A Deterministic Algorithm for the COST-DISTANCE Problem

Chandra Chekuri *　　Sanjeev Khanna †　　Joseph (Seffi) Naor‡

1 Introduction

We study the COST-DISTANCE network design problem, recently defined by Meyerson, Munagala, and Plotkin [4], as a common generalization of several useful problems in network design and facility location (see [4]). In this problem, we are given an undirected graph $G = (V, E)$ with a designated *root* vertex r and a set of terminals S. We are also given two non-negative real valued functions defined on E, namely, a cost function c and a length function ℓ, and a non-negative weight function w on the set S. The goal is to construct a tree T that connects the terminals in S to the root r and minimizes $\sum_{e\in T} c(e) + \sum_{t\in S} w(t)\ell_T(r,t)$, where $\ell_T(r,t)$ is the length of the path in T from t to r. This problem can also be seen as a Lagrangian relaxation of the budgeted versions of network design problems studied earlier in [3, 1, 2]. Building on the basic framework developed in [3], Meyerson *et al* [4] give a *randomized* approximation algorithm with an expected ratio of $O(\log k)$, where $k = |S|$. An interesting question raised in [4] is whether this ratio can be achieved deterministically. In this paper we answer this question in the affirmative.

Matching Graph: Define a complete graph H on $S \cup \{r\}$ with a weight function b on its edges; $b(u, v)$ is assigned the weight of the shortest path in G connecting u and v with respect to the metric $m(e) = c(e) + \min\{w(u), w(v)\}\ell(e)$. We assume that $w(r) = \infty$ and $w(v) = 0$ for any $v \notin S \cup \{r\}$.

The algorithm of [4] is based on finding a minimum weight (near) perfect matching in H, and proceeds as follows: (a) Compute a minimum weight (near) perfect matching M in H. (b) For all $(u, v) \in M$, u is chosen to be a center with probability $p = w(u)/(w(u) + w(v))$, otherwise v becomes the center (with probability $1 - p$). The weight of the center is set to $w(u) + w(v)$ and the non-center vertex is removed from the terminal set. (c) Continue recursively, where the set of terminals is restricted to be the set of centers computed in the previous step. Connect the non-center vertices from the previous step to the root via their respective centers. [4] shows that the expected cost of each step is bounded by the value of the optimal solution and therefore the expected value of the approximation factor of their algorithm is $O(\log k)$. We show how to derandomize this algorithm.

2 Derandomization

For any terminal t, let $L^*_t = \ell_{T^*}(r, t)$ denote the distance from the root r to t in some optimal tree T^*. The *distance cost* of a terminal t in a given tree T is simply $w_t \cdot \ell_T(r, t)$. Then, for any edge (u, v) in the matching M, it is shown in [4] that the expected distance cost of the new center is $w(u)L^*_u + w(v)L^*_v$ which is exactly the sum of the distance costs of the two terminals u and v before merging. Thus, the expected distance cost of the new instance obtained after the merging is unchanged. Further, the random process works without any knowledge of L^*_u or L^*_v. Derandomizing the algorithm using the standard method of conditional probabilities requires pessimistic estimators for the quantities L^*_u and L^*_v. However, the algorithm in [4] does not seem to provide them.

In this note we describe an LP-based approach that obtains a deterministic $O(\log k)$-approximation algorithm for the COST-DISTANCE problem. We obtain a pessimistic estimator for L^*_t via the LP relaxation. As a byproduct we also show that the integrality gap of the LP is $O(\log k)$. Only a constant factor hardness is known for the COST-DISTANCE problem, and the worst LP integrality gap known is also $O(1)$. Our result thus raises the question of whether the LP integrality gap is $o(\log k)$; an affirmative answer would give improved approximation for the COST-DISTANCE problem as well as several other network design problems.

LP Formulation: We will view G as a directed graph (replace each undirected edge by two anti-symmetric arcs) and the solution as a tree rooted at r with directed paths to terminals in S. For all $t \in S$, $\mathcal{P}_t$ denotes the set of directed paths from the root r to t, and for each $e \in E$, a variable $x(e)$ indicates its capacity. Note that $\mathcal{P}_t \cap \mathcal{P}_{t'} = \emptyset$ for $t \neq t'$.

$$(P)\ \min \sum_{e\in E} c(e)x(e) + \sum_{t\in S} w(t) \sum_{p\in\mathcal{P}_t} \ell(p)f(p)$$

*Bell Labs, 600 Mountain Ave, Murray Hill, NJ 07974. E-mail: chekuri@research.bell-labs.com.

†Dept. of Computer & Information Science, University of Pennsylvania. E-mail: sanjeev@cis.upenn.edu. Supported in part by an Alfred P. Sloan Research Fellowship.

