LAST but not Least: Online Spanners for Buy-at-Bulk*

Anupam Gupta† R. Ravi‡ Kunal Talwar§ Seeun William Umboh¶

Abstract

The online (uniform) buy-at-bulk network design problem asks us to design a network, where the edge-costs exhibit economy-of-scale. Previous approaches to this problem used tree-embeddings, giving us randomized algorithms. Moreover, the optimal results with a logarithmic competitive ratio requires the metric on which the network is being built to be known up-front; the competitive ratios then depend on the size of this metric (which could be much larger than the number of terminals that arrive).

We consider the buy-at-bulk problem in the least restrictive model where the metric is not known in advance, but revealed in parts along with the demand points seeking connectivity arriving online. For the single sink buy-at-bulk problem, we give a deterministic online algorithm with competitive ratio that is logarithmic in k, the number of terminals that have arrived, matching the lower bound known even for the online Steiner tree problem. In the *oblivious* case when the buy-at-bulk function used to compute the edge-costs of the network is not known in advance (but is the same across all edges), we give a deterministic algorithm with competitive ratio polylogarithmic in k, the number of terminals.

At the heart of our algorithms are optimal constructions for online Light Approximate Shortest-path Trees (LASTs) and spanners, and their variants. We give constructions that have optimal trade-offs in terms of cost and stretch. We also define and give constructions for a new notion of LASTs where the set of roots (in addition to the points) expands over time. We expect these techniques will find applications in other online network-design problems.

1 Introduction

The model of (uniform) buy-at-bulk network design captures economies-of-scale in routing problems. Given an undirected graph $G = (V, E)$ with edge lengths $d : E \to R_{\geq 0}$—we can assume the lengths form a metric—the cost of sending x_e flow over any edge e is $d(e) \cdot f(x_e)$ where f is some concave cost function. The total cost is the sum over all edges of the per-edge cost. Given some traffic matrix (a.k.a. demand), the goal is now to find a routing for the demand to minimize the total cost. This model is well studied both in the operations research and approximation algorithms communities, both in the offline and online settings. In the offline setting, an early result was an $O(\log k)$-approximation due to Awerbuch and Azar, one of the first uses of tree embeddings in approximation algorithms [AA97]—here k is the number of demands. For the single-sink case, the first $O(1)$-approximation was given by [GMM09]. In fact, one can get a constant-factor even for the "oblivious" single-sink case where the demands are given, but the actual concave function f is known only after the network is built [GP12].

The problem is just as interesting in the *online context*: in the online single-sink problem, new demand locations (called *terminals*) are added over time, and these must be connected to the central root node as they arrive. This captures an increasing demand for telecommunication services as new customers arrive, and must be connected via access networks to a central core of nodes already provisioned for high bandwidth. The Awerbuch-Azar approach of embedding G into a tree metric T with $O(\log n)$ expected stretch (say using [FRT04]), and then routing on this tree, gives an $O(\log n)$-competitive randomized algorithm even in the online case. But *this requires that the metric is known in advance*, and the dependence is on n, the number of nodes in the metric, and not on the number of terminals k! This may be undesirable in situations when $n \gg k$; for example, when the terminals come from a Euclidean

*The full version of this paper can be found at: `https://arxiv.org/abs/1611.00052`

†Computer Science Department, Carnegie Mellon University, Pittsburgh, PA 15213, USA. Research partly supported by NSF awards CCF-1319811, CCF-1540541 and CCF-1617790.

‡Tepper School of Business, Carnegie Mellon University, USA, ravi@cmu.edu; This material is based upon research supported in part by the U. S. Office of Naval Research under award number N00014-12-1-1001, and the U. S. National Science Foundation under award number CCF-1527032.

§Google Research

¶Department of Mathematics and Computer Science, Eindhoven University of Technology, Netherlands. Email: `seeun.umboh@gmail.com`. This work was supported in part by ERC consolidator grant 617951 and NSF grant CCF-1320854. Part of this work was done while a student at the University of Wisconsin - Madison, and while visiting the Simons Institute for the Theory of Computing.

space $\mathbb{R}^d$ for some large d. Moreover, we only get a *randomized* algorithm (competitive against oblivious adversaries).[1]

In this paper, we study the Buy-at-Bulk (BaB) problem in the online setting, in the least restrictive model where the metric is not known in advance, so the distance from some point to the previous points is revealed only when the point arrives. This forces us to focus on the problem structure, since we cannot rely on powerful general techniques like tree embeddings. Moreover, we aim for deterministic algorithms for the problem. Our first main result is an asymptotically optimal deterministic online algorithm for single-sink buy-at-bulk.

