Stochastic Steiner Tree with Non-Uniform Inflation

Anupam Gupta[1], MohammadTaghi Hajiaghayi[2], and Amit Kumar[3]

[1] Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213.
Supported in part by an NSF CAREER award CCF-0448095, and by an Alfred P. Sloan Fellowship.
[2] Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213.
Supported in part by NSF ITR grant CCR-0122581 (The Aladdin Center)
[3] Department of Computer Science and Engineering, Indian Institute of Technology, New Delhi, India – 110016. Part of this work was done while the author was at Max-Planck-Institut für Informatik, Saarbrücken, Germany.

Abstract. We study the Steiner Tree problem in the model of two-stage stochastic optimization with *non-uniform inflation factors*, and give a poly-logarithmic approximation factor for this problem. In this problem, we are given a graph $G = (V, E)$, with each edge having two costs c_M and c_T (the costs for Monday and Tuesday, respectively). We are also given a probability distribution $\pi : 2^V \to [0, 1]$ over subsets of V, and will be given a *client set* S drawn from this distribution on Tuesday. The algorithm has to buy a set of edges E_M on Monday, and after the client set S is revealed on Tuesday, it has to buy a (possibly empty) set of edges $E_T(S)$ so that the edges in $E_M \cup E_T(S)$ connect all the nodes in S. The goal is to minimize the $c_M(E_M) + \mathbf{E}_{S \leftarrow \pi}[\, c_T(\, E_T(S)\,)\,]$.

We give the first poly-logarithmic approximation algorithm for this problem. Our algorithm builds on the recent techniques developed by Chekuri et al. (FOCS 2006) for multi-commodity Cost-Distance. Previously, the problem had been studied for the cases when $c_T = \sigma \times c_M$ for some constant $\sigma \geq 1$ (i.e., the *uniform* case), or for the case when the goal was to find a tree spanning *all the vertices* but Tuesday's costs were drawn from a given distribution $\widehat{\pi}$ (the so-called "stochastic MST case").

We complement our results by showing that our problem is at least as hard as the single-sink Cost-Distance problem (which is known to be $\Omega(\log \log n)$ hard). Moreover, the requirement that Tuesday's costs are fixed seems essential: if we allow Tuesday's costs to dependent on the scenario as in stochastic MST, the problem becomes as hard as Label Cover (which is $\Omega(2^{\log^{1-\varepsilon} n})$-hard). As an aside, we also give an LP-rounding algorithm for the multi-commodity Cost-Distance problem, matching the $O(\log^4 n)$ approximation guarantee given by Chekuri et al. (FOCS 2006).

1 Introduction

This paper studies the Steiner tree problem in the framework of two-stage stochastic approximation, which is perhaps best (albeit a bit informally) de-

scribed as follows. *On Monday, we are given a graph with two cost functions c_M and c_T on the edges, and a distribution π predicting future demands; we can build some edges E_M at cost c_M. On Tuesday, the actual demand set S arrives (drawn from the distribution π), and we must complete a Steiner tree on the set S, but any edges E_T bought on Tuesday cost c_T. How can we minimize our expected cost*

$$c_M(E_M) + \mathbf{E}_{S \leftarrow \pi}[\, c_T(\, E_T(S)\,)\,] \quad ?$$

The Stochastic Steiner tree problem has been studied before in the special case when Tuesday's cost function c_T is a scaled-up version of Monday's costs c_M (i.e., there is an constant *inflation factor* $\sigma > 1$ such that $c_T(e) = \sigma \times c_M(e)$); for this case, constant-factor approximations are known [9, 10, 12]. While these results can be generalized in some directions (see Section 1.1 for a detailed discussion), it has been an open question whether we could handle the case when the two costs c_M and c_T are unrelated. (We will refer to this case as the *non-uniform inflation* case, as opposed to the uniform inflation case when the costs c_M and c_T are just scaled versions of each other.)

This gap in our understanding was made more apparent by the fact that many other problems such as Facility Location, Vertex Cover and Set Cover, were all shown to admit good approximations in the non-uniform inflation model [22,24]: in fact, the results for these problems could be obtained even when the edge cost could depend on the day *as well as on the demand set appearing on Tuesday.*

Theorem 1 (Main Theorem). *There is an $O(\log^2(\min(N, \lambda)) \log^4 n \log\log n)$-approximation algorithm for the two-stage stochastic Steiner tree problem with non-uniform inflation costs with N scenarios, on a graph with n nodes. Here $\lambda = \max_{e \in E} c_T(e)/c_M(e)$, i.e., the maximum inflation over all edges.*

This is the first non-trivial approximation algorithm for this problem. Note that the cost of an edge can either increase or decrease on Tuesday; however, we would like to emphasize that our result holds only when Tuesday's costs c_T do not depend on the materialized demand set S. (Read on for a justification of this requirement.)

We also show that the two-stage stochastic Steiner tree problem is at least as hard as the single-source cost-distance problem.

Theorem 2 (Hardness). *The two-stage stochastic Steiner tree problem is at least $\Omega(\log\log n)$-hard unless $NP \subseteq DTIME(n^{\log\log\log n})$.*

The hardness result in the above theorem holds even for the special case of Stochastic Steiner tree when the cost of some edges remain the same between days, and the cost of the remaining edges increases on Tuesday by some universal factor.

Finally, we justify the requirement that Tuesday's costs c_T are fixed by showing that the problem becomes very hard without this requirement. Indeed, we can show the following theorem whose proof is deferred to the journal paper.

Theorem 3. *The two-stage stochastic Steiner tree problem when Tuesday's costs are dependent on the materialized demand is at least $\Omega(2^{\log^{1-\varepsilon} n})$ hard for every fixed $\varepsilon > 0$.*

Finally, we also give an LP-rounding algorithm for the multi-commodity Cost-Distance problem, matching the $O(\log^4 n)$ approximation guarantee given by Chekuri et al. [4]; however, we note that the LP we consider is not the standard LP for the problem.

