Buy-at-Bulk Network Design

Baruch Awerbuch *

Department of Computer Science
Johns Hopkins University
Baltimore, MD 21218

Yossi Azar †

Department of Computer Science
Tel Aviv University
Tel Aviv 69978, Israel

Abstract

The essence of the simplest buy-at-bulk network design problem is buying network capacity "wholesale" to guarantee connectivity from all network nodes to a certain central network switch. Capacity is sold with "volume discount": the more capacity is bought, the cheaper is the price per unit of bandwidth. We provide $O(\log^2 n)$ randomized approximation algorithm for the problem. This solves the open problem in [15]. The only previously known solutions were restricted to special cases (Euclidean graphs) [15].

We solve additional natural variations of the problem, such as multi-sink network design, as well as selective network design. These problems can be viewed as generalizations of the the Generalized Steiner Connectivity and Prize-collecting salesman (K-MST) problems.

In the selective network design problem, some subset of k wells must be connected to the (single) refinery, so that the total cost is minimized.

1 Introduction

1.1 The basic problem

Consider an oil company that wishes to connect a network of pipelines to carry oil from several remote well to a major refinery [15]. The company may use several types of pipes of various diameters, each has a different cost per unit distance which reflect an "economy of scale" ("volume discount"). For example diameter of 1 foot may cost \$500 per mile, a diameter of 3 feet may cost \$750 and a a diameter of 10 feet may cost \$1000 per mile. We consider general "volume discount" pricing function. (Formal definitions follow in section 2.) The goal is to design (off-line) a minimum cost network that would be sufficient to transport the oil to the refinery, assuming fixed oil supply at each well.

In case of *linear* pricing functions pipes would always go along the shortest path between the well and the refinery. More complex methods are necessary to capture "volume discount" pricing (with economy of scale). An interesting special case is the one of *step function* pricing, namely the case in which the only available pipes have capacity that exceeds the total demand, and thus single pipe is sufficient between each pair of nodes.

We consider the following basic variations of the problem:

- *Single-sink network design*: all the wells must be connected to the refinery.

  For the special case of step function pricing, this reduces to *Steiner Tree problem*, i.e., minimizing the cost of a tree spanning given set of points. We comment that this as well as following variations of the problems are NP-hard.

- *Multi-sink network design*: different wells are connected to different refineries (multiple wells may be connected to the same refinery).

*E-mail: baruch@blaze.cs.jhu.edu. Supported by Air Force Contract TNDGAFOSR-86-0078, ARPA/Army contract DABT63-93-C-0038, ARO contract DAAL03-86-K-0171, NSF contract 9114440-CCR, DARPA contract N00014-J-92-1799, and a special grant from IBM.

†E-Mail: azar@math.tau.ac.il. Research supported in part by Allon Fellowship and by the Israel Science Foundation administered by the Israel Academy of Sciences.

For the special case of step function pricing, this reduces the generalized network connectivity problem, solved in [9, 1]. The essence of this problem is minimizing the cost of a sub-graph in which given pairs of points are connected.

- *Selective network design*: some subset of k wells must be connected to the (single) refinery, so that the total cost is minimized.

  For the special case of step function pricing, this reduces to the *k-MST*, or *prize collecting salesman* problem (e.g., [5, 8, 10, 14, 3, 7].) The essence of this problem is selecting a subset of k points whose Steiner tree has the minimum cost.

Same problem comes up in others contexts, e.g., capacity planning for an Internet provider, that buys capacity (in bulk) from a phone company. The provider needs to buy enough capacity to guarantee enough bandwidth connectivity between communicating network nodes. The more capacity is bought, the less is the price per unit of bandwidth. A possible constraint is that the flow between a source and a sink is indivisible, e.g., routed along a single path.

In the context of Internet provider, additional complications may come up. One of them is that the requests for connections between sinks and sources come online, and the decisions on a route between source and sink, as well as decision on purchasing additional capacity, must be taken at the time the request is generated, and without knowledge of the future requests. This solution provided by online algorithm is then compared against optimal offline algorithm.