THEOREM 1.1. (DETERMINISTIC BAB) *There exists a deterministic $O(\log k)$-competitive algorithm for online single-sink buy-at-bulk, where k is the number of terminals.*

Note that the guarantee is best possible, since it matches the lower bound [IW91] for the special case of a single cable type encoding the online Steiner tree problem.

En route, we consider a generalization of the *Light Approximate Shortest-path Trees* (LASTs). Given a set of "sources" and a sink, a LAST is a tree of weight close to the minimum spanning tree (MST) on the sources and the sink, such that the tree distance from any source to the sink is close to the shortest-path distance. Khuller, Raghavachari, and Young [KRY95] defined and studied LASTs in the offline setting and showed that one can get constant stretch with cost constant times the MST. Ever since their work, LASTs have proved very versatile in network design applications. We give (in the full version) a simple construction of LASTs in the online setting where terminals arrive online. We get constant stretch, and cost at most $O(\log k)$ times the MST, which is the best possible in the online case.

For our algorithms, we extend the notion of LASTs to the setting of MLASTs (*Multi-sink LASTs*) where both sources *and sinks* arrive over time. We have to maintain a set of edges so that every source preserves its distance to the closest sink arriving before it, at minimum total cost. We provide a tight deterministic online algorithm also for MLASTs, which we think is of independent interest. This construction appears in §3. Then we use MLASTs to prove Theorem 1.1 in §4.

Oblivious Buy-at-Bulk. We then change our focus to the *oblivious* problem. Here we are given neither the terminals nor the buy-at-bulk function f in advance. When the terminals arrive, we have to choose paths for them to the root, so that for every concave cost function f, the chosen routes are competitive to the optimal solution for f. Our first result for this problem is the following:

THEOREM 1.2. (OBLIVIOUS BAB, RANDOMIZED) *There exists a randomized online algorithm for the buy-at-bulk problem that produces a routing P such that for all concave functions f,*

$$\mathbb{E}\big[\operatorname{cost}_f(P)\big] \le O(\log^2 k)\,\mathrm{OPT}_f\,.$$

This randomized algorithm has the same approximation guarantee as one obtained using Bartal's tree-embedding technique. The benefit of this result, however, is in the ideas behind it. We give constructions of low-stretch spanners in the online setting. Like LASTs, spanners have been very widely studied in the offline case; here we show how to maintain light low-stretch spanners in the online setting. Then we use the spanners to prove the above theorem. Moreover, building on these ideas, we give a deterministic algorithm. (We defer our spanner construction and oblivious algorithms to the full version.)

THEOREM 1.3. (OBLIVIOUS BAB, DETERMINISTIC) *There exists a deterministic online algorithm for the buy-at-bulk problem that produces a routing P such that for all concave functions f,*

$$\operatorname{cost}_f(P) \le O(\log^{2.5} k)\,\mathrm{OPT}_f\,.$$

A question that remains open is whether there is an $O(\log k)$-competitive algorithm for this problem. The only other deterministic oblivious algorithm we know for the buy-at-bulk problem is a derandomization of the oblivious network design algorithm from [GHR06], which requires the metric to be given in advance.

LASTs and Spanners. A central contribution of our work is to demonstrate the utility of online spanners in building networks exploiting economies-of-scale. We record the following two theorems on maintaining LASTs and spanners in an online setting, since they are of broader interest. These results are near-optimal, as we discuss in the respective sections (in the full version).

THEOREM 1.4. (ONLINE LAST) *There exists a deterministic online algorithm for maintaining a tree with cost $O(\log k)$ times the MST, and a stretch of 7 for distances from terminals to the sink.*

[1] The tree embeddings of Bartal [Bar96] can indeed be done online with $O(\log k \log \Delta)$ expected stretch, where Δ is the ratio of maximum to minimum distances in the metric. Essentially, this is because the probabilistic partitions used to construct the embedding can be computed online. This gives an $O(\log^2 k)$-competitive randomized algorithm, alas sub-optimal by a logarithmic factor, and still randomized.