Our Techniques. Our approach will be to reduce our problem to a more general problem which we call Group-Cost-Distance:

Definition 4 (Group-Cost-Distance) *Consider a (multi)graph $G = (V, E)$ with each edge having a* buying cost b_e *and a* renting cost c_e*. Given a set of subsets $S_1, S_2, \ldots, S_N \subseteq V$, find for each i a tree T_i that spans S_i, so as to minimize the total cost*

$$\sum_{e \in \cup_i T_i} b_e \; + \; \sum_{i=1}^{N} \sum_{e \in T_i} c_e. \tag{1.1}$$

Defining $\mathbb{F} = \cup_i T_i$ and $x_e =$ number of trees using edge e, we want to minimize $\sum_{e \in \mathbb{F}} (b_e + x_e \, c_e)$.

The problem can also be called *"multicast" cost-distance*, since we are trying to find multicast trees on each group that give the least cost given the concave cost functions on each edge. Note that when each $S_i = \{s_i, t_i\}$, then we get the (Multicommodity) Cost-Distance problem, for which the first poly-logarithmic approximation algorithms were given only recently [4]; in fact, we build on the techniques used to solve that problem to give the approximation algorithm for the Group-Cost-Distance problem.

1.1 Related Work

Stochastic Problems. For the Stochastic Steiner tree problem in the *uniform inflation* case where all the edge-costs increase on Tuesday by the same amount σ, an $O(\log n)$-approximation was given by Immorlica et al. [16], and constant-factor approximations were given by [9, 10, 12]. These results were extended to handle the case when the inflation factors could be random variables, and hence the probability distribution would be over tuples of the form (demand set S, inflation factor σ) [11, 15].

A related result is known for the Stochastic Minimum Spanning Tree problem, where one has to connect *all* the vertices of the graph. In this case, we are

given Monday's costs c_M, and the probability distribution is over possible Tuesday costs c_T. For this problem, Dhamdhere et al. [7] gave an $O(\log n + \log N)$ approximation, where N is the number of scenarios. They solve an LP and randomly round the solution; however, their random rounding seems to crucially require that all nodes need to be connected up, and the idea does not seem to extend to the *Steiner* case. (Note that their problem is incomparable to ours: in this paper, we assume that Monday's and Tuesday's costs were deterministic whereas they do not; on the other hand, in our problem, we get a random set of terminals on Tuesday, whereas they have to connect *all* the vertices which makes their task easier.)

Approximation algorithms for several other problems have been given in the non-uniform stochastic setting; see [22,24]. For a general overview of some techniques used in stochastic optimization, see, e.g., [10, 24]. However, nothing has been known for the Stochastic Steiner tree problem with non-uniform inflation costs.

In many instances of the stochastic optimization problem, it is possible that the number of possible scenarios on Tuesday (i.e., the support of the distribution π) is exponentially large. Charikar et al. [2] gave a useful technique by which we could reduce the problem to a a much smaller number of scenarios (polynomial in the problem size and inflation factors) by random sampling. We shall use this tool in our algorithm as well.

Buy-at-Bulk and Cost-Distance Problems. There has been a huge body of work on so-called *buy-at-bulk* problems which model natural economies-of-scale in allocating bandwidth; see, e.g., [3] and the references therein. The (single-source) Cost-Distance problem was defined by Meyerson, Munagala and Plotkin [20]: this is the case of Group-Cost-Distance with a root $r \in V$, and each $S_i = \{t_i, r\}$. They gave a randomized $O(\log k)$-approximation algorithm where $k = |\cup_i S_i|$, which was later derandomized by Chekuri, Khanna and Naor [5]. (An online poly-logarithmic competitive algorithm was given by Meyerson [19].) These results use a randomized pairing technique that keeps the expected demand at each node constant; this idea does not seem to extend to Group-Cost-Distance. The Multicommodity Cost-Distance problem (i.e., with arbitrary source-sink pairs) was studied by Charikar and Karagiozova [3] who gave an $\exp\{\sqrt{\log n \log\log n}\}$-approximation algorithm. Very recently, this was improved to poly-logarithmic approximation ratio by Chekuri, Hajiaghayi, Kortsarz, and Salavatipour [4] (see also [13,14]). We will draw on several ideas from these results.

Embedding Graph Metrics into Subtrees. Improving a result of Alon et al. [1], Elkin et al. [8] recently showed the following theorem that every graph metric can be approximated by a distribution over its subtrees with a distortion of $O(\log^2 n \log\log n)$.

Theorem 5 (Subtree Embedding Theorem). *Given a graph $G = (V, E)$, there exists a probability distribution $\mathcal{D}_G$ over spanning trees of G such that for*

every $x, y \in V(G)$, the expected distance $\mathbf{E}_{T \leftarrow \mathcal{D}_G}[d_T(x,y)] \leq \beta_{EEST}\, d_G(x,y)$ for $\beta_{EEST} = O(\log^2 n \log\log n)$.

Note that spanning trees T trivially ensure that $d_G \leq d_T$. The parameter β_{EEST} will appear in all of our approximation guarantees.

2 Reduction to Group Cost-Distance

Note that the distribution π may be given as a black-box, and may be very complicated. However, using a theorem of Charikar, Chekuri, and Pál on using sample averages [2, Theorem 2], we can focus our attention on the case when the probability distribution π is the *uniform* distribution over some N sets $S_1, S_2, \ldots, S_N \subseteq V$, and hence the goal is to compute edge sets $E_0 \doteq E_M$, and $E_1, E_2, \ldots, E_N$ (one for each scenario) such that $E_0 \cup E_i$ contains a Steiner tree on S_i. Scaling the objective function by a factor of N, we now want to minimize

$$N \cdot c_M(E_0) + \sum_{i=1}^{N} c_T(E_i) \tag{2.2}$$

Basically, the N sets will just be N independent draws from the distribution π. We set the value $N = \Theta(\lambda^2 \epsilon^{-5} m)$, where λ is a parameter that measures the "relative cost of information" and can be set to $\max_e c_T(e)/c_M(e)$ for the purposes of this paper, m is the number of edges in G, and ϵ is a suitably small constant. Let ρ_{ST} be the best known approximation ratio for the Steiner tree problem [23]. The following reduction can be inferred from [2, Theorem 3] (see also [25]):