1.2 The results in this paper

The main contribution of the paper is providing polynomial-time algorithms that yield $O(\log^2 n)$ approximations for all of the above problems in general graphs and general pricing functions. Our algorithms are randomized and the approximation guarantees apply to the expectations over the coin flips of the algorithm.

There were no previous sub-linear approximations for any one of these problems. The problem of efficient single-sink network design was stated as the open problem in [15].

Our solutions for the single-sink and multi-sink problems also work in the *online* setting, with only constant additional performance degradation.

Our algorithms heavily rely on recent fundamental result of Bartal [6] concerning approximations of metric spaces.

1.3 Comparison with existing solutions

Salman, Cheriyan, Ravi and Subramanian [15] formulated single-sink network design problem and provided logarithmic ($O(\log D)$, where D is the total demand) approximation for general pricing functions for the special case of (Euclidean) graphs. The case of general pricing for the multi-sink and selective network design were not previously considered in the literature, to the best of our knowledge.

Another interesting special case is that of *offset linear* pricing function, which is the sum of a linear and a step function. This case, considered by Mansour and Peleg [13], captures the case in which there is only one type of cable and installing an edge has a fixed cost (similar to our model) as well as variable cost per unit flow. The solution in this case must be able to trade-off the weight of the tree with the distances. Mansour and Peleg [13] provide $O(\log n)$ approximation algorithm for the multi-sink network design, and [15] provides constant approximation ratio, for single-sink case.

2 Definitions and notations

We are given an undirected graph $G = (V, E)$, where $|V| = n$. Subsets of the nodes are specified as sources and sinks, each pair may have a demand of $dem_{u,v}$. The edges of G have lengths $l : E \to R^+$. Without loss of generality we may assume that for every pair of nodes u, v we can use the shortest-path distance $dist(u, v)$ as the length of the edge between u and v, i.e., take the metric completion of the given graph. All the traffic from u to v should follow one single path $P_{u,v}$ (i.e. it is indivisible). The edges of the network must be installed by purchasing zero or more copies from the set of cables where each cable type i has a specified capacity u_i and a specified cost c_i per unit of length. We make no assumption on the (non-negative) values u_i, c_i and the ratio c_i/u_i although it is natural to assume that if $u_i < u_j$ than $c_i < c_j$ (otherwise we get rid of type i) and that $c_j/u_j < c_i/u_i$ (which reflects the economy of scale) but this assumption is not really necessary for our algorithm.

We consider the multi-sink network design problem where the goal is to design minimum cost network that can simultaneously route all the demands. Here a solution can be characterized by

- specifying for each pair (u, v) of source and sink a path $P_{u,v}$
- specifying for each edge of the network induced by the paths the combinations of the cables used, zero or more copies of each cable may be installed on each edge. Clearly the total capacity of the cables installed on an edge must be at least the sum of the demands of the pairs that used this edge on their path.

An important special case is the single-sink network design problem (also called single-sink edge installation problem) in which all the paths ends at the specified sink. In that case we denote the supply of each source as dem_v. Here the paths of the solution are denoted by P_u.

We also consider the more complex problem of selective network design. In this case we consider, for simplicity a clean problem where the sink requires a total demand of k and each of (at least k sources) can supply a demand of 1. Thus we also need to chose which k sources to connect to the sink in addition to selecting the routes and the cables.

As claimed in [15] even for the single-sink network design the optimal choice of the routing depends on the choice of the cables, as they determine the cost of the edges. Yet, the optimal choice of cables on each edge depends on the flow on each edge, which is determined by the routing selection. Hence the optimal solution requires to select the routes and the cables simultaneously.

3 The algorithms and their analysis

Our approximation algorithms decouple between the two ingredients of selecting the routes and selecting the cables for each edge. Assume that we already decides on the routes. Thus we need to choose the cables for each edge separately with total capacity that covers the demand that flows through each edge. We call it the single-edge covering problem. Let us define the function $C(dem)$ as the minimum cost required to cover a total demand of dem for a unit distance. The problem of evaluation $C(dem)$ for some value dem is an integer min-knapsack problem known to be NP-hard (see [12]). It is well known that the knapsack problem has a polynomial approximation scheme. Similarly, one can derive a polynomial approximation scheme for the single-edge cover problem. In fact since we are providing only approximate solution it is easy to provide 2 approximation (linear time) algorithm for evaluation $C(dem)$ for each parameter dem which would be used for installing the cables once the routes have been chosen.