THEOREM 1.5. (ONLINE SPANNER) *There exists a deterministic online algorithm that given k pairs of terminals, maintains a forest with cost $O(\log k)$ times the Steiner forest on the pairs, and a stretch of $O(\log k)$ for distances between all given pairs of terminals. Moreover, the total number of edges in the forest is $O(k)$, i.e. linear in the number of terminals.*

Our results on online LAST and multicommodity spanners give us an optimal $O(\log k)$-competitive deterministic algorithms for the "single cable version" of the buy-at-bulk problems, where the function is a single-piece affine concave function [MP98].

1.1 Our Techniques We now outline some of the ideas behind our algorithms and the role of spanner constructions in them. All our algorithms share the same high-level framework: (1) each terminal v is assigned an integer *type* $\tau(v)$; (2) it is then routed through a sequence of terminals with increasing types; (3) the routes are chosen using a spanner-type construction. In particular, each algorithm is specified by a *type-selection rule*, the sequence of terminals to route through, and whether to use a spanner or MLAST.

The analysis for our deterministic algorithms also has a common thread: while these algorithms are not based on tree-embeddings, the analysis uses tree-embeddings crucially. Both analyses are based on the analysis framework of [Umb15] and follow the same template: we fix an HST embedding T of the metric, and charge the cost of the algorithm to the cost of the optimal solution on this tree (losing some factor α). Since HSTs can approximate general metrics to within a factor of $O(\log k)$, this gives us an $O(\alpha \log k)$-competitive algorithm.

Functions vs. Cables. As is common with buy-at-bulk algorithms, we represent the function $f(x)$ as the minimum of affine functions $\min_i\{\sigma_i + \beta_i x\}$. And when we route a path, we even specify which of the linear functions we use on each edge of this path. Each i is called a "cable type", since one can think of putting down cable-i with an up-front cost of σ_i, and then paying β_i for every unit of flow over it.

Non-oblivious algorithm. In the non-oblivious setting where we know the function f, we will want to route each terminal v through a path $P(v)$ with non-decreasing cable types. So $\tau(v)$ should simply be the cable of lowest type that we will install on $P(v)$—how should we choose this value? It makes sense to choose $\tau(v)$ based on the number of other terminals close to v. Intuitively, the more terminals that are nearby, the larger the flow that can be aggregated at v, making it natural to select a larger type for v.

Once we have chosen the type, the route selection is straightforward: first we route v's demand to the nearest terminal of higher type using cable type $\tau(v)$. Then we iterate: while v's demand is at a terminal w (which is not the root r), we route it to a terminal w' of type higher than $\tau(w)$ that nearest to w, using cable type $\tau(w)$.

Finally, how do we select the path when routing v's demand from w to w'? This is where our Multi-sink LAST (MLAST) construction comes in handy. We want that for each cable type i, the set of edges H_i on which we install cable type i has a small total cost while ensuring that each terminal of type i has a short path to its nearest terminal of higher type. We can achieve these properties by having H_i be an MLAST with the sources being the terminals of type exactly i, and the sinks being terminals of type higher than i.

Randomized Oblivious Algorithm. While designing oblivious algorithms seems like a big challenge because we have to be simultaneously competitive for all concave cost functions, Goel and Estrin [GE05] showed that functions of the form $g_i(x) = \min\{x, 2^i\}$ form a "basis" and hence we just have to be good against all these so-called "rent-or-buy" functions. This is precisely our goal.

Note that the optimum solution for the cost function g_0 is the optimal Steiner tree, and that for the cost function g_M, for $M \gg k$, is the shortest path tree rooted at the sink. Thus being competitive against g_0 and g_M already requires us to build a LAST. Thus it is not surprising that our online spanner algorithm is a crucial ingredient in our algorithm.