Lemma 1 (Scenario Reduction). *Given an α-approximation algorithm for the above instance of the stochastic Steiner tree problem with N scenarios, run it independently $\Theta(1/\epsilon)$ times and take the best solution. With constant probability, this gives an $O((1+\epsilon)\alpha)$-approximation to the original stochastic Steiner tree problem on the distribution π.*

Before we go on, note that E_0 and each of the $E_0 \cup E_i$ are acyclic in an optimal solution. We now give the reduction to Group-Cost-Distance. Create a new (multi)graph, whose vertex set is still V. For each edge $e \in E$ in the original graph, we add two parallel edges e_1 (with *buying cost* $b_{e_1} = N \cdot c_M(e)$ and *renting cost* $c_{e_1} = 0$) and e_2 (with *buying cost* $b_{e_2} = 0$ and *renting cost* $c_{e_2} = c_T(e)$). The goal is to find a set of N trees $T_1, \ldots, T_N$, with T_i spanning the set S_i, so as to minimize

$$\textstyle\sum_{e \in \cup_i T_i} b_e \;+\; \sum_{i=1}^{N} c(T_i). \tag{2.3}$$

It is easily verified that the optimal solution to the two objective functions (2.2) and (2.3) are the same when we define the buying and renting costs as described above. Using Lemma 1, we get the following reduction.

Lemma 2. *An α-approximation for the Group-Cost-Distance problem implies an $O(\alpha)$-approximation for the non-uniform Stochastic Steiner tree problem.*

(As an aside, if the distribution π consists of N scenarios listed explicitly, we can do an identical reduction to Group-Cost-Distance, but now the value of N need not have any relationship to λ.)

3 Observations and Reductions

Recall that a solution to the Group-Cost-Distance problem is a collection of trees T_i spanning S_i; their union is $\mathbb{F} = \cup_i T_i$, and x_e is the number of trees that use edge e. Note that if we were just given the set $\mathbb{F} \subseteq E$ of edges, we could use the ρ_{ST}-approximation algorithm for finding the minimum cost Steiner tree to find trees T_i' such that $c(T_i') \leq \rho_{ST}\, c(T_i)$ for any tree $T_i \subseteq \mathbb{F}$ spanning S_i. Let us define

$$\mathsf{cost}(\mathbb{F}) \doteq b(\mathbb{F}) + \textstyle\sum_i c(T_i); \tag{3.4}$$

where $T_i \subseteq \mathbb{F}$ is the minimum cost Steiner tree spanning S_i. We use $\mathsf{OPT} = \mathbb{F}^*$ to denote the set of edges used in an optimal solution for the Group-Cost-Distance instance, and hence $\mathsf{cost}(\mathsf{OPT})$ is the total optimal cost. Henceforth, we may specify a solution to an instance of the Group-Cost-Distance problem by just specifying the set of edges $\mathbb{F} = \cup_i T_i$, where T_i is the tree spanning S_i in this solution.

As an aside, note that $\mathsf{cost}(\mathbb{F})$ is the optimal cost of *any* solution using the edges from $\mathbb{F} \subseteq E$; computing $\mathsf{cost}(\mathbb{F})$ is hard given the set $\mathbb{F}$, but that is not a problem since it will be used only as an accounting tool. Of course, given $\mathbb{F}$, we can build a solution to the Group-Cost-Distance problem of cost within a ρ_{ST} factor of $\mathsf{cost}(\mathbb{F})$.

We will refer to the sets S_i as *demand groups*, and a vertex in one of these groups as a *demand vertex*. For simplicity, we assume that for all i, $|S_i|$ is a power of 2; this can be achieved by replicating some vertices.

3.1 The Pairing Cost-Distance Problem: A Useful Subroutine

A *pairing* of any set A is a perfect matching on the graph $(A, \binom{A}{2})$. The following *tree-pairing lemma* has become an indispensable tool in network design problems (see [21] for a survey):

Lemma 3 ([18]). *Let T' be an arbitrary tree and let $v_1, v_2, \ldots, v_{2q}$ be an even number of vertices in T'. There exists a pairing of the v_i into q pairs so that the unique paths joining the respective pairs are edge-disjoint.*

Let us define another problem, whose input is the same as that for Group-Cost-Distance.

Definition 6 (Pairing Cost-Distance) *Given a graph $G = (V, E)$ with buy and rent costs b_e and c_e on the edges, and a set of demand groups $\{S_i\}_i$, the Pairing Cost-Distance problem seeks to find a* pairing $\mathcal{P}_i$ *of the nodes in S_i, along with a path connecting each pair of nodes $(x, y) \in \mathcal{P}_i$.*

Let $\mathbb{F}'$ be the set of edges used by these paths, and let x'_e be the number of pairs using the edge $e \in \mathbb{F}'$, then the cost of a solution is $\sum_{e \in \mathbb{F}'} (b_e + x'_e\, c_e)$. As before, given the set $\mathbb{F}'$, we can infer the best pairing that only uses edges in $\mathbb{F}'$ by solving a min-cost matching problem: we let $\mathsf{cost}'(\mathbb{F}')$ denote this cost, and let OPT' be the optimal solution to the Pairing Cost-Distance instance. [4] So, again, we can specify a solution to the Pairing Cost-Distance problem by specifying this set $\mathbb{F}'$. The following lemma relates the costs of the two closely related problems:

Lemma 4. *For any instance, the optimal cost* $\mathsf{cost}'(\mathsf{OPT}')$ *for Pairing Cost-Distance is at most the optimal cost* $\mathsf{cost}(\mathsf{OPT})$ *for Group-Cost-Distance.*

Proof. Let $\mathbb{F}$ be the set of edges bought by OPT for the Group-Cost-Distance problem. We construct a solution for the Pairing Cost-Distance problem. Recall that OPT builds a Steiner tree T_i spanning S_i using the edges in $\mathbb{F}$. By Lemma 3, we can pair up the demands in S_i such that the unique paths between the pairs in T_i are pair-wise edge-disjoint. This gives us a solution to Pairing Cost-Distance, which only uses edges in $\mathbb{F}$, and moreover, the number of times an edge is used is at most x_e, ensuring a solution of cost at most $\mathsf{cost}(\mathsf{OPT})$.

