6-2007

Infrastructure Leasing Problems

Barbara Anthony
Carnegie Mellon University

Anupam Gupta
Carnegie Mellon University, anupamg@cs.cmu.edu

Published In
.

Infrastructure Leasing Problems*

Barbara M. Anthony[1] and Anupam Gupta[2]

[1] Dept. of Mathematical Sciences, Carnegie Mellon University, Pittsburgh PA 15213
banthony@andrew.cmu.edu
[2] Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213
anupamg@cs.cmu.edu

Abstract. Consider the following Steiner Tree leasing problem. Given a graph $G = (V, E)$ with root r, and a sequence of terminal sets $D_t \subseteq V$ for each day $t \in [T]$. A feasible solution to the problem is a set of edges E_t for each t connecting D_t to r. Instead of obtaining edges for a single day at a time, or for infinitely long (both of which give Steiner tree problems), we *lease* edges for say, { *a day, a week, a month, a year* }. Naturally, leasing an edge for a longer period costs less per unit of time. What is a good leasing strategy? In this paper, we give a general approach to solving a wide class of such problems by showing a close connection between deterministic leasing problems and problems in multistage stochastic optimization. All our results are in the offline setting.

Keywords: Approximation algorithms, graph and network algorithms, stochastic combinatorial optimization, randomized algorithms.

1 Introduction

Traditional network design problems require us to make decisions about how to send data, and how to provision bandwidth on various links of the network. A standard feature in most models for network design that have been considered, and in the algorithms that have been developed, has been the *permanence* of the bandwidth allocation—and this has been true even in cases where demands arrive online: once some amount of bandwidth is allocated on an edge, this bandwidth can be used *at any time in the future* (perhaps by paying some additional incremental "routing cost" per unit of flow). Some works have also considered the question of buying versus renting, but the simplifying assumption again has been that buying gives permanent access to the commodity. *But what if we are allowed only to lease bandwidth on the links of the network for fixed lengths of time: which leases on which network links should we obtain over time to satisfy our demands?*

Given a situation with multiple lease lengths, it is natural to assume that a longer lease is a cheaper one (per day), and that we pay more dearly for the

* This research is partly supported by an NSF CAREER award CCF-0448095, and by an Alfred P. Sloan Fellowship.

flexibility afforded by the short-term leases.[3] Hence, if our traffic consists of some stable parts and other bursty parts, we can use long-term leases to satisfy the stable traffic, and the short-term leases to handle the more volatile demands: a clever leasing strategy can reduce costs substantially over a naïve one. Note that solving this problem requires us to simultaneously perform clustering over space (in order to figure out which edges to allocate bandwidth on) and over time (to figure out which traffic is stable and requires longer leases, and which is bursty and is best served by shorter leases).

The question of finding good leasing strategies is relevant in the context of other problems as well: in planning for demands arriving over multiple periods in classical facility location problems, one might want to lease warehouses/plants for varying lengths of time. Moreover, the idea that leases of varying lengths are available is fairly natural: even in situations where there is a standard lease length (say plants are usually leased for a year), the presence of a secondary market for reselling or sub-letting might naturally give rise to the situation with multiple lease lengths we consider in this paper.

In this paper, we initiate a systematic study of *Leasing problems*, and give algorithms for several classic infrastructure design problems in the presence of finite-duration leases. To illustrate our general model, we will use the Steiner Tree Leasing problem as our running example.

> We are given a graph $G = (V, E)$ with a root r. For each day t, we are given a set of terminals D_t and a set K of permissible lease lengths, where the cost of leasing any edge e for length $\ell \in K$ is $c(\ell)$: we ensure that for any lengths $\ell_1 < \ell_2$ in K, $c(\ell_2) \leq c(\ell_1) \times \frac{\ell_2}{\ell_1}$. Note that an edge leased on day t for duration ℓ can be used on any of the days $t, t+1, \ldots, t+\ell-1$, and is said to be *active* on all these days. Define $X_t(\ell) \subseteq E$ to be the set of edges leased for duration ℓ on day t, and $F_t = \cup_{\ell \in K} \cup_{j \in [t-\ell+1,t]} X_j(\ell)$ to be the set of active edges on day t. A solution (given by edge sets $X_t(\ell)$ for all t and ℓ) is feasible if on each day t, the induced active edge sets F_t connect the demand set D_t to the root r. The goal is to find a feasible solution of minimum cost $\sum_{t,\ell}[c(\ell) \times |X_t(\ell)|]$.

One can follow this general idea and define other infrastructure design problems: in Facility Location Leasing, we are given demand sets D_t for each day, and may want to lease different facilities for different periods of time, with the goal of minimizing the resulting cumulative facility opening costs plus the connection costs for the clients on their respective days. (In this case, one may even imagine a "non-uniform" scenario where the different facilities have different lease cost functions.) And an even more general problem is that of Set Cover Leasing, where we are given sets $D_t \subseteq U$ of elements to cover on the t^{th} day, and want to lease sets such that the active sets at time t form a feasible cover of the set D_t.

While such problems of finite-period leases are related to the substantial body of work on *perishable commodities* [29,13] in inventory theory, we are not aware

[3] More formally, we assume leasing for length ℓ costs no more than two leases of length $\ell/2$. This sub-additive cost structure also allows amortization of one-time costs.

of work that directly addresses the questions under consideration in this paper. Loosely speaking, given supply of a perishable good—e.g., cartons of milk with a lifetime of ℓ days—and demands over time, research on perishable commodities has considered questions pertaining to inventory positions (in deterministic vs stochastic settings, with several classes of customers, etc.), and to pricing such perishable goods. At a high level, our leasing problems can be viewed as solving multiple perishable goods problems to solve a global network design problem.

1.1 Our Results and Techniques

The main result of this paper is the following, showing a close connection between leasing problems as described above, and stochastic optimization problems.