There are two key ideas. The first is that to approximate a given rent-or-buy function, it suffices to figure out which terminals should be connected to the root via "buy" edges—i.e., via edges of cost 2^i regardless of the load on them—these terminals we call the *"buy" terminals*. The rest of the terminals are simply connected to the buy terminals via shortest paths. One way to choose a good set of buy terminals is by random sampling: if we wanted to be competitive against function g_i, we could choose each terminal to be a "buy" terminal with probability 2^{-i}. (See, e.g., [AAB04].) Since in the oblivious case we don't know which function g_i we are facing, we have to hedge our bets. Hence we choose each terminal v to have $\tau(v) = i$ with probability 2^{-i}. Thus terminal v is a good buy terminal for all g_i with $i \leq \tau(v)$.

Next, we need to ensure that the path $P(v)$ we choose for v simultaneously approximates the shortest path from v to the set of terminals with type at least i for all i. This is where our online spanner construction comes in handy. For each type i, we will build a spanner F_i on terminals of type at least i.

Deterministic Oblivious Algorithm. To obtain our deterministic oblivious algorithm, we make two further modifications. First, we remove the need for randomness by using the deterministic type selection rule of the non-oblivious algorithm. This modification already yields an $O(\log^3 k)$-competitive algorithm. A technical difficulty that arises is that our type-classes are no longer nested. Thus it is not obvious how to route from a node of type i to one of a higher type, as these nodes may not belong to a common spanner. Adding nodes to multiple spanners can remedy this, but leads to a higher buy cost; being stingy in adding nodes to spanners can lead to higher rent cost. By carefully balancing these two effects and using a more sophisticated routing scheme, we are able to improve the competitive ratio to $O(\log^{2.5} k)$.

1.2 Other Related Work

Offline approximation algorithms for the (uniform) buy-at-bulk network design problem were initiated in [SCRS97]—here *uniform* denotes the fact that all edges have the same cost function $f(\cdot)$, up to scaling by the length of the edge. Early results for approximation algorithms for buy-at-bulk network design [MP98, AA97] already observed the relationship to spanners, and tree embeddings. Using the notion of probabilistic embedding of arbitrary metrics into tree metrics [AKPW95, Bar96, Bar98, Bar04, FRT04], logarithmic factor approximations are readily derived for the buy-at-bulk problem in the offline setting. A hardness result of $O(\log^{1/4-\varepsilon} n)$ shows we cannot do much better in the worst case [And04].

For the offline single-sink case, new techniques were developed to get $O(1)$-approximations [GMM09], as well as to prove $O(1)$-integrality gaps for natural LP formulation [GKK+01, Tal02]; other algorithms have been given by [GKR03, GI06, JR09]. Apart from its inherent interest, the single-sink buy-at-bulk problem can also be used to solve other network design problems [GRS11]. The oblivious single-sink version was first studied in [GE05], and $O(1)$-approximations for this very general version was derived in [GP12].

In the setting of online algorithms, the Steiner tree and generalized Steiner forest problems have tight $O(\log k)$-competitive algorithms [BC97, IW91], where k is the number of terminals. These algorithms work in the model where new terminals only reveal their distances to previous terminals when they arrive, and the metric is not known *a priori*. It is well-known that the tree-embedding result of Bartal [Bar96] can be implemented online to give an $O(\log k \cdot \min(\log \Delta, \log k))$-competitive algorithm for the online single-sink oblivious buy-at-bulk problem, where Δ is the ratio of maximum to minimum distances in the metric. For online rent-or-buy, Awerbuch, Azar, and Bartal [AAB04] gave an $O(\log^2 k)$-deterministic and an $O(\log k)$-randomized algorithm; recently, [Umb15] gave an $O(\log k)$-deterministic algorithm.

If the metric is known a priori, the results depend on n, the size of the metric, and not on k, the number of terminals.[2] E.g., tree-embedding results of [FRT04] give a randomized $O(\log n)$-competitive algorithm, or a derandomization of oblivious network design from [GHR06] gives an $O(\log^2 n)$-competitive algorithm.

A generalization which we do not study here is the *non-uniform* buy-at-bulk problem, where we can specify a different concave function on each edge. A polylogarithmic approximation for this problem was recently given by Ene et al. [ECKP15]; see the references therein for the rich body of prior work.

2 Preliminaries

Formally, in the online buy-at-bulk problem, we have a complete graph $G = (V, E)$, and edge lengths $d(e)$ satisfying the triangle inequality. In other words, we can treat (V, d) as a metric. We have M *cable types*. The i-th cable type has *fixed cost* σ_i and *incremental cost* β_i, with $\sigma_i > \sigma_{i-1}$ and $\beta_i < \beta_{i-1}$. Routing x units of demand through cable type i on some edge e costs $(\sigma_i + \beta_i x)d(e)$.