Lemma 5 (Reducing Group-Cost-Distance to Pairing Cost-Distance). *If there is an algorithm $\mathcal{A}$ for Pairing Cost-Distance that returns a solution $\mathbb{F}'$ with $\mathsf{cost}'(\mathbb{F}') \leq \alpha\ \mathsf{cost}(\mathsf{OPT})$, we get an $O(\alpha \log n)$-approximation for the Group-Cost-Distance problem.*

Note that $\mathcal{A}$ is not a true approximation algorithm for Pairing Cost-Distance, since we compare its performance to the optimal cost for Group-Cost-Distance; hence we will call it an *α-pseudo-approximation algorithm.*

Proof. In each iteration, when we connect up pairs of nodes in S_i, we think of taking the traffic from one of the nodes and moving it to the other node; hence the number of "active" nodes in S_i decreases by a factor of 2. This can only go on for $O(\log n)$ iterations before all the traffic reaches one node in the group,

[4] An important but subtle point: note that x'_e counts the number of paths that pass over an edge. It may happen that all of these paths may connect pairs that belong to the *same set* S_i, and cost might have to pay for this edge only once: regardless, we pay multiple times in cost'.

ensuring that the group is connected using these pairing paths. Since we pay at most $\alpha\,\mathsf{cost}(\mathsf{OPT})$ in each iteration, this results in an $O(\alpha \log n)$ approximation for Group-Cost-Distance.

4 An Algorithm for Pairing Cost-Distance

In this section, we give an LP-based algorithm for Pairing Cost-Distance; by Lemma 5 this will imply an algorithm for Group-Cost-Distance, and hence for Stochastic Steiner Tree.

We will prove the following result for Pairing Cost-Distance (PCD):

Theorem 7 (Main Result for Pairing Cost-Distance). *There is an $\alpha = O(\beta_{EEST} \log^2 H \cdot \log n)$ pseudo-approximation algorithm for the Pairing Cost-Distance problem, where $H = \max\{\sum_i |S_i|, n\}$.*

Since $H = O(Nn)$ and we think of $N \geq n$, this gives us an $O(\log^2 N \log^3 n \log\log n)$ pseudo-approximation. Before we present the proof, let us give a high-level sketch. The algorithm for Pairing Cost-Distance follows the general structure of the proofs of Chekuri et al. [4]; the main difference is that the problem in [4] already comes equipped with $\{s,t\}$-pairs that need to be connected, whereas our problem also requires us to figure out which pairs to connect—and this requires a couple of new ingredients.

Loosely, we first show the existence of a "good" *low density* pairing solution—this is a solution that only connects up *some* pairs of nodes in *some* of the sets S_i (instead of pairing up *all* the nodes in *all* the S_i's), but whose "density" (i.e., ratio of cost to pairs-connected) is at most a β_{EEST} factor of the density of OPT. Moreover, the "good" part of this solution will be that all the paths connecting the pairs will pass through a single "junction" node. The existence of this single junction node (which can now be thought of as a *sink*) makes this problem look somewhat like a low-density "pairing" version of single-sink cost-distance. We show how to solve this final subproblem within an $O(\log H \cdot \log n)$ factor of the best possible such solution, which is at most β_{EEST} times OPT's density. Finally, finding these low-density solutions iteratively and using standard set-cover arguments gives us a Pairing Cost-Distance solution with cost $O(\beta_{EEST} \log^2 H \cdot \log n)\,\mathsf{cost}(\mathsf{OPT})$, which gives us the claimed theorem.

4.1 Defining the Density

Consider an instance of the Pairing Cost-Distance (PCD) problem in which the current demand sets are $\widehat{S}_i$. Look at a partial PCD solution that finds for each set $\widehat{S}_i$ some set $\mathcal{P}_i$ of $t_i \geq 0$ mutually disjoint pairs $\{x^i_j, y^i_j\}_{j=1}^{t_i}$ along with paths P^i_j connecting these pairs. Let $\mathcal{P} = \cup_i \mathcal{P}_i$ be the (multi)set of all these

$t = \sum_i t_i$ paths. We shall use $\mathcal{P}$ to denote both the pairs in it and the paths used to connect them. Denote the cost of this partial solution by $\mathsf{cost}'(\mathcal{P}) = b(\cup_{P\in\mathcal{P}} P) + \sum_{P\in\mathcal{P}} c(P)$. Let $|\mathcal{P}|$ be the number of pairs being connected in the partial solution. The *density* of the partial solution $\mathcal{P}$ is defined as $\frac{\mathsf{cost}'(\mathcal{P})}{|\mathcal{P}|}$. Recall that $H = \max\{\sum_i |S_i|, n\}$ is the total number of terminals in the Pairing Cost-Distance instance.

Definition 8 (f-dense Partial PCD solution) *Consider an instance $\mathcal{I}$ of the Pairing Cost-Distance problem: a Partial PCD solution $\mathcal{P}$ is called* f-dense *if*

$$\frac{\mathsf{cost}'(\mathcal{P})}{|\mathcal{P}|} \leq f \cdot \frac{\mathsf{cost}(\mathsf{OPT})}{H(\mathcal{I})},$$

where $H(\mathcal{I})$ is the total number of terminals in the instance $\mathcal{I}$.

Theorem 9. *Given an algorithm to find f-dense Partial PCD solutions, we can find an $O(f \log H)$-pseudo-approximation to Pairing Cost-Distance.*

To prove this result, we will use the following theorem which can be proved by standard techniques (see e.g., [17]), and whose proof we omit.

Theorem 10 (Set Covering Lemma). *Consider an algorithm working in iterations: in iteration ℓ it finds a subset $\mathcal{P}_\ell$ of paths connecting up $|\mathcal{P}_\ell|$ pairs. Let H_ℓ be the number of terminals remaining before iteration ℓ. If for every ℓ, the solution $\mathcal{P}_\ell$ is an f-dense solution with $\mathsf{cost}'(\mathcal{P}_\ell)/|\mathcal{P}_\ell| \leq f \cdot \frac{\mathsf{cost}(\mathsf{OPT})}{H_\ell}$, then the total cost of the solution output by the algorithm is at most $f \cdot (1 + \ln H) \cdot \mathsf{cost}(\mathsf{OPT})$.*

In the next section, we will show how to find a Partial PCD solution which is $f = O(\beta_{EEST} \log H \cdot \log n)$-dense.