Theorem 1 (General Leasing Theorem). *The offline leasing version of a subadditive combinatorial optimization problem Π with $|K| = k$ lease lengths can be reduced to the stochastic optimization version of Π in the model of k-stage stochastic optimization with recourse.*

We feel this theorem is somewhat surprising: even though the leasing version of the problem Π can be *completely deterministic* with a given input and no stochastic component, this theorem shows that an algorithm to solve the (multi-stage) stochastic version of the problem suffices to solve the (non-stochastic) leasing problem. The proof of this theorem turns out to be fairly clean, and appears in Section 4.1. Given this main theorem, we can use recently-developed algorithms for multistage stochastic combinatorial optimization [34,37] to infer:

Corollary 1 (Optimal Algorithms for Leasing). *There exist an $O(1)$-approximation algorithm for the Vertex Cover Leasing problem, an $O(k)$-approximation*[4] *for the Facility Location Leasing problem, and an $O(\log n)$-approximation for the Set Cover Leasing problem.*

All these results are asymptotically optimal (up to constants). For the Steiner Tree Leasing problem we were using as our running example, we get the following result by combining Theorem 1 with known results [17,19].

Theorem 2 (Steiner Tree Leasing). *There is an $O(\min\{k, \log n\})$-approximation algorithm for offline Steiner Tree Leasing with $|K| = k$ lease lengths.*

It seems improving the approximation to $o(k)$ requires techniques that also improve results for the Stochastic Steiner Problem, which remains an open question.

New Algorithms for Network Problems: We go on to study other network leasing problems that generalize the Steiner Tree Leasing problem. In these problems, instead of just connecting up the terminals, we are now required to allocate "sufficient" bandwidth on the connecting edges as well. However, the cost of allocating bandwidth is itself a concave function $g(b)$ of the amount of bandwidth

[4] Corrected from the proceedings version, which erroneously stated $O(1)$.

b allocated on the edge: these are commonly known as *buy-at-bulk* problems. In the leasing framework, this translates into problems where the cost of leasing b units of bandwidth for a period of length ℓ is $c(\ell) \times g(b)$.

Theorem 3 (Buy-at-Bulk Theorems). *There is an $O(k)$ approximation for the k-stage Stochastic versions of the single-sink Rent-or-Buy, and the single-sink Buy-at-Bulk problems. Moreover, the Stochastic Buy-at-Bulk problem with multiple sinks has an $O(k \log n)$ approximation algorithm.*

By Theorem 1, we get the same approximation ratios for the corresponding network leasing versions of these problems as well.

Related Work. There has been a tremendous amount of work on network design where the the cost of bandwidth obeys natural economies of scale (often called "buy-at-bulk" network design). It is beyond the scope of this paper to survey this body of work, so we point the reader to [25,26,4,32,2,14,12,38,1,10] and the many references therein. This line of work is related to our work both in spirit, as well as in some of the technical methodology. In this paper, we also show how we can extend some of the current algorithms for these "buy-at-bulk" problems to the case when the bandwidth is leased and not bought permanently.

As mentioned above, leasing for finite periods is related to a large body of work on perishable commodities [29,13] in inventory theory; however, to the best of our knowledge, such problems have not been directly considered in the literature.

The Steiner Tree Leasing problem was first explored in a paper on the "parking permit problem" [27]. The paper noted that dynamic programming could be used to solve the Steiner Tree Leasing problem when the graph was a single edge (or to obtain an approximation scheme if the numbers are large), and gave an $O(\log k)$ competitive algorithm in the *online* case where the terminal set D_t is revealed only on day t. These results can be extended naturally to general graphs using standard tree-approximation techniques [5,11] by losing an extra $O(\log n)$ factor. However, it does not seem clear how to improve their techniques directly in the offline case to avoid this loss of $O(\log n)$ and obtain an approximation dependent only on k, or to extend them to the other problems we consider here.

In this paper, we show a concrete connection of network leasing to multistage stochastic optimization problems. While the history of stochastic optimization begins in the 1950s, this work is directly related to recent work on approximation algorithms for stochastic combinatorial problems [9,20,31,16,19,33,8,7]. We draw most directly from the results of [19,17,34] on the multistage stochastic optimization problems, and on the results in [16,17] to convert algorithms for the non-stochastic versions of problems to their multistage stochastic counterparts.

A standard tool in algorithms design today is the tree approximation technique of [5,11], as well as the general techniques for solving covering problems from, e.g., [30,35,36,23]. These techniques will allow us to get some simple approximation bounds; one of the goals of this paper is to develop algorithms that beat these naïve bounds by making use of the combinatorics of the problems, and to explore connections to problems in multistage stochastic optimization.

As an aside, let us note that a problem called the "Network Leasing" problem has been previously studied in the literature [3]; since that problem has come to be better known as the "Rent-or-Buy" problem, we have taken the liberty of claiming the term "leasing" to refer to an orthogonal concept in this paper.

2 Models and Notation

Consider a general subadditive optimization problem Π with k lease lengths. Formally, we are given a set U of potential *clients* or *demands*, such that on each day $t \in \{1, 2, \ldots, T\}$, some subset $D_t \subseteq U$ of these clients actually appear and demand service. (We will soon discuss how these sets D_t are given to us.) We also have a set of *elements* X that we can use to build solutions: for each subset of clients $D \subseteq U$, we are given some set of *solutions* $\mathsf{Sols}(D) \subseteq 2^X$ to the client set D. On day t, we would like to own a set of elements $F_t \in \mathsf{Sols}(D_t)$.

If each element could only be leased for a single day at a time, then this would just require us to solve T instances of the problem Π; on the other hand, if elements could only be leased indefinitely (i.e., "bought"), we would just solve the problem on $\cup_t D_t$. The "leasing" aspect of the problem is reflected in the fact that each of these elements $e \in X$ can be leased for several periods: i.e., on any day t, given any duration $\ell \in K$, one can obtain a lease of length ℓ on element $e \in X$ for cost $c_e(\ell)$ and use it on days $t, t+1, \ldots, t+\ell-1$. Formally, let $X_t(\ell)$ be the elements for which leases of length ℓ were obtained on day t, and $F_t = \cup_{\ell \in K} \cup_{t'=t-\ell+1}^{t} X_t(\ell)$, then a *feasible leasing strategy* is a sequence of sets $X_t(\ell)$ which results in $F_t \in \mathsf{Sols}(D_t)$ for each day t.