In the single-sink version, we are given a root vertex $r \in V$. Initially, no cables are installed on any edges. When a terminal v arrives, we install some cables on some edges and choose a path $P(v)$ on which to route v's unit demand. (This routing has to be unsplittable, i.e., along a single path.) We are allowed to install multiple cables on the same edge; if $e \in P(v)$ has multiple cables installed on it, v's demand is routed on the one with highest type, i.e., the one with least incremental cost. The choice of $P(v)$ and cable installations are irrevocable. Given a routing solution, if $\mathrm{load}_i(e)$ is the total amount of demand routed through cable type i on edge e, the total cost is $\sum_{e\in E}\left[\sum_{i:\mathrm{load}_i(e)>0} \sigma_i + \beta_i \,\mathrm{load}_i(e)\right] d(e)$. We call $\sum_{e\in E}\sum_{i:\mathrm{load}_i(e)>0} \sigma_i d(e)$ the *fixed cost* of the solution and $\sum_{e\in E}\sum_{i:\mathrm{load}_i(e)>0} \beta_i \,\mathrm{load}_i(e) d(e)$ the *incremental cost* of the solution.

Let OPT denote the cost of the optimal solution. We assume the cable costs satisfy the conditions of the following Lemma 2.1.

LEMMA 2.1. *[GMM09] We can prune the set of cables such that for all the retained cable types i, (a) $\sigma_i \geq 3\sigma_{i-1}$, and (b) $\beta_i \leq (1/3^2)\beta_{i-1}$, so that the cost of any*

[2] Note that k is at most n, and it can be much smaller.

solution using only the pruned cable types increases only by an $O(1)$ factor.

2.1 HST embeddings Let (X, d) be a metric over a set of points X with distances d at least 1.

Definition 2.1. (HST embeddings [Bar96]) *A hierarchically separated tree (HST) embedding T of metric (X, d) is a rooted tree with height $\lceil \log_2(\max_{u,v \in X} d(u, v)) \rceil$ and edge lengths that are powers of 2 satisfying:*

a. *The leaves of T are exactly X.*
b. *The length of the edge from any node to each of its children is the same.*
c. *The edge lengths decrease by a factor of 2 as one moves along any root-to-leaf path.*

For $e \in T$, we use $T(e)$ to denote the length of e and say that e is a level-j edge *if $T(e) = 2^j$. Furthermore, we write $T(u, v)$ to denote the distance between u and v in T. We also use $L(e)$ denote the leaves that are "downstream" of e, i.e. they are the leaves that are separated from the root of T when e is removed from T.*

The crucial property of HST embeddings that we will exploit in our analyses is the following proposition, which follows directly from Properties (2) and (3) of Definition 2.1.

Proposition 2.1. *Let T be a HST embedding T of (X, d). For any level-j edge $e \in T$, we have $d(u, v) < 2^j$ for any $u, v \in L(e)$.*

Corollary 2.1. ([AA97, FRT04]) *For any online buy-at-bulk instance with k terminals X and distances d, there exists a HST embedding T of (X, d) such that $\mathrm{OPT}(T) \le O(\log k)\,\mathrm{OPT}$, where $\mathrm{OPT}(T)$ is the cost of the optimal solution for online buy-at-bulk with terminals X in the tree T.*

2.2 Decomposition into rent-or-buy instances Since buy-at-bulk functions can be complicated, it will be useful to deal with simpler *rent-or-buy* functions where to route a load of x on an edge, we can either *buy* the cable for unlimited use at its buy cost, or pay the rental cost times the amount x. Given a buy-at-bulk instance as above, for each i, define the rent-or-buy instance with the rent-or-buy function $f_i(x) = \min\{\sigma_i, \beta_{i-1}x\}$. Let OPT_i to be the cost of the optimum solution with respect to this function f_i. Note that under this function when the load aggregates up to $\frac{\sigma_i}{\beta_{i-1}}$, it becomes advantageous to switch from renting to buying.