4.2 Finding a Low-Density Partial PCD Solution

We now show the existence of a partial pairing $\mathcal{P}$ of demand points which is β_{EEST}-dense, and where all the pairs in $\mathcal{P}$ will be routed on paths that pass through a common *junction* point. The theorems of this section are essentially identical to corresponding theorems in [4].

Theorem 11. *Given an instance of Pairing Cost-Distance on $G = (V, E)$, there exists a solution $\mathbb{F}'$ to this instance such that **(a)** the edges in $\mathbb{F}'$ induce a forest, **(b)** $\mathbb{F}'$ is a subset of OPT' and hence the buying part of $\mathsf{cost}'(\mathbb{F}') \leq b(\mathsf{OPT}')$, and **(c)** the renting part of $\mathsf{cost}'(\mathbb{F}')$ is at most $O(\beta_{EEST})$ times the renting part of $\mathsf{cost}'(\mathsf{OPT}')$.*

Proof Sketch. The above theorem can be proved by dropping all edges in $E \setminus \mathsf{OPT}'$ and approximating the metric generated by rental costs c_e in each resulting

component by a random subtree drawn from the distribution guaranteed by Theorem 5. We chose a subset of OPT', and hence the buying costs cannot be any larger. Since the expected distances increase by at most β_{EEST}, the expected renting cost increases by at most this factor. And since this holds for a random forest, by the probabilistic method, there must exist one such forest with these properties. □

Definition 12 (Junction Tree) *Consider a solution to the Partial PCD problem with paths $\mathcal{P}$, and which uses the edge set $\mathbb{F}'$. The solution is called a* junction tree *if the subgraph induced by $\mathbb{F}'$ is a tree and there is a special* root *vertex r such that all the paths in $\mathcal{P}$ contain the root r.*

As before, the *density* of a solution $\mathcal{P}$ is the ratio of its cost to the number of pairs connected by it. We can now prove the existence of a low-density junction tree for the Partial PCD problem. The proof of this lemma is deferred to the journal paper.

Lemma 6 (Low-Density Existence Lemma). *Given an instance of Pairing Cost-Distance problem, there exists a solution to the Partial PCD solution which is a junction tree and whose density is $2\beta_{EEST} \cdot \frac{\mathsf{cost}'(\mathsf{OPT}')}{H} \leq 2\beta_{EEST} \cdot \frac{\mathsf{cost}(\mathsf{OPT})}{H}$.*

In the following section, we give an $O(\log H \cdot \log n)$-approximation algorithm for finding a junction tree with minimum density. Since we know that there is a "good" junction tree (by Lemma 6), we can combine that algorithm with Lemma 6 to get a Partial PCD solution which is $f = O(\beta_{EEST} \log H \cdot \log n)$-dense.

4.3 Finding a Low-Density Junction Tree

In this section, we give an LP-rounding based algorithm for finding a junction tree with density at most $O(\log H \cdot \log n)$ times that of the min-density junction tree. Our techniques continue to be inspired by [4]; however, in their paper, they were given a fixed pairing by the problem, and had to figure out which ones to connect up in the junction tree. In our problem, we have to both figure out the pairing, and then choose which pairs from this pairing to connect up; we have to develop some new ideas to handle this issue.

The Linear-Programming Relaxation. Recall the problem: we are given sets S_i, and want to find some partial pairings for each of the sets, and then want to route them to some root vertex r so as to minimize the density of the resulting solution. We will assume that we know r (there are only n possibilities), and that the sets S_i are disjoint (by duplicating nodes as necessary).

Our LP relaxation is an extension of that for the Cost-Distance problem given by Chekuri, Khanna, and Naor [5]. The intuition is based on the following: given a junction-tree solution $\mathbb{F}'$, let $\mathcal{P}'$ denote the set of pairs connected via the root r.

Now $\mathbb{F}'$ can also be thought of as a solution to the Cost-Distance problem with root r and the terminal set $\cup_{(u,v)\in\mathcal{P}'}\{u,v\}$. Furthermore, the cost $\mathsf{cost}'(E')$ is the same as the optimum of the Cost-Distance problem. (This is the place when the definition of cost' becomes crucial—we can use the fact that the cost measure cost' is paying for the number of *paths* using an edge, not the number of *groups* using it.)

Let us write an IP formulation: let $\mathcal{S} = \cup_i S_i$ denote the set of all terminals. For each demand group S_i and each pair of vertices $u, v \in S_i$, the variable z_{uv} indicates whether we match vertices u and v in the junction tree solution or not. To enforce a matching, we ask that $\sum_u z_{uv} = \sum_v z_{uv} \leq 1$. For each $e \in E$, the variable y_e denotes whether the edge e is used; for each path from some vertex u to the root r, we let f_P denote whether P is the path used to connect u to the root. Let $\mathbb{P}_u$ be the set of paths from u to the root r. Clearly, we want $\sum_{P\in\mathbb{P}_u} f_P \leq x_e$ for all $e \in P$. Moreover, $\sum_{P\in\mathbb{P}_u} f_P \geq \sum_{v\in S_i} z_{uv}$ for each $u \in S_i$, since if the node u is paired up to someone, it must be routed to the root. Subject to these constraints (and integrality), we want to minimize

$$\min \frac{\sum_{e\in E} b_e x_e + \sum_{u\in\mathcal{S}} \sum_{P\in\mathbb{P}_u} c(P)\, f_P}{\sum_{i=1}^{N} \sum_{u,v\in S_i} z_{uv}} \tag{4.5}$$

It is not hard to check that this is indeed an ILP formulation of the min-density junction tree problem rooted at r. As is usual, we relax the integrality constraints, guess the value $M \geq 1$ of the denominator in the optimal solution, and get:

$$\begin{aligned}
\min \quad & \textstyle\sum_{e\in E} b_e\, x_e + \sum_{u\in\mathcal{S}} \sum_{P\in\mathbb{P}_u} c(P)\, f_P && && \text{(LP1)}\\
\text{s.t.} \quad & \textstyle\sum_{i=1}^{N} \sum_{u,v\in S_i} z_{uv} = M && &&\\
& \textstyle\sum_{P\in\mathbb{P}_u : P\ni e} f_P \leq x_e && \text{for all } u \in \mathcal{S} &\\
& \textstyle\sum_{P\in\mathbb{P}_u} f_P \geq \sum_{v\in S_i} z_{uv} && \text{for all } u \in S_i, i \in [1..N] &\\
& \textstyle\sum_{v\in S_i} z_{uv} \leq 1 && \text{for all } u \in S_i, i \in [1..N] &\\
& x_e, f_P, z_{uv} = z_{vu} \geq 0 && &
\end{aligned}$$

We now show that the integrality gap of the above LP is small.