Definition 1 (Uniform vs. Non-Uniform). *A leasing problem is called* uniform *if the cost functions $c_e(\cdot)$ for all elements $e \in X$ are identical (here we will drop the subscript and refer to it as $c(\cdot)$), and is called* non-uniform *otherwise.*

As may be expected, we will be able to obtain better results for uniform problems in some cases. One immediate advantage of uniform network design problems will be the applicability of tree-approximation techniques (see Lemma 3); see also Section 4.1 for other advantages of uniformity.

Stochastic Optimization. The relevant stochastic model is k-stage stochastic optimization with recourse: the demand set D is revealed on day-k drawn from some known distribution π, but on each of days $1, 2, \ldots, k-1$, we are given additional information about the set D. (One can view this process as having a joint distribution over "signals" $s_1, s_2, \ldots, s_{k-1}, s_k$ received on the various days, with the actual demand set some known function of this signals.) One can see, e.g., [34,17] for more details about the model. The costs of elements change over time (usually getting more expensive over time): the *uniform inflation* model assumes the cost $cost_t(e)$ of element $e \in X$ on day-t (or *stage-t*) to be $\sigma_i \times cost_{t-1}(e)$ (and hence $cost_1(e)$ $\prod_{1<j\leq i} \sigma_j$). Note that the σ_i's are uniform, and independent of the element e. In the more powerful *non-uniform* model, the costs of different elements can change differently as time progresses.

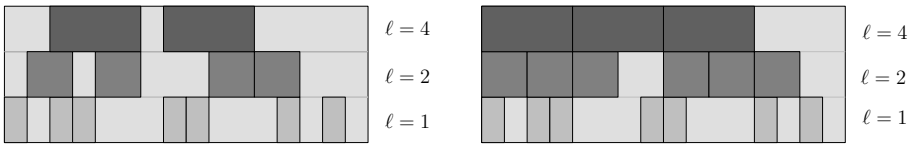

Fig. 1. A solution, and the corresponding nested version (right), as in Lemma 2.

We use the Boosted Sampling framework to develop new algorithms for some network design problems: these will require us to use terminology about cost shares, which can be found in Appendix A.

3 Observations and Reductions

Before we give the main results of this paper, we give some observations which will be helpful in the rest of the paper. We investigate how solutions can be assumed to have a simple structure, what results tree-approximations can give for Steiner Tree Leasing (giving us a baseline to compare to), and what tree-approximation techniques can give for more complex network leasing problems.

Structure of Solutions. The following two lemmas allow us to impose a simple structure on the instances we solve and solutions we seek. They are fairly standard (e.g., [27, Thms 2.1 & 2.2]) and are given for completeness. Recall that the set of permissible lease lengths is $K = \{\ell_1, \ell_2, \ldots, \ell_k\}$ with $\ell_1 < \ell_2 < \ldots < \ell_k$.

Lemma 1. *Given any instance $\mathcal{I}$ of a leasing problem, we can convert it into an instance $\mathcal{I}'$ in which the lengths of leases exactly divide each other (i.e., $\ell_i | \ell_j$ for $i < j$), and where the costs satisfy $c(\ell_j) < c(\ell_i) \times (\ell_j/\ell_i)$. Moreover, there is an optimal solution to $\mathcal{I}'$ which has cost at most 2 times the optimal cost for $\mathcal{I}$.*

The above lemma can be proved, e.g., by rounding all the lease lengths down to the closest powers of 2, and by discarding leases that do not satisfy the sub-additivity property. The following lemma shows that we can focus our attention only on "nested" solutions; i.e., solutions where we never have a short-term lease still active when a longer-term lease begins or ends.

Lemma 2 (Nested Solutions). *Given an instance $\mathcal{I}$ of a leasing problem, there is a solution which has cost at most 2 times the optimum, where a lease of length ℓ is obtained only for intervals of the form $[t, t+\ell)$ with t a multiple of ℓ.*

See Fig. 1 for an example of a non-nested solution on the left, and a nested solution whose cost is at most twice the cost of the former.

Reduction to Trees/Single-Edges. Given a graph $G = (V, E)$, a theorem of Fakcharoenphol et al. [11] (see also [5]) says that there is a distribution $\mathcal{D}$ over dominating trees T (i.e., $d_G(u, v) \leq d_T(u, v)$ for any T in the support of $\mathcal{D}$) such that the expected stretch $\frac{E_{T \leftarrow \mathcal{D}}[d_T(u,v)]}{d_G(u,v)} \leq O(\log n)$. The following use of this result is fairly standard by now (see [2]).

Lemma 3 (Reduction to Trees/Edges). *Given an instance of Steiner Tree Leasing which is* uniform *(where the cost functions $c_e(\cdot)$ are the same for all edges), an α-approximation for the single-edge case gives an α approximation for trees, and an $O(\alpha \log n)$ approximation for the general graph case.*

The proof uses the fact that the reduction to a tree instance loses an $O(\log n)$; once on a tree, the paths to be chosen are unique, and hence it suffices to run the single-edge algorithm on each edge to determine when to lease it. (The simple details are deferred to the final version of the paper.) Since we can solve the leasing problem on a single edge exactly, we get an $O(\log n)$-approximation for the Steiner Tree Leasing problem.