Observation 3.1 $\min_i c_i \lceil dem/u_i \rceil$ *is a 2 approximation for $C(dem)$ and can be computed in linear time.*

Proof: Clearly this is a feasible solution since we may use cable of type i $\lceil dem/u_i \rceil$ times. If optimal uses one cable we get the exact solution. If optimal uses more than one cable its value is at least $\min_{i|u_i<dem} c_i dem/u_i$. Our cost is at most twice as much since $\lceil dem/u_j \rceil < dem/u_j + 1 \leq 2dem/u_j$ where j is the index that the minimum is achieved. Clearly, the value can be computed in linear time. ∎

Observation 3.2 *The function $C(dem)$ is subadditive, i.e., $C(x + y) \leq C(x) + C(y)$*

Proof: The union of the optimal set of cables that cover a demand x with the optimal set of cables that cover a demand y is a cover for a demand $x + y$. ∎

In fact, our algorithm for choosing the routes is so robust that it does not depend on the exact function $C(dem)$ but only on the fact that it is sub-additive. In other words we provide one approximate solution for the routes that does not depend on the types of the cables. Of course, once we have chosen the routes we should install the cables as described above by a simple linear algorithm for each edge.

For choosing the routes we use the notion of probabilistic metric approximations [11, 2, 6]. We use the following notations following [6]. Let V a set of n points and M a metric space over V where the distance between u and v is denoted by $d_M(u,v)$.

Definition 3.1 *A metric space N over V, dominates a metric space M over V if for every $u,v \in V$, $d_N(u,v) \geq d_M(u,v)$.*

Definition 3.2 *A metric space N over V, α-approximates a a metric space M over V, if it dominates M and for every $u,v \in V$, $d_N(u,v) \leq \alpha d_M(u,v)$.*

We are interested with the following notion

Definition 3.3 *A set of metric spaces S over V, α-probabilistically-approximates a metric space a metric space M over V, if every metric space in S dominates M and there exists a probability distribution over the metric spaces $N \in S$ such that for every $u,v \in V$, $E(d_N(u,v)) \leq \alpha d_M(u,v)$.*

We use the following a simplified version of the main theorem of [6]:

Theorem 3.1 *Every weighted connected graph G can be α-probabilistically-approximated by a set of trees where $\alpha = O(\log^2 n)$ by polynomially computable probability distribution.*

We use the above metric approximations for our main theorem:

Theorem 3.2 *Consider a graph G and the set of trees which α-probabilistically-approximate it. Then there is a feasible solution to all network design problems on the set of trees whose expected cost (over the distribution on the trees) is at most α times the optimal cost in G.*

Proof: The optimal solution uses a path $Q_{u,v}$ in G to route the demand between u and v. Let f_e be the demand that flows through an edge e, i.e

$$f_e = \sum_{(u,v)|e \in Q_{u,v}} dem_{u,v}$$

Clearly the value of the optimal solution is

$$\sum_{e \in E(G)} l(e) C(f_e).$$

Consider some edge $e = (x,y)$ in the graph G. We associate with this edge in each tree $T \in S$ a path T_e between x and y of length $d_T(x,y) = l(e)\alpha_T$ where $E(\alpha_T) = O(\alpha)$. The cost of designing a path T_e with a flow of f_e on each edge $e' \in T_e$ satisfies

$$\sum_{e' \in T_e} l(e') C(f_e) = d_T(x,y) C(f_e) = l(e)\alpha_T C(f_e).$$

If we design a network in T for all paths T_e for all $e \in E(G)$ each with demand f_e then the flow $f'_{e'}$ on each edge $e' \in E(T)$ is