Theorem 13. *The integrality gap of (LP1) is $O(\log H \cdot \log n)$. Hence there is an $O(\log H \cdot \log n)$-approximation algorithm for finding the minimum density junction tree solution for a given Partial PCD instance.*

Proof. Consider an optimal fractional solution given by (x^*, f^*, z^*) with value LP^*. We start off with $z = z^*$, and will alter the values in the following proof. Consider each set S_i, and let $w_i = \sum_{u,v\in S_i} z^*_{uv}$ be the total size of the fractional matching within S_i. We find an approximate maximum weight cut in the complete graph on the nodes S_i with edge weights z_{uv}—this gives us a bipartite graph which we denote by B_i; we zero out the z_{uv} values for all edges $(u, v) \in S_i \times S_i$

that do not belong to the cut B_i. How does this affect the LP solution? Since the weight of the max cut we find is at least $w_i/2$, we are left with a solution where $\sum_{i=1}^{N} \sum_{u,v \in S_i} z_{uv} \geq M/2$ (and hence that constraint is *almost* satisfied).

Now consider the edges in the bipartite graph B_i, with edge weights z_{uv}—if this graph has a cycle, by alternatively increasing and decreasing the values of z variables along this even cycle by ϵ in one of the two directions, we can make $z_{u'v'}$ zero for at least one edge (u', v') of this cycle without increasing the objective function. (Note that this operation maintains all the LP constraints.) We then delete the edge (u', v') from B_i, and repeat this operation until B_i is a forest.

Let us now partition the edges of the various such forests $\{B_i\}_{i=1}^{N}$ into $O(\log H)$ classes based on their current z values. Let $Z_{\max} = \max_{u,v} z_{uv}$, and define $p = 1+2\lceil \log H \rceil = O(\log H)$. For each $a \in [0..p]$, define the set $\mathcal{C}_a$ to contain all edges (u, v) with $Z_{\max}/2^{a+1} < z_{uv} \leq Z_{\max}/2^a$; note that the pairs $(u, v) \notin \cup_{a=1}^{p} \mathcal{C}_a$ have a cumulative z_{uv} value of less than (say) $Z_{\max}/4 \leq M/4$. Hence, by an easy averaging argument, there must be a class $\mathcal{C}_a$ with $\sum_{(u,v)\in\mathcal{C}_a} z_{uv} \geq \Omega(M/\log H)$. Define $Z_a = Z_{\max}/2^a$; hence $|\mathcal{C}_a|\, Z_a = \Omega(M/\log H)$.

Since we have restricted our attention to pairs in $\mathcal{C}_a$, we can define $B_{ia} = B_i \cap \mathcal{C}_a$, which still remains a forest. For any tree T in this forest, we apply the tree-pairing lemma on the nodes of the tree T, and obtain a matching $\mathcal{C}'_{ia}$ on S_i of size $\lceil |V(T)|/2 \rceil$. Defining $\mathcal{C}'_a = \cup_i \mathcal{C}'_{ia}$, we get that $|\mathcal{C}'_a|\, Z_a = \Omega(M/\log H)$ as well.

Finally, we create the following instance of the Cost-Distance problem. The terminal set contains all the terminals that are matched in $\mathcal{C}'_a$, and the goal is to connect them to the root. Set the values of the variables $\tilde{f}_P = f^*_P/Z_a$ and $\tilde{x}_e = x_e/Z_a$. These settings of variables satisfy the LP defined by [5] for the instance defined above. The integrality gap for this LP is $O(\log n)$ and so we get a solution with cost′ at most $O(\log n) \cdot LP^*/Z_a$. However, this connects up $|\mathcal{C}'_a| = \Omega(\frac{M}{Z_a \log H})$ pairs, and hence the density is $O(\log H \cdot \log n)\frac{LP^*}{M}$, hence proving the theorem.

5 Reduction from Single-Sink Cost-Distance

Theorem 14. *If there is a polynomial time α-approximation algorithm for the two-stage stochastic Steiner tree problem, then there is a polynomial time α-approximation algorithm for the single-source cost-distance problem.*

The hardness result of Theorem 2 follows by combining the above reduction with a result of Chuzhoy et al. [6] that the single-source cost-distance problem cannot be approximated to better than $\Omega(\log \log n)$ ratio under complexity theory assumptions.

Proof of Theorem 14. Consider an instance of the Cost-Distance problem: we are given a graph $G = (V, E)$, a root vertex r, and a set S of terminals. Each edge e has buying cost b_e and rental cost c_e. A solution specifies a set of edges E'

which spans the root and all the nodes in S: if the shortest path in (V, E') from $u \in S$ to r is P_u, then the cost of the solution is $b(E') + \sum_{u \in S} c(P_u)$. We take any edge with buying cost b_e and rental cost c_e, and subdivide this edge into two edges, giving the first of these edges a buying cost of b_e and rental cost ∞, and the other edge gets buying cost ∞ and rental cost c_e.

We reduce this instance to the two-stage stochastic Steiner tree problem where the scenarios are explicitly specified. The instance of the stochastic Steiner tree problem has the same graph. There are $|S|$ scenarios (each with probability $1/|S|$), where each scenario has exactly one unique demand from S. For an edge e which can only be bought, we set $c_M(e) = b_e$ and $c_T(e) = \infty$; hence any such edge must necessarily be bought on Monday, if at all. For an e which can only be rented, we set $c_M(e) = c_T(e) = |S| \cdot c_e$; note that there is no advantage to buying such an edge on Monday, since we can buy it on Tuesday for the same cost if needed — in the rest, we will assume that any optimal solution is lazy in this way.