The following lemma will prove very useful, since we can charge different parts of our cost to different $\mathrm{OPT}_i(T)$s for some HST T, and then sum them up to argue that the total cost is $O(1)\,\mathrm{OPT}(T)$, and hence $O(\log k)\,\mathrm{OPT}$ using Corollary 2.1.

Lemma 2.2. (OPT Decomposition on Trees) *For every tree T, we have $\sum_i \mathrm{OPT}_i(T) \le O(1)\,\mathrm{OPT}(T)$.*

Proof. For an edge $e \in T$, let $\phi_i^*(e)$ and $\phi^*(e)$ be the costs incurred by $\mathrm{OPT}_i(T)$, and $\mathrm{OPT}(T)$ respectively, on e. We will show that for every edge $e \in T$, we have $\sum_i \phi_i^*(e) \le O(1)\phi^*(e)$. Let $X(e)$ be the set of terminal-pairs whose tree paths (in the single-sink case, terminals whose paths to r) in T include e, and $T(e)$ denote the length of the tree edge e. So, $\phi_i^*(e) = T(e) \cdot \min\{\sigma_i, \beta_{i-1}|X(e)|\}$ and $\phi^*(e) = T(e) \cdot \min_i\{\sigma_i + \beta_i|X(e)|\}$. For any fixed j, we have

$$\begin{aligned}\sum_i \min\{\sigma_i, \beta_{i-1}|X(e)|\} &\le \sum_{i \le j} \sigma_i + \sum_{i > j} \beta_{i-1}|X(e)| \\ &\le O(1)(\sigma_j + \beta_j|X(e)|),\end{aligned}$$

where the last inequality follows from the fact that the fixed costs σ_i increase geometrically with i and the incremental costs β_i decrease geometrically with i, as assumed in Lemma 2.1.

Thus, $\sum_i \phi_i^*(e) \le O(1)\phi^*(e)$ for every edge $e \in T$ and so $\sum_i \mathrm{OPT}_i(T) \le O(1)\,\mathrm{OPT}(T)$. □

The next lemma proves that the rent-or-buy functions form a "basis" for buy-at-bulk functions. For $i \in \mathbb{Z}_{\ge 0}$, define the rent-or-buy function $g_i(x) = \min\{x, 2^i\}$. (See, e.g., [GP12, Section 2].)

Lemma 2.3. (Basis of Rent-or-Buy Functions) *Fix some routing of demands. If for every i, the cost of this routing under g_i is within a factor of ρ of the optimal routing for g_i, then for any monotone, concave function f with $f(0) = 0$, the cost of the routing under f is within $O(\rho)$ of the optimal cost of routing under f.*

3 Multi-Sink LASTs

Recall that LASTs were Light Approximate-Shortest-path Trees, i.e., trees where we maintain the shortest-path distance of sources to a sink, using a light tree. (This was traditionally done offline, though the reader is not required to know the offline construction for this section.) In this section, we are interested in the *multi-sink LAST* (MLAST) problem. The input is a sequence of terminals in the underlying metric G, each of which is either a *source* or a *sink* (we assume the first terminal is always a sink), and our algorithm has to maintain a subgraph H such that (a) the distance of any source to its closest sink in H should be comparable to the distance of that source to its closest sink in G, and moreover (b) the cost $d(H)$ should not be "too large".

Note that when a new sink s arrives, it may be close to many sources, and hence we may need to add "shortcut" edges to reduce their distance to s. Moreover, what does it mean for the cost of H to not be "too large", since the distance from a source to its closest sink can fall dramatically over time. Our notion is the following. Note that when a *source* v arrives, it has to pay at least the distance to its closest terminal (source or sink) at that time, just to maintain connectivity. Loosely, we want our cost $d(H)$ to be not much more than the sum of these distances. (N.b.: this is the intuition, formally we will pay $O(2^{\mathrm{class}(v)})$ which will be defined soon.)

3.1 The Algorithm The first terminal to arrive is a sink we denote as $s^\star$. We use R and S to denote the sets of sources and sinks that have arrived. For a terminal u, let $R(u)$ and $S(u)$ be the sources and sinks that arrived strictly before u. Our algorithm Mlast-Alg maintains a subgraph H that consists of two parts: a forest F which is a "backbone" connecting each source to some sink cheaply, and an edge set A which "augments" F to ensure that each source is not too far from the sinks in $H = A \cup F$.