General (Uniform) Leasing Strategies and CIPs. Consider a much more general network design problem where at each time step t we are given a traffic matrix D_t, and want to allocate enough bandwidth to route D_t. We are now given a set $\mathcal{L} = \{L_j = (I_j, b_j, p_j)\}_j$ of possible leases, where each lease L_j in $\mathcal{L}$ is specified by a time interval I_j during which this lease is active, an amount b_j of bandwidth and a price p_j for it. Moreover, for any lease L_j, we may have an upper bound u_j on the number of copies of this lease we can buy per edge. This is a much more general model than the one we have been looking at, since we allow "one-time-only" offers (a special deal valid only for some days at a special price, limit one only), etc: this captures Buy-at-Bulk Leasing, and much more.

However, as long as the problem is *uniform* (i.e., each edge e has the same set $\mathcal{L}$ of potential leases), we can use a reduction akin to Lemma 3 to randomly reduce the problem to a tree and hence to a single edge, where it can be solved using general theorems on CIPs, covering integer problems techniques (e.g., see [30,35,6,36,23]). Applying these techniques to our problems give us approximation ratios that typically depend on $\log \ell_{\max}$, and $\log b_{\max}$, where $b_{\max}$ is the maximum bandwidth requirement. (See the full version for precise details.) In this paper, we attempt to give algorithms that are better—i.e., independent of $\ell_{\max}$; it is easy to see that $\log \ell_{\max} \geq k$, and we think of $\log \ell_{\max} \gg k$.

4 Algorithms for Leasing Problems

In this section, we will prove the main result: that Leasing Problems can be cast as Stochastic Optimization problems. This will allow us to get approximation algorithms for a variety of leasing problems from the corresponding algorithms for stochastic optimization. While we use many stochastic algorithms already in the literature, we will give new algorithms for some problems like Stochastic Rent-or-Buy and Stochastic Buy-at-Bulk, and hence for their leasing versions.

4.1 Reduction to Multistage Stochastic Optimization

Let us assume, without loss of generality, that $\ell_1 = 1$, and denote the maximum lease length by $\ell_{\max}$. By Lemma 2 we can assume that our solutions are nested.

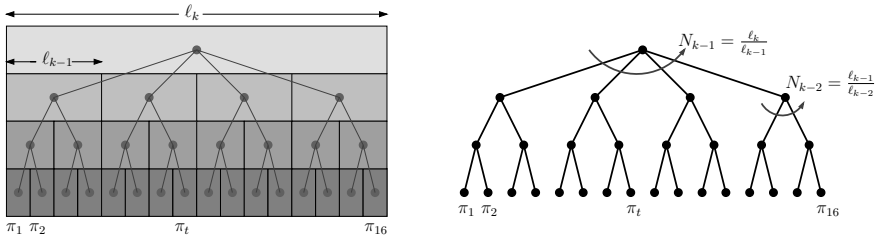

Fig. 2. A nested leasing instance ($k = 4$), and the resulting stochastic tree $\mathfrak{T}$

Theorem 4 (Reduction to k-stage Stochastic Optimization). *Any (non-uniform) offline problem Π in the above framework with $|K| = k$ lease lengths can be reduced to the standard k-stage stochastic optimization version of Π.*

Proof. As in the proof of Theorem 2, let us consider tiling time by intervals of length ℓ_k, each of which are divided into $N_{k-1} = \frac{\ell_k}{\ell_{k-1}}$ consecutive intervals of length ℓ_{k-1}, each of which are further subdivided into $N_{k-2} = \frac{\ell_{k-1}}{\ell_{k-2}}$ intervals of length ℓ_{k-2}, and so on. Note that this gives a different representation of time: we can describe time $t = \sum_{p=1}^{k} j_p \ell_p$ as a k-tuple of the form $(j_k, j_{k-1}, \ldots, j_1)$—and we will denote this tuple by $\bar{\tau}(t)$. (Note that j_p is simply $\lfloor t/\ell_p \rfloor \pmod{\ell_{p+1}}$, where we assume $\ell_{k+1} = \infty$). Corresponding to this notation, we will refer to the set $X_t(\ell_i)$ also as $X_{(j_k, j_{k-1}, \ldots, j_1)}(\ell_i)$, where t, ℓ_i and the j_k's are as above.

Recall that we are looking for nested solutions, and hence each lease of length ℓ_i will be obtained at the beginning of some interval of length ℓ_i; hence $X_t(\ell_i) = \emptyset$ for $t \not\equiv 0 \pmod{\ell_i}$. Moreover, since the longest interval is of length ℓ_k, *all* permits will have to be purchased afresh at the end of each length ℓ_k interval, and hence we can focus on the time interval from 0 to $T = \ell_k - 1$. Using these facts, consider a leasing solution that for each $t \in [T]$ and $p \in [k]$, buys leases of length ℓ_p on the elements in $X_t(\ell_p)$ at time t. The (expected) cost of this solution is[5]

$$\mathbf{E}\left[\sum_{e \in X_0(\ell_k)} c_e(\ell_k) + \sum_{t:\ell_{k-1}|t} \sum_{e \in X_t(\ell_{k-1})} c_e(\ell_{k-1}) + \sum_{t:\ell_{k-2}|t} \sum_{e \in X_t(\ell_{k-2})} c_e(\ell_{k-2}) + \ldots\right]. \quad (1)$$

We now define an instance of the k-stage stochastic optimization problem $\mathsf{Stoc}_k(\Pi)$ with the same optimal value as (1), and hence an α-approximation to the stochastic problem gives an α-approximation to our network leasing problem.