It can now be verified that there is an optimal solution to the Stochastic Steiner Tree problem where the subset F' of edges bought in the first stage are only of the former type, and we have to then buy the "other half" of these first-stage edges to connect to the root in the second stage, hence resulting in isomorphic optimal solutions. □

6 Summary and Open Problems

In this paper, we gave a poly-logarithmic approximation algorithm for the stochastic Steiner tree problem in the non-uniform inflation model. Several interesting questions remain open. When working in the black-box model, we apply the scenario reduction method of Charikar et al. [2], causing the resulting number of scenarios N to be a polynomial function of the parameter λ, which is bounded by the maximum inflation factor on any edge. Hence our running time now depends on λ, and the approximation ratio depends on $\log \lambda$. Can we get results where these measures depend only on the number of nodes n, and not the number of scenarios N?

In another direction, getting an approximation algorithm with similar guarantees for (a) the stochastic Steiner *Forest* problem, i.e., where each scenario is an instance of the Steiner forest problem, or (b) the k-stage stochastic Steiner tree problem, remain open.

References

1. N. Alon, R. M. Karp, D. Peleg, and D. West, *A graph-theoretic game and its application to the k-server problem*, SIAM J. Comput., 24 (1995), pp. 78–100.

2. M. Charikar, C. Chekuri, and M. Pál, *Sampling bounds for stochastic optimization*, in RANDOM, vol. 3624 of Lecture Notes in Comput. Sci., Springer, Berlin, 2005, pp. 257–269.
3. M. Charikar and A. Karagiozova, *On non-uniform multicommodity buy-at-bulk network design*, in STOC, New York, NY, USA, 2005, ACM Press, pp. 176–182.
4. C. Chekuri, M. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour, *Approximation algorithms for non-uniform buy-at-bulk network design problems*, in FOCS, 2006.
5. C. Chekuri, S. Khanna, and J. S. Naor, *A deterministic algorithm for the cost-distance problem*, in SODA, 2001, pp. 232–233.
6. J. Chuzhoy, A. Gupta, J. S. Naor, and A. Sinha, *On the approximability of network design problems*, in SODA, 2005, pp. 943–951.
7. K. Dhamdhere, R. Ravi, and M. Singh, *On two-stage stochastic minimum spanning trees*, in IPCO, vol. 3509 of Lecture Notes in Comput. Sci., Springer, Berlin, 2005, pp. 321–334.
8. M. Elkin, Y. Emek, D. A. Spielman, and S.-H. Teng, *Lower-stretch spanning trees*, in STOC, New York, NY, USA, 2005, ACM Press, pp. 494–503.
9. A. Gupta and M. Pál, *Stochastic Steiner trees without a root.*, in ICALP, vol. 3580 of Lecture Notes in Computer Science, 2005, pp. 1051–1063.
10. A. Gupta, M. Pál, R. Ravi, and A. Sinha, *Boosted sampling: Approximation algorithms for stochastic optimization problems*, in STOC, 2004, pp. 417–426.
11. A. Gupta, M. Pál, R. Ravi, and A. Sinha, *What about Wednesday? approximation algorithms for multistage stochastic optimization.*, in APPROX, vol. 3624 of Lecture Notes in Computer Science, 2005, pp. 86–98.
12. A. Gupta, R. Ravi, and A. Sinha, *An edge in time saves nine: LP rounding approximation algorithms for stochastic network design.*, in FOCS, 2004, pp. 218–227.
13. M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour, *Approximating buy-at-bulk k-steiner trees*, Electronic Colloquium on Computational Complexity (ECCC), (2006).
14. ———, *Polylogarithmic approximation algorithm for non-uniform multicommodity buy-at-bulk*, Electronic Colloquium on Computational Complexity (ECCC), (2006).
15. A. Hayrapetyan, C. Swamy, and E. Tardos, *Network design for information networks*, in SODA, 2005, pp. 933–942.
16. N. Immorlica, D. Karger, M. Minkoff, and V. Mirrokni, *On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems*, in SODA, 2004, pp. 684–693.
17. D. S. Johnson, *Approximation algorithms for combinatorial problems*, J. Comput. System Sci., 9 (1974), pp. 256–278.
18. P. Klein and R. Ravi, *A nearly best-possible approximation algorithm for node-weighted Steiner trees*, J. Algorithms, 19 (1995), pp. 104–115.
19. A. Meyerson, *Online algorithms for network design*, in SPAA, 2004, pp. 275–280.
20. A. Meyerson, K. Munagala, and S. Plotkin, *Cost-distance: Two metric network design*, in FOCS, 2000, pp. 624–630.
21. R. Ravi, *Matching based augmentations for approximating connectivity problems.*, in LATIN, vol. 3887 of Lecture Notes in Computer Science, 2006, pp. 13–24.
22. R. Ravi and A. Sinha, *Hedging uncertainty: Approximation algorithms for stochastic optimization problems*, in IPCO, 2004, pp. 101–115.
23. G. Robins and A. Zelikovsky, *Tighter bounds for graph Steiner tree approximation*, SIAM J. Discrete Math., 19 (2005), pp. 122–134.

24. D. Shmoys and C. Swamy, *Stochastic optimization is (almost) as easy as deterministic optimization.*, in FOCS, 2004, pp. 228–237.
25. C. Swamy and D. B. Shmoys, *Sampling-based approximation algorithms for multi-stage stochastic*, in FOCS, 2005, pp. 357–366.

A An LP-based Algorithm for Multicommodity Cost-Distance

In their paper, Chekuri et al. [4] give an approximation algorithm for the Multicommodity Cost-Distance problem using a combination of combinatorial and LP-based techniques. We now give an LP-based algorithm for the Multicommodity Cost-Distance problem with an approximation guarantee of $O(\log^4 n)$, thus matching the result of [4] using a different approach. (Note that this is not the standard LP for the Multicommodity Cost-Distance problem, which we do not currently know how to round.)