The forest F is constructed using nets which we define now. Let $\Delta > 0$ be some distance scale. Then, a Δ-*net* Z is a subset of terminals (called *net points*) such that every terminal v has $d(v, Z) < \Delta$ and for every pair of net points $u, v \in Z$, we have $d(u, v) \geq \Delta$. Initially, both F and A are empty. The algorithm maintains a 2^j-net Z_j on the entire set of terminals, for each distance scale $j \in \mathbb{Z}$. When a terminal v arrives, for every distance scale j, it is added to Z_j if $d(u, Z_j) \geq 2^j$. The *class* of v is defined to be $\mathrm{class}(v) := \max\{j : v \in Z_j\}$, i.e., $\mathrm{class}(v)$ is the largest distance scale such that v belongs to the net of that scale. Hence the class of the first terminal $s^\star$ is ∞.

Let u be the nearest terminal in $\bigcup_{j:j>\mathrm{class}(v)} Z_j$, i.e., u is the nearest net-point lying in any net at a higher scale. If the current vertex v is a source, then the edge (u, v) is added to F; otherwise, F is left untouched. Next, the algorithm goes through every source u (including v if it is also a source), checks if $d_{A\cup F}(u, S) \leq 3d(u, S)$, and otherwise adds the edge (u, u') to A where $u' \in S$ is the nearest sink to u. (See Algorithm 1.)

High-Level Idea of the Analysis. We defer the proofs to the full version. Here, are the main properties we will use in the remainder of the paper.

Algorithm 1 Mlast-Alg(R, S)

$Z_j \leftarrow \emptyset$ for all j, $A \leftarrow \emptyset$, $F \leftarrow \emptyset$, $\Delta = 0$;
while terminal v arrives **do**
  If $d(v, S(v)) > \Delta$, set $\Delta \leftarrow d(v, S(v))$
  for all $j \in \mathbb{Z}$ **do**
    if $d(v, Z_j) \geq 2^j$ **then**
      Add v to Z_j
  $\mathrm{class}(v) \leftarrow \max\{j : v \in Z_j\}$
  if $v \in R$ **then**
    Add edge (u, v) to F where u is a terminal of higher class nearest to v
  for $x \in R$ **do**
    if $d_{A\cup F}(x, S) > 3d(x, S)$ **then**
      Add (x, x') to A where x' is the terminal in S nearest to u

Lemma 3.1. *Given an online sequence of sources R and sinks S, the algorithm* Mlast-Alg *maintains a subgraph H and assigns classes to terminals such that*

a. $d(H) \leq O(1) \sum_{v\in R} 2^{\mathrm{class}(v)}$,
b. $d_H(u, S) \leq 3d(u, S)$ *for every* $u \in R$,
c. *if* $\mathrm{class}(u) = \mathrm{class}(v) = j$, *then* $d(u, v) \geq 2^j$,
d. $d(u, R(u) \cup S(u)) \leq 2^{\mathrm{class}(u)} \leq d(u, s^\star)$ *for every* $u \in R$, *where* $s^\star$ *is the first terminal that arrived, and* $R(u)$ *and* $S(u)$ *are the sources and sinks that arrived before* u.

Property (a) is the formal bound on the cost of the MLAST. One should think of $2^{\mathrm{class}(v)}$ as being the radius of some "dual" ball around source v, that we will later use to give lower bounds on our buy-at-bulk instances. Property (b) guarantees distance preservation. Properties (c) and (d) ensure that terminals of the same class are well-separated, and that $\mathrm{class}(u)$ is not too large. Out of these, property (a) is the most non-trivial one to prove. The cost of F is easy to bound, since each source v adds in a single edge of length at most $O(2^{\mathrm{class}(v)})$. For the cost of edges in A, the argument at a high level is that the sources adding two different edges of the same length must be far from each other in H (compared to the length of the edges added) else the later one could use a path to the earlier one, and then the added edge, to fulfill the distance guarantee.

4 Non-Oblivious Buy at Bulk

In this section we prove our main result for online buy-at-bulk for the case when the function f is known.