The Stochastic Instance Consider the tree $\mathfrak{T}$ in Fig. 2 where the root has $N_{k-1} = \frac{\ell_k}{\ell_{k-1}}$ children, each node at depth 1 has $N_{k-2} = \frac{\ell_{k-1}}{\ell_{k-2}}$ children, and so on. This gives rise to ℓ_k leaves associated with the distributions $\pi_1, \pi_2, \ldots, \pi_{\ell_t}$ from

[5] We allow randomized leasing policies, and so expectation is over the coin tosses of our algorithm, as well as over randomness in the choice of the sets S_t in case we are working in the *stochastic* offline model where S_t is drawn from the distribution π_t.

left to right. The k-stage stochastic problem now involves k stages of decision-making. In the first stage, a particle is placed at the root, and we buy a set $Y_1 \subseteq X$, where element $e \in X$ costs $c_e(\ell_k)$. After this, the particle moves to one of the children of the root at random; after we learn the identity of this vertex of $\mathfrak{T}$, we can buy a "stage-2" set $Y_2 \subseteq X$, but the cost of e now becomes $c_e(\ell_{k-1}) \times N_{k-1}$. In this way, after t steps, the particle reaches some node at depth t, whence we buy some "stage-t+1" set $Y_{t+1} \subseteq X$ with the costs $c_e(\ell_{k-t}) \times \prod_{1 \le p \le t} N_{k-p} = c_e(\ell_{k-t}) \times \frac{\ell_k}{\ell_{k-t}}$. Finally, when the particle reaches some leaf v_k (at depth $k-1$, say it is the t^{th} leaf), the algorithm finally gets a random set of clients $S_t \in_R \pi_t$, and must output a set Y_k such that $Y_1 \cup Y_2 \cup \ldots \cup Y_k \in \mathsf{Sols}(S_t)$; as above, the costs are now $c_e(\ell_1) \times \prod_{1 \le p \le k} N_{k-p} = c_e(\ell_1) \times \frac{\ell_k}{\ell_1}$.

The Correspondence Note that a solution to this process associates a (potentially) random set $Y(v)$ with each vertex of tree $\mathfrak{T}$; the expected cost is

$$\mathbf{E}\left[\sum_{e \in Y(\text{root})} c_e(\ell_k) + \sum_{p=1}^{k-1} \sum_{v \text{ at depth p}} \Pr[\text{reach } v] \sum_{e \in Y(v)} c_e(\ell_{k-p}) \times \frac{\ell_k}{\ell_{k-p}}\right] \quad (2)$$

Finally, we place the nodes at level p of $\mathfrak{T}$ in correspondence with integers t such that $\ell_{k-p} | t$, associate $Y(v)$ with $X_t(\ell_{k-p})$, and observe the probability of reaching any fixed node at level p is $\frac{\ell_{k-p}}{\ell_k}$ to get that (2) and (1) are identical.

Costs and Inflations The instances of $\mathsf{Stoc}_k(\Pi)$ created by the reduction above have the property that when we go *from stage $p-1$ to stage p* of the stochastic problem, the cost of each element e increases by an *inflation factor* of

$$\sigma_{e,p} \doteq \frac{c_e(\ell_{k-p+1}) \times N_{k-p+1}}{c_e(\ell_{k-p+2})}, \quad (3)$$

which by our assumptions is at least 1. If the leasing problem was uniform (the functions $c_e(\cdot)$ were the same for all $e \in X$), this inflation parameter depends only on the stage p but not on the element e (the *uniform inflation* case). But, if the leasing problem was non-uniform, we get a *non-uniform* inflation stochastic problem. This distinction will be useful, since depending on the problem Π, different approximation guarantees exist for uniform and non-uniform versions.

4.2 Leasing Algorithms from Existing Stochastic Algorithms

There has been much recent work on designing algorithms for multistage stochastic optimization with provable guarantees; see [34,17,19]; some are in the uniform inflation model, whereas others are more general. Using Theorem 4, we get:[6]

[6] The proceedings version erroneously claimed $O(1)$-approximations for facility location due to Srinivasan. No such approximations are currently known; his paper provides improved results for $k = 2$.

Problem	Inflation type	Approximation Ratio for Leasing problem	Stochastic Citation
Steiner Tree	uniform	$8k$	$2k$ [17,19]
Facility Location	non-uniform	$6.84k$	$1.71k$ [34]
Vertex Cover	non-uniform	8	2 [28,37]
Set Cover	non-uniform	$4 \ln n$	$\ln n$ [37]

We note that as presented, the algorithms for the k-stage stochastic problems specify which elements to buy in an "online-like" fashion; given the observations of what has happened in the past, the stochastic algorithms prescribe the elements to buy at the current time instant. In particular, they do not give an *explicit representation* of the sets $Y(v)$ of elements to buy for each node v of the distribution tree $\mathfrak{T}$. However, the above algorithms can easily be altered to give all these sets; the details are deferred to the final version of the paper.

5 New Stochastic/Leasing Approximations

In this paper, we give new results for k-stage stochastic optimization (and hence for Network Leasing) on a group of network design problems, all of which lie under the umbrella of "buy-at-bulk"-type problems. In these problems, the demand D_t for day t is not just a set of clients that have to be connected (as in Steiner Tree), but instead is a traffic matrix specifying how much traffic flows between various pairs of nodes in the network. In addition to the lease-cost function $c : K \to \mathbb{R}_+$ given earlier, we are also given a "bandwidth-cost" function $g : \mathbb{R}_+ \to \mathbb{R}_+$. The cost of leasing b bandwidth on an edge for ℓ length of time is now $\mathsf{Cost}(b, \ell) = g(b) \times c(\ell)$. (We consider these problems only in the *uniform model*, and hence both the functions $c(\cdot)$ and $g(\cdot)$ are the same for each edge.)

We will give the following results for some buy-at-bulk type problems, using the Boosted Sampling approach and defining "strict" cost-shares to prove these results; a quick overview is provided in Appendix A.

Problem	Inflation	Approximation Ratio	Citation
Buy-at-Bulk	uniform	$O(k \log n)$	Theorem 5
Single-Sink Rent-or-Buy	uniform	$O(k)$	Theorem 6
Single-Sink Buy at Bulk	uniform	$O(k)$	Theorem 7

5.1 Multiple-Sink Buy-at-Bulk

There are many ways to specify the Buy-at-Bulk problem which are all equivalent to within a factor of 2 (see, e.g., [38]), so let us fix one. We are given a demand matrix $D \in \mathbb{R}^{n \times n}$ where D_{ij} gives the traffic from v_i to v_j. We have a monotone subadditive cost function $g(\cdot)$, where the cost of bandwidth b is $g(b)$. By well-

known properties of subadditive functions, we can find a *concave* cost function $h(\cdot)$ such that $g(b) \leq h(b) \leq 2g(b)$ for all $b \neq 0$. We assume that the cost of bandwidth allocation is $h(b)$ for all non-zero values of b; this only changes the problem by a factor of 2.