Take an instance of Multicommodity Cost-Distance: let $S_i = \{s_i, t_i\}$ be each terminal pair we want to connect, and $\mathcal{S} = \cup_i S_i$. Consider the following linear program:

$$
\begin{aligned}
Z^*_{MCD} = \min \quad & \textstyle\sum_{e\in E} b_e\, z_e(p) + \sum_{u\in\mathcal{S}} \sum_{P\in\mathbb{P}_{u*}} c(P)\, f_P && \text{(LP-MCD)}\\
\text{s.t.} \quad & \textstyle\sum_{p\in V} x(u,p) \geq 1 && \text{for all } u \in \mathcal{S}\\
& \textstyle\sum_{P\in\mathbb{P}_{u,p}} f_P \geq x(u,p) && \text{for all } u \in \mathcal{S}, p \in V\\
& \textstyle\sum_{P\in\mathbb{P}_{u,p}:P\ni e} f_P \leq z_e(p) && \text{for all } u \in \mathcal{S}, p \in V\\
& x(s_i,p) = x(t_i,p) && \text{for all } i, p \in V\\
& x(u,p), f_P, z_e(p) \geq 0
\end{aligned}
$$

To understand this, consider the ILP obtained by adding the constraints that the variables are all $\{0,1\}$. Each solution assigns each terminal u to one junction node p (specified by $x(u,p)$) such that each s_i-t_i pair is assigned to the same junction (since $x(s_i,p) = x(t_i,p)$). It then sends the unit flow at the terminal using flow f_P to this junction p, ensuring that if an edge e is used, it has been purchased (i.e., $z_e(p) = 1$). The unusual part of this ILP is that the buying costs of the edge are paid not just once, but *once for each junction* that uses this edge. (If all the variables $z_e(p)$ would be replaced by a single variable z_e, we would get back the standard ILP formulation of Multicommodity Cost-Distance.)

Lemma 7. *Each integer solution to (LP-MCD) is an solution to the Multicommodity Cost-Distance problem with the same cost. Moreover, any solution to the Multicommodity Cost-Distance problem with cost* OPT *can be converted into a solution for (LP-MCD) with cost at most* $O(\log n) \times$ OPT.

While we defer the proof of this theorem to the final version of the paper, we note that the former statement is trivial from the discussion above, and the second

statement follows from the paper of Chekuri et al. [4, Theorem 6.1]. Hence, it suffices to show how to round (LP-MCD).

A.1 Rounding the LP for Multicommodity Cost-Distance

In this section, we show how to round (LP-MCD) to get an integer solution with cost $O(\log^3 n) \times Z^*_{MCD}$. The basic idea is simple:

- Given a fractional solution, we first construct a (feasible) partial solution with $\widehat{x}(u,p) \in \{0,1\}$ but with fractional f_P and $z_e(p)$ values. The expected cost of this partial solution is $O(\log^2 n) \times Z^*_{MCD}$.
- Since the $\widehat{x}$ values are integral in this partial solution, each terminal u sends unit flow to some set of junctions p (chosen by $\widehat{x}(u,p)$), such that each pair s_i, t_i sends flow to the same set of junctions. We can choose one of these junctions arbitrarily, and hence each terminal pair is assigned to some junction.
- Finally, note that all the flows to any junction p can be supported by fractional $z_e(p)$ capacities; these capacities are entirely separate from the capacities $z_e(p')$ for any other terminal p' in the graph. Hence the problem decomposes into several single-sink problems (one for each junction), and we can use the rounding algorithm of Chekuri et al. [5] to round the solutions for each of the junctions p *independently* to obtain integer flows f_P and integer capacities $z_e(p)$ while losing only another $O(\log n)$ in the approximation.

The last two steps are not difficult to see, so we will focus on the first rounding step (to make the $x(u,p)$ variables integral). The rounding we use is a fairly natural one, though not the obvious one.

1. Modify the LP solution to obtain a feasible solution where all $x(u,p)$ values lie between $1/n^3$ and 1, and each $x(u,p) = 2^{\eta(u,p)}$ for $\eta(u,p) \in Z_{\geq 0}$. This can be done losing at most $O(1)$ times the LP cost.
2. For each junction p, pick a random threshold $T_p \in_R [1/n^3, 1]$ independently and uniformly at random. Then round up f_P for all $P \in \mathcal{P}_{*p}$, and all $z_e(p)$ by a factor of $1/T$. Note that if $x(u,p) \geq T$, then we can send at least 1 unit of (fractional) flow from u to p using these scaled-up capacities $z_e(p)$. We set $\widehat{x}(u,p) = 1$, and call the terminal u *satisfied* by the junction p. (Note that since $x(s_i,p) = x(t_i,p)$, if s_i is satisfied by u then so is t_i.)
3. We repeat the threshold rounding scheme in the previous step $O(\log n)$ times. We return a solution $\{\widehat{x}, \widehat{f}, \widehat{z}\}$, where $\widehat{x}(u,p)$ is as described above, and $\widehat{f}, \widehat{z}$ are obtained by summing up all the scaled-up values of f and z over all the $O(\log n)$ steps.

We now show that each terminal has been satisfied with high probability, and that the expected cost of each solution produced in the second step above is $O(\log n) \times Z^*_{MCD}$.

Lemma 8. *The expected cost of the second step above is* $O(\log n) \times Z^*_{MCD}$.

Proof. The variable $z_e(p)$ gets scaled up to $z_e(p)/T$, whose expected value is $\int_{T=1/n^3}^{1} z_e(p)/T dT = z_e(p) \times O(\log N)$. The same holds for the f_P variables.

Lemma 9. *Each terminal* u *is satisfied by a single round of threshold rounding with constant probability, and hence will be satisfied in at least one of* $O(\log n)$ *rounds with probability* $1 - 1/\operatorname{poly}(n)$.

Lemma 10. *The above scheme gives a feasible solution* $\{\widehat{x}, \widehat{f}, \widehat{z}\}$ *to (LP-MCD) with* $\widehat{x} \in \{0, 1\}$ *and cost at most* $O(\log^2 n) \times Z^*_{MCD}$.

Finally, looking at this solution, decomposing it for each junction p, and using the [5] rounding scheme converts the flows $\widehat{f}$ and the capacities $\widehat{z}$ to integers as well. Combining this with Lemma 7 gives us another proof for the following result.

Theorem 15. *There is an* $O(\log^4 n)$ *approximation algorithm for the* Multicommodity Cost-Distance *problem based on LP rounding.*