$$f'_{e'} = \sum_{e \in E(G)|e' \in T_e} f_e$$

and its cost is (by the fact that the function C is sub-additive)

$$\begin{aligned}
\sum_{e' \in E(T)} & l(e') \quad C(f'_{e'}) \\
&= \sum_{e' \in E(T)} l(e') C\Big(\sum_{e \in E(G)|e' \in T_e} f_e \Big) \\
&\leq \sum_{e' \in E(T)} l(e') \sum_{e \in E(G)|e' \in T_e} C(f_e) \\
&= \sum_{e' \in E(T)} \sum_{e \in E(G)|e' \in T_e} l(e') C(f_e) \\
&= \sum_{e \in E(G)} \sum_{e' \in T_e} l(e') C(f_e) \\
&= \sum_{e \in E(G)} l(e) \alpha_T C(f_e) \\
&= \alpha_T \sum_{e \in E(G)} l(e) C(f_e)
\end{aligned}$$

This implies that the expected cost over all trees is at most α times the optimal cost. A path $Q_{u,v}$

in G is associated with a (maybe non-simple) path $Q_{T,(u,v)}$ in T which consists of concatenating the paths in T associated with the edges of $Q_{u,v}$ and

$$\begin{aligned} f'_{e'} &= \sum_{e \in E(G) | e' \in T_e} f_e \\ &= \sum_{e \in E(G) | e' \in T_e} \sum_{(u,v) | e \in Q_{u,v}} dem_{u,v} \\ &= \sum_{(u,v) | e' \in T_e, e \in Q_{u,v}} dem_{u,v} \\ &= \sum_{(u,v) | e' \in Q_{T,(u,v)}} dem_{u,v} \end{aligned}$$

which implies that the network that we design for each T is feasible solution for the demand. ∎

We define the following algorithm for the multi-sink (and thus single-sink) network design problem. Choose at random tree $T \in S$ from the set which α-probabilistically-approximate G according to the probability distribution. Then the route $P_{u,v}$ is the route on the tree. Once the routed have been selected we choose the cables on each edge as described in the single-edge cover problem.

Theorem 3.3 *The randomized algorithm above α-approximates the multi-sink network design (and thus for the single-sink network design).*

Proof: Since our algorithm is optimal for each tree than by using theorem 3.2 the cost is as required. ∎

Theorem 3.4 *There is a randomized algorithm which α-approximates the selective network design problem.*

Proof: Similar to the previous theorem, it is enough to design an algorithm for a tree. We choose the sink s to be the root of the tree. We first describe the solution for binary trees. Then, we easily show how to transform every tree to binary tree. At each node u we compute a table A_u of the minimum cost of connecting to u exactly i sources $0 \leq i \leq k$ in the subtree of u. The desired value is $A_s(k)$ (it will be easy to see that not only the value of the optimal solution can be found but the solution itself). Clearly $A_u(0) = 0$ for all u. The computation is done from the leaves to the root. We start with the leaves. If it is a source than $A_u(1) = 1$ and $A_u(i) = \infty$ for $i > 1$. If it is not a source $A_u(i) = \infty$ for $i \geq 1$. Assume that we computed the table for the two children x and y of a node u than we can compute A_u as follows: If u is not a source node than its value of entry i is

$$\begin{aligned} A_u(i) &= \min_{0 \leq j \leq i} A_u(j) + l(u,x)C(j) \\ &+ A_u(i-j) + l(u,y)C(i-j) \end{aligned}$$

which is considering all choices of j source in the subtree of x and $i - j$ in the subtree of y plus the cost of the edges from x and y to u. If u is a source node than its value of entry i is

$$\begin{aligned} A_u(i) &= \min_{0 \leq j \leq i-1} A_u(j) + l(u,x)C(j) \\ &+ A_u(i-j-1) + l(u,y)C(i-j-1) \end{aligned}$$

since we would always use u as a source. The correctness of the table follows from its definition. To complete the proof we need to deal with non-binary trees. If a node has only one child than add a non-source leaf as a second child. If a node has $d > 2$ children then we replace it with a binary tree with $d - 1$ internal binary nodes where the distances in that tree are zeros, the root is u the d leaves are the children of u. All internal nodes (except maybe u) are not sources. Clearly, this transformation preserves the distances in the tree and at most double the size of the tree. ∎

4 Online algorithms for single-sink and multi-sink problems

Surprisingly, the algorithms for the multi-sink (and thus single-sink network design) are almost on-line algorithms.