The best-known algorithm for the Buy-at-Bulk problem is by Awerbuch and Azar [2]. We approximate the graph by a random tree (as in Lemma 3), and given the Buy-at-Bulk problem on the tree, we can solve it on an edge-by-edge basis. We now show how to get an algorithm for the stochastic version.

Theorem 5. *The k-stage stochastic version of the Buy-at-Bulk problem on the tree has an $O(k)$ approximation, and hence Buy-at-Bulk on general graphs has an $O(k \log n)$ approximation.*

Proof. Let us give an algorithm for a single edge in the tree that separates V into A and $V \setminus A$: we can calculate the traffic crossing this edge e as $D_e = \sum_{ij \in \partial A} D_{ij}$. For this edge, we allocate capacity D_e and divide the cost $h(D_e)$ equally among each of the D_e units of demand. Clearly the cost shares are cross-monotone: if more demand passes through the edge, the cost only decreases because h is concave. Moreover, the algorithm is a 1-approximation with respect to these cost-shares, since we share the exact cost of the algorithm amongst the players.

Moreover, we can check that these cost shares are 1-c-strict (as defined in (5)): indeed, if we divided the traffic D_e into two parts S and T, and allocated S units of bandwidth first, then the cost shares $\xi(X/\mathcal{A}(S), T, T) = h(S+T) - h(S)$ would be at most the cost-shares $\xi(X, S \cup T, T) = h(S+T) \times \frac{T}{S+T}$ ascribed to T when both S and T were in the fray; this follows from the concavity of h.

Given that we have 1-c-strict and cross-monotone cost shares ξ and a 1-approximation algorithm $\mathcal{A}$ with respect to ξ, we can apply Theorem 8 to infer a k-approximation (with respect to the cost function h), and hence a $2k$-approximation with respect to the original cost function g. Finally, since we moved to a random tree, we lose another $O(\log n)$ in translating the solution back to the original graph G. This concludes the proof.

5.2 Single-Sink Buy-at-Bulk Problems

In the Single-Sink Rent-or-Buy problem (a special case of the Buy-at-Bulk problem), we are given a graph $G = (V, E)$ with a distinguished *root* vertex r. Each vertex j wants to send d_j amount of traffic to r. The bandwidth cost function is $g(b) = \min\{b, M\}$ for some parameter M. We show the following result:

Theorem 6. *The Single-Sink Rent-or-Buy problem has an $O(1)$-approximation algorithm with respect to 1-c-strict cost sharing functions; moreover, these cost-shares are cross-monotone.*

Combined with Theorem 8, this gives an $O(k)$-approximation for stochastic Single-Sink Rent-or-Buy, and hence an $O(k)$-approximation for the Single-Sink Rent-or-Buy Leasing problem, where buying b bandwidth for ℓ costs $g(b) \cdot c(\ell)$.

Proof. The algorithm $\mathcal{A}$ is the SimpleCFL algorithm from [15]. This algorithm starts off with $F = \{r\}$, and add each vertex j to F independently with probability d_j/M. It then builds an approximate Steiner tree on F using the MST heuristic, and allocates unlimited capacity on its edges (hence paying M on each such edge). It then sends d_j units of flow from j to its closest vertex in F (which may be j itself, in case $j \in F$); for this it pays cost 1 per unit of flow.

Define the cost-share for node j as $\xi_{RoB}(v) = \mathbf{E}[M\,\xi_{MST}(v)] + \mathbf{E}[d_j\,l(v,F)]$. (Here ξ_{MST} is a cross-monotonic cost-sharing function ξ_{MST} for the minimum spanning tree problem—e.g., given in [22,21], and $l(v,F)$ is the distance from v to the nearest vertex in F.) It is known that ξ_{RoB} is cross-monotone, and moreover that $\mathcal{A}$ is a 4-approximation for Single-Sink Rent-or-Buy with respect to these cost-shares ξ_{RoB} [24,18].

We claim ξ_{RoB} is 1-c-strict with respect to $\mathcal{A}$. By the definition of 1-c-strictness, we want to show that given $S, T \subseteq V$, $\xi(G, S \cup T, T) \geq \mathbf{E}[\xi(G/\mathcal{A}(S), T, T)]$; here the expectation on the right hand side is over the coins flipped by $\mathcal{A}(S)$.[7] Of course, to compute both the cost shares ξ's, we also have to take expectations. Since the expressions on the left and the right both involve flipping an independent coin for each of the nodes in $S \cup T$, let us couple the two random processes in the natural way by making the same set of coin tosses in both expressions.

Consider a particular choice of coin flips for $S \cup T$, which chooses $F_S \subseteq S$ and $F_T \subseteq T$; set $F = \{r\} \cup F_S \cup F_T$. The cost-shares on the right involve paying for the MST on F_T (in the graph $G/\mathcal{A}(S)$), and paying for connections from each $j \in T \setminus F_T$ to F. Charging for the latter is easy, since we pay for the distance from j to F in the left expression too. To pay for the former, we look at the primal-dual process that generates ξ_{MST}. In the run on $G/\mathcal{A}(S)$ with terminals F_T, a node j in F_T obtains cost-shares as long as its moat does not contain the root of the graph $G/\mathcal{A}(S)$. Since all nodes in F_S are contracted to the root in $G/\mathcal{A}(S)$, in the process for the left hand side the moat of j must not have hit any moat of $F_S \cup \{r\}$, and hence must get at least as much cost-share. This implies that for any particular set of coin flips, the cost-share on the right is bounded above by the cost-share on the left, and hence this holds in expectation as well.