‡Computer Science Dept., Technion, Haifa 32000, Israel. Most of this work was done while the author was at Bell Labs, 600 Mountain Ave, Murray Hill, NJ 07974. E-mail: naor@cs.technion.ac.il.

subject to:

$$\begin{array}{rcll} \sum_{p\in\mathcal{P}_t \mid e\in p} f(p) & \leq & x(e) & \forall\, e\in E,\, t\in S \\ \sum_{p\in\mathcal{P}_t} f(p) & \geq & 1 & \forall\, t\in S \\ f(p), x(e) & \geq & 0 & \forall\, e\in E,\, t\in S,\, p\in\mathcal{P}_t \end{array}$$

The LP assigns fractional capacities to edges such that one unit of flow can be shipped from the root r to each terminal $t \in S$. The dual program is as follows.

$$(D)\ \max\ \sum_{t\in S}\alpha(t)$$

subject to:

$$\begin{array}{rcll} \sum_{t\in S}\beta^t(e) & \leq & c(e) & \forall\, e\in E \\ \alpha(t) - \sum_{e\in p}\beta^t(e) & \leq & w(t)\cdot\ell(p) & \forall\, t\in S,\, p\in\mathcal{P}_t \\ \alpha(t), \beta^t(e) & \geq & 0 & \forall\, e\in E,\, t\in S \end{array}$$

LEMMA 2.1. *The graph H contains a matching of size at least $|S|/3$ such that its weight is no more than the optimal cost for the linear program (P) on G.*

Proof. Consider the following LP (MP) for the minimum weight perfect matching problem in H, and its dual (MD).

(MP)	(MD)
$\min \sum_{(u,v)} b(u,v)x(u,v)$	$\max \sum_u y(u)$
$\sum_v x(u,v) = 1$	$y(u)+y(v) \leq b(u,v)$
$x(u,v) \geq 0$	$y(u) \geq 0$

Notice that we do not have the *odd-set constraints* that are necessary to obtain an optimal matching. However, it is well known that a basic feasible solution to (MP) is half-integral and that the half-integral edges induce odd cycles. Consider an odd half-integral cycle of length $i \geq 3$. It is easy to see that we can recover at least $\lfloor i/2 \rfloor$ edges of weight no more than that of the fractional cycle weight. Hence we can obtain a matching M of size at least $\lceil k/3 \rceil$ with total weight upper bounded by the optimum of (MP).

Let x^* and y^* be optimum solutions to (MP) and (MD) respectively. It follows that $y^*(u) + y^*(v) \leq b(u,v)$ holds and by duality $\sum_{u\in H} y^*(u)$ is at least the weight of M.

We claim that there is a feasible solution to (D) of value at least $\sum_u y^*(u)$. To prove this, set $\alpha(t) = y^*(t)$ for all terminals t that are matched vertices in M, and set $\alpha(t) = 0$ otherwise. In the graph G, define the ball B_t centered at t of radius $\alpha(t)$ where distance is measured with respect to the metric $c(e) + w(t)\ell(e)$. The main observation is that the balls $\{B_t\}_{t\in S}$ are disjoint (they do not overlap). This follows from the fact that $y^*(u) + y^*(v) \leq b(u,v)$ for all matched edges, where $b(u,v)$ is the distance between u and v in the metric $c(e) + \min\{w(u), w(v)\}\ell(e)$. For each edge e, let $g^t(e)$ be the fraction of the edge contained in B_t. We set $\beta^t(e) = g^t(e)\cdot c(e)$. It can be easily verified that these settings are feasible for (D). Hence, the optimal value of (D) (and (P)) on G is at least the weight of the matching M obtained from (MP) as above. □

Once we have the matching M as above, suppose we choose the center for each matched pair of terminals as in step (b) of the algorithm in [4] with randomization. After merging the terminals, we can reuse the current LP solution as a solution for the new instance (only the weights on terminals are altered, yet the edge and flow path capacities can remain the same), and repeat this process. Let O and O' denote the LP cost on the original and the new instances respectively. Let C_M denote the expected cost of routing the demand of the non-center vertices to their respective centers.

LEMMA 2.2. *$C_M \leq 2O$ and $O' \leq O$ and hence $C_M + \alpha\log(2k/3)O' \leq \alpha\log(k)O$ for some fixed constant α.*

Thus, by a simple inductive argument it follows that the expected cost of our final solution is $O(\log k)$ times the optimal LP cost. For a given matching M and a deterministic choice of centers A, let $C_M(A)$ and $O'(A)$ be the merging cost and the cost of the new instance respectively. To obtain a deterministic algorithm, we use the method of conditional probabilities to find a set A such that $C_M(A) + \alpha\log(2k/3)O'(A)$ is bounded by the expected value of the random choice, which is at most $\alpha\log(k)O$ (Lemma 2.2) . For this purpose we use the *LP-distance* for any terminal t, $\gamma_t = \sum_{p\in\mathcal{P}_t}\ell(p)\cdot f^*(p)$, as a pessimistic estimator for L^*_t. We omit further details here.

THEOREM 2.1. *There exists a* deterministic $O(\log k)$-*approximation algorithm for the* COST-DISTANCE *problem.*

References

[1] G. Kortsarz and D. Peleg, Approximating shallow-light trees, In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 103-110, 1997.

[2] S. Khuller, B. Raghavachari, and N. Young, Balancing minimum spanning trees and shortest path trees, *Algorithmica*, 14:305–321, 1995.

[3] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, Bicriteria network design problems. *J. of Algorithms*, 28:142–171, 1998.

[4] A. Meyerson, K. Munagala, and S. Plotkin. Cost-Distance: Two Metric Network Design. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, 2000.