Theorem 4.1 *There in an on-line randomized algorithm for the multi-sink network design (and thus for the single-sink network design) whose competitive ratio is $O(\alpha)$.*

Proof: Since our off-line algorithm chooses a tree and then route all path on the tree than choosing the routes is done on-line. The only thing that we have to show that we can chose which cable to install in an on-line fashion. In fact we show that for each edge separately, we have an on-line algorithm which is constant competitive for installing the cables on one edge, i.e., for the bulk covering problem. Therefore the overall algorithm remains $O(\alpha)$ competitive.

The algorithm for bulk covering problem is simple and similar to the algorithm of [4]. When a new cable is needed to cover the demand the algorithm buys a cable with the minimum c_i/u_i among the types i such that c_i is at most twice the total cost of the cables spent on that edge. The first cable of an edge is the cheapest cable. It can be shown that this is a constant competitive strategy. ∎

References

[1] A. Agrawal, P. Klein, and R. Ravi. When trees collide: An approximation algorithm for the generalized Steiner problem in networks. In *Proceedings of the 23rd ACM Symposium on Theory of Computing*, pages 134–144, 1991.

[2] Noga Alon, Richard M. Karp, David Peleg, and Douglas West. A graph-theoretic game and its applications to the k-server problem. *SIAM J. on Computing*, 24:78–100, 1995.

[3] B. Awerbuch, Y. Azar, A. Blum, and S. Vempala. Improved approximation guarantees for minimum-weight k-trees and prize-collecting salesmen. In *Proc. 27th ACM Symp. on Theory of Computing*, pages 277–283, 1995.

[4] Y. Azar, Y. Bartal, E. Feuerstein, A. Fiat, S. Leonardi, and A. Rosén. On capital investment. In *Proceedings of ICALP 96*, 1996.

[5] E. Balas. The prize collecting traveling salesman problem. *Networks*, 19:621–636, 1989.

[6] Y. Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *Proc. 37th IEEE Symp. on Found. of Comp. Science*, 1997.

[7] A. Blum, R. Ravi, and S. Vempala. A constant-factor approximation algorithm for the k-mst problem. In *Proc. 28th ACM Symp. on Theory of Computing*, pages 442–448, 1996.

[8] N. Garg and D. Hochbaum. An $o(\log k)$ approximation algorithm for the k minimum spanning tree problem in the plane. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 432–438, 1994.

[9] M. Goemans and D. Williamson. General approximation technique for constrained forest problems. In *Proceedings of the 3rd Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 307–315, 1992.

[10] B.L. Golden, L. Levy, and R. Vohra. The orienteering problem. *Naval Research Logistics*, 34:307–318, 1987.

[11] R.M. Karp. A 2k- competitive algorithm for the circle. Manuscript, 1989.

[12] G.S. Lueker. Two np-complete problems in non-negative integer programming. Technical report, Princeton University, Princeton, NJ, CS-178, 1975.

[13] Y. Mansour and D. Peleg. An approximation algorithm for minimum- cost network design. Technical report, The Weizman Institute of Science, Rehovot, Israel, CS94-22, 1994.

[14] R. Ravi, R. Sundaram, M.V. Marathe, D.J. Rosenkrantz, and S.S. Ravi. Spanning trees short and small. In *Proceedings of the 5th Annual ACM-SIAM Symposium on Discrete Algorithms*, 1994.

[15] F.S. Salman, J. Cheriyan, R. Ravi, and S .Subramanian. Buy-at-bulk network design:approximating the single-sink edge installation problem. In *Proc. 8th ACM-SIAM Symp. on Discrete Algorithms*, 1997.