This can be extended to give the following theorem:

Theorem 7. *The Single-Sink Buy-at-Bulk problem has an $O(1)$-approximation algorithm with respect to 1-c-strict cross-monotone cost sharing functions.*

The proof of Theorem 7 extends the proof of Theorem 6. While we defer it until the final version of the paper, we sketch it here: the algorithm is essentially the SimpleSSBB algorithm from [15], which uses the above SimpleCFL algorithm repeatedly to collect the traffic, which is then aggregated at some randomly chosen locations. Each time the aggregation is done using cables of larger capacity, and results in fewer and fewer locations, until finally all the traffic is at one location, whence it is sent to the root. Since we repeatedly use the algorithm SimpleCFL, the cost-share of a node u is just the expected cost-share of u accumulated over

[7] The added expectation sign over the definition (5) is required since $\mathcal{A}$ is randomized.

the various runs of SimpleCFL (where its cost-share is zero when there is no more traffic at u). The proof of strictness again proceeds by coupling the run on $S \cup T$ to the run where we build a solution on S, and then augment it to T.

6 Conclusions

In this paper, we defined several natural "Leasing" problems, in which an optimization problem is solved repeatedly over time (each time with a different set of clients), and the elements chosen to serve the clients can be leased for extended periods of time to take advantage of temporal trends in the sets of clients. The costs of these leases satisfy standard economies of scale, and hence longer leases cost less per unit of time. We study leasing problems in an offline setting, and give approximation algorithms for them via a connection with multistage stochastic optimization. We also give new algorithms for some network design problems in the multistage stochastic framework.

Many future directions of research suggest themselves: an important one is to extend the results to *online* or *stochastic* versions of leasing problems. In this paper, the demands D_t were given up front, but one can also consider cases where the demands D_t appear only on day t, chosen adversarially (i.e., the online model) or from some probability distribution (i.e., the stochastic model). While some of these problems can be solved by solving associated LPs and rounding them online (as in [27]), obtaining general results for these online problems is a direction we are exploring in ongoing work. There seem to be interesting questions involved in pricing these leases as well. It would be good to extend the "buy-at-bulk" results to cases where the cost function is not separable $g(b)f(\ell)$. Finally, getting $o(k)$-approximations for the Steiner Tree Leasing problem is an intriguing question—it seems that the ideas for such an improvement would be useful for the multistage stochastic versions as well.

References

1. Andrews, M., Zhang, L.: Wavelength assignment in optical networks with fixed fiber capacity. In: 31st ICALP. Volume 3142 of LNCS. (2004) 134–145
2. Awerbuch, B., Azar, Y.: Buy-at-bulk network design. In: 38th FOCS. (1997) 542–547
3. Awerbuch, B., Azar, Y., Bartal, Y.: On-line generalized Steiner problem. Theoret. Comput. Sci. **324**(2-3) (2004) 313–324
4. Balakrishnan, A., Magnanti, T.L., Mirchandani, P.: Network design. In Dell'Amico, M., Maffioli, F., Martello, S., eds.: Annotated bibliographies in combinatorial optimization. John Wiley & Sons Ltd., Chichester (1997) 311–334
5. Bartal, Y.: Probabilistic approximations of metric spaces and its algorithmic applications. In: 37th FOCS. (1996) 184–193
6. Carr, R., Fleischer, L., Leung, V., Phillips, C.: Strengthening integrality gaps for capacitated network design and covering problems. In: 11th SODA. (2000) 106–115
7. Charikar, M., Chekuri, C., Pál, M.: Sampling bounds for stochastic optimization. In: 9th APPROX. Volume 3624 of LNCS. Springer, Berlin (2005) 257–269

8. Dhamdhere, K., Ravi, R., Singh, M.: On two-stage stochastic minimum spanning trees. In: IPCO. Volume 3509 of LNCS. Springer, Berlin (2005) 321–334
9. Dye, S., Stougie, L., Tomasgard, A.: The stochastic single resource service-provision problem. Naval Research Logistics **50**(8) (2003) 869–887
10. Eisenbrand, F., Grandoni, F., Oriolo, G., Skutella, M.: New approaches for virtual private network design. In: 32nd ICALP. Volume 3580 of LNCS. (2005) 1151–1162
11. Fakcharoenphol, J., Rao, S., Talwar, K.: A tight bound on approximating arbitrary metrics by tree metrics. J. Comput. System Sci. **69**(3) (2004) 485–497
12. Garg, N., Khandekar, R., Konjevod, G., Ravi, R., Salman, F.S., Sinha, A.: On the integrality gap of a natural formulation of the single-sink buy-at-bulk network design formulation. In: 8th IPCO. Volume 2081 of LNCS. (2001) 170–184
13. Goyal, S., Giri, B.C.: Recent trends in modeling of deteriorating inventory. European Journal of Operational Research **134**(1) (2001) 1–16
14. Guha, S., Meyerson, A., Munagala, K.: Hierarchical placement and network design problems. In: 41th FOCS. (2000) 603–612
15. Gupta, A., Kumar, A., Roughgarden, T.: Simpler and better approximation algorithms for network design. In: 35th STOC. (2003) 365–372
16. Gupta, A., Pál, M., Ravi, R., Sinha, A.: Boosted sampling: Approximation algorithms for stochastic optimization problems. In: 36th STOC. (2004) 417–426
17. Gupta, A., Pál, M., Ravi, R., Sinha, A.: What about Wednesday? approximation algorithms for multistage stochastic optimization. In: 8th APPROX. (2005) 86–98
18. Gupta, A., Srinivasan, A., Tardos, É.: Cost-sharing mechanisms for network design. In: 7th APPROX. Volume 3122 of LNCS. (2004) 139–150
19. Hayrapetyan, A., Swamy, C., Tardos, E.: Network design for information networks. In: 16th SODA. (2005) 933–942
20. Immorlica, N., Karger, D., Minkoff, M., Mirrokni, V.: On the costs and benefits of procrastination: Approximation algorithms for stochastic combinatorial optimization problems. In: 15th SODA. (2004) 684–693
21. Jain, K., Vazirani, V.V.: Equitable cost allocations via primal-dual-type algorithms. In: 34th STOC, ACM Press (2002) 313–321
22. Kent, K.J., Skorin-Kapov, D.: Population monotonic cost allocations on MSTs. In: Proceedings of the 6th International Conference on Operational Research (Rovinj, 1996), Croatian Oper. Res. Soc., Zagreb (1996) 43–48
23. Kolliopoulos, S.G., Young, N.E.: Tight approximation results for general covering integer programs. In: 42nd FOCS. (2001) 522–528
24. Leonardi, S., Schäfer, G.: Cross-monotonic cost sharing methods for connected facility location games. Theoret. Comput. Sci. **326**(1-3) (2004) 431–442
25. Magnanti, T.L., Wong, R.T.: Network design and transportation planning: Models and algorithms. Transportation Science **18** (1984) 1–55
26. Magnanti, T.L., Mirchandani, P., Vachani, R.: Modeling and solving the two-facility capacitated network loading problem. Oper. Res. **43**(1) (1995) 142–157
27. Meyerson, A.: The parking permit problem. In: 46th FOCS. (2005) 274–284
28. Munagala, K. personal communication
29. Nahmias, S.: Perishable inventory theory: A review. Operations Research **30**(4) (1982) 680–708
30. Raghavan, P., Thompson, C.D.: Randomized rounding: a technique for provably good algorithms and algorithmic proofs. Combinatorica **7**(4) (1987) 365–374
31. Ravi, R., Sinha, A.: Hedging uncertainty: Approximation algorithms for stochastic optimization problems. In: 10th IPCO. (2004) 101–115

