension of the local access network design problem: Instead of specifying a single sink node, any node v in the graph can be used as a node that sinks u units of demand at a cost of f_v . A node is allowed to sink more than u units of demand by paying $\lceil \frac{dem}{u} \rceil \cdot f_v$ cost to sink dem units of flow. The problem is to open sufficient number of sinks and route all the demands to these sinks at minimum cable plus sink opening costs.

To model this extension, we extend the metric in two steps: 1) create a new sink node t with edges to every vertex v of cost f_v , 2) take the metric completion of this augmented network. Notice that the second step may decrease some of the costs on the edges incident on the new sink t (e.g., if $f_i + dist(j, i) < f_j$,

then the cost of the edge (j, t) can be reduced from f_j to $f_i + \text{dist}(j, i)$, or between any pair of original nodes (e.g., if $\text{dist}(i, j) > f_i + f_j$, then we may replace the former by the latter). Bearing this in mind, it is not hard to see that any solution in the new graph to the single cable problem with t as the sink and with the modified costs can be converted to a solution to the original problem of the same cost. Thus, our algorithms in the previous sections apply to give the same performance guarantees.

References

- [AG88] K. Altinkemer and B. Gavish, "Heuristics with constant error guarantees for the design of tree networks," *Management Science*, **34**, (1988) 331–341
- [AZ98] M. Andrews and L. Zhang, "The access network design problem," In *Proc. of the 39th Ann. IEEE Symp. on Foundations of Computer Science*, (1998) 42–49
- [CL83] K. M. Chandy and T. Lo, "The capacitated minimum tree," *Networks*, **3**, (1973) 173–182
- [KR98] Kawatra, R. and D. L. Bricker, "A multiperiod planning model for the capacitated minimal spanning tree problem", to appear in *European Journal of Operational Research* (1998)
- [KB83] A. Kershenbaum and R. Boorstyn, "Centralized teleprocessing network design," *Networks*, **13**, (1983) 279–293
- [KRY 93] S. Khuller, B. Raghavachari and N. E. Young, "Balancing minimum spanning and shortest path trees," *Algorithmica*, **14**, (1993) 305–322
- [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.
- [Pap78] C. H. Papadimitriou, "The complexity of the capacitated tree problem," *Networks*, **8**, (1978) 217–230
- [RZ00] G. Robins and A. Zelikovsky, "Improved steiner tree approximation in graphs", *Proc. of the 10th Ann. ACM-SIAM Symp. on Discrete Algorithms*, (2000) 770–779
- [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*, (1997) 619–628
- [S83] R. L. Sharma, "Design of an economical multidrop network topology with capacity constraints," *IEEE Trans. Comm.*, **31**, (1983) 590–591
- [SS99] B. Sanso and P. Soriano, Editors, "Telecommunications Network Planning," Kluwer Academic Publishers, 1999.