32. Salman, F.S., Cheriyan, J., Ravi, R., Subramanian, S.: Approximating the single-sink link-installation problem in network design. SIAM J. Optimization **11**(3) (2000) 595–610
33. Shmoys, D., Swamy, C.: Stochastic optimization is (almost) as easy as deterministic optimization. In: 45th FOCS. (2004) 228–237
34. Shmoys, D., Swamy, C.: Sampling-based approximation algorithms for multi-stage stochastic optimization. In: 46th FOCS. (2005) 357–366
35. Srinivasan, A.: Improved approximation guarantees for packing and covering integer programs. SIAM J. Comput. **29**(2) (1999) 648–670
36. Srinivasan, A.: New approaches to covering and packing problems. In: 12th SODA. (2001) 567–576
37. Srinivasan, A.: Approximation algorithms for stochastic and risk-averse optimization. In: 18th SODA. (2007) 1305–1313
38. Talwar, K.: Single-sink buy-at-bulk LP has constant integrality gap. In: 9th IPCO. Volume 2337 of LNCS. (2002) 475–486

A Cost Shares and Stochastic Algorithms

We will draw on some techniques developed in recent work on converting approximation algorithms for standard (non-stochastic) versions of optimization problems into those for the stochastic versions of the problems [16,17]. In particular, we use the following theorem.

Theorem 8 ([17]). *Given a problem Π, if $\mathcal{A}$ is an α-approximation algorithm w.r.t. a 1-c-strict cost-sharing function ξ, and if ξ is cross-monotone, then there is an αk-approximation algorithm for the k-stage stochastic version of Π.*

Let us briefly discuss the basics of the cost-sharing concepts we will use in this paper; we refer the reader to [17] for a detailed discussion of the concepts. Loosely, a *cost-sharing* function ξ divides the cost of a solution among the client set S. We use the notation $\xi(G, S, j)$ to denote the share of the client j when the input is the graph G and the set of clients is S. (By convention, we will assume that $\xi(G, S, j) > 0 \implies j \in S$.) The function ξ is *cross-monotone* if for every pair of client sets $S \subseteq T$ and a client j such that $j \in S$, we have $\xi(G, T, j) \leq \xi(G, S, j)$. (I.e., if more clients join the system, the share of any individual client does not increase.)

Competitiveness. We will try to relate algorithms $\mathcal{A}$ to cost-sharing functions ξ, and hence ξ will conceptually behave like a "dual". Hence a crucial property is that ξ give a lower bound on the cost of the optimal solution: A cost-sharing function ξ is *competitive* if for every client set S, it holds that

$$\textstyle\sum_{j \in S} \xi(G, S, j) \leq \mathsf{OPT}(X, S). \tag{4}$$

We will focus only on competitive cost-sharing functions. (Henceforth, we will use the notation $\xi(G, S, S')$ to denote $\sum_{j \in S'} \xi(G, S, j)$.)

Strictness. Let $S, T \subseteq V$ be sets of users. Suppose G is the original graph, and $G/\mathcal{A}(G,S)$ is the graph after the client set S has already been served by running the algorithm $\mathcal{A}$ on it. Then the cost-sharing function ξ is β-c-strict if

$$\xi(G/\mathcal{A}(G,S), T, T) \leq \beta \times \xi(G, S \cup T, T). \tag{5}$$

In other words, the total cost shares for the set T of users in the reduced instance $G/\mathcal{A}(G,S)$ is at most β times the cost-shares for T if the users in S were present as well. In this paper, we will deal only with the case when $\beta = 1$; i.e., cases where the cost shares for T when it appears with S are at least as much as when S is served earlier, and then T has to be served by itself.

Finally, we call $\mathcal{A}$ *an α-approximation algorithm with respect to the cost-sharing function ξ*

$$c(\mathcal{A}(G,S)) \leq \alpha\, \xi(G,S,S). \tag{6}$$

Note that chaining the inequalities (6) and (4) implies that $\mathcal{A}$ is an α-approximation algorithm in the conventional sense as well.