Theorem 4.1. (Deterministic BaB) *There exists a deterministic $O(\log k)$-competitive algorithm for online single-sink buy-at-bulk, where k is the number of terminals.*

For a high-level intuition behind the algorithm, see the discussion in §1.1. Here we explain precisely how to assign types, update layers and route demands.

Type assignment. Consider a terminal v. At the time of its arrival, for each type i, let $d_i(v)$ be the distance to the nearest type-i terminal, and define the ball $B_i(v) := \{u : d(u,v) \le d_i(v)/2^3\}$. Let $n_i(v) := |B_i(v)|$ be the number of terminals in this ball. Terminal v is assigned type

$$\tau(v) := \max\left\{i : n_i(v) \ge \tfrac{\sigma_i}{\beta_{i-1}}\right\}.$$

To make sense of this, observe that the threshold for being of a certain type is the same as the threshold for the load at which it is advantageous to buy rather than rent for that type. This means vertices assigned a certain type can collect enough "witnesses" whose paths to the root in the optimal solution can be used to pay for edges constructed by v in the online solution.

Layers. Let R_i be the current set of terminals of type exactly i, and $X_i := \cup_{i' \ge i} R_{i'}$ be the current set of terminals of type at least i. Let $X_i(v)$ (resp., $R_i(v)$) be the set of terminals of type at least (resp., exactly) i that arrived strictly before v. Layer H_i is maintained by running the online algorithm Mlast-Alg with R_i as sources and X_{i+1} as sinks. This ensures that every $u \in R_i$ has $d_{H_i}(u, X_{i+1}(v)) \le 3d(u, X_{i+1}(v))$.

Routing. We route v's demand to the root on path $P(v)$ that is constructed as follows. The path $P(v)$ is constructed iteratively and consists of M different segments $P_i(v)$, one per type. Initially, v's demand is located at v and the segments are all empty. While the current terminal w containing v's demand is not the root, choose w' to be the terminal in $H_{\tau(w)}$ with $\tau(w') > \tau(w)$ nearest to w (nearest in terms of distances in $H_{\tau(w)}$) and route v's demand to w' along the shortest path between w and w' in $H_{\tau(w)}$.

This completes the description of the algorithm. See Algorithm 2 for a summary.

4.1 Analysis Let us now consider the cost of the algorithm's solution due to each cable type. The fixed cost due to cables of type i is $\sigma_i d(H_i)$ since they are installed on layer H_i. For each terminal v of type $\tau(v) \le i$, $P_i(v)$ is the segment of its routing path consisting of type i cables. Hence the total cost of the algorithm's solution is

$$\text{ALG} := \sum_i \left(\sigma_i d(H_i) + \sum_{v:\tau(v)\le i} \beta_i d(P_i(v)) \right). \tag{4.1}$$

Proof outline. We want to show that ALG $\le$ OPT(T) for any HST embedding T, since Corollary 2.1 would then imply that ALG $\le O(\log k)$ OPT. The key is to decompose OPT on trees as follows: in §2.2 we defined the rent-or-buy functions $f_i(x) := \min\{\sigma_i, \beta_{i-1}\}$, and defined OPT$_i$ to be the cost of

Algorithm 2 Online Algorithm for Single-Sink Buy-at-Bulk

```
τ(r) ← ∞
H_i ← ∅ for all i
while terminal v arrives do
   // Determine type of v
   Let d_i(v) = d(v, X_i), B_i(v) = {u : d(u,v) ≤ d_i(v)/2^3}, and n_i(v) = |B_i(v)| for 1 ≤ i ≤ M
   τ(v) ← max{i : n_i(v) ≥ σ_i/β_{i-1}}
   // Update layers H_i
   for i ≤ τ(v) do
      Update H_i using MLAST-ALG with R_i(v) as sources and X_{i+1}(v) as sinks
      Install cable of type i on new edges of H_i
   // Find routing path P(v) to route v's demand to the root
   P_i(v) ← ∅ for all i; P(v) ← ∅
   while v's demand is not yet at r do
      Let w be the terminal containing v's demand and i = τ(w)
      Let P_i(v) ⊆ H_i be the shortest path from w to X_{i+1}(v) in H_i
      Route v's demand from w to X_{i+1}(v) along the path P_i(v)
```

the optimal solution with respect to f_i. Our proof (specifically Lemmas 4.3 and 4.2) will use the structure of the HST to give a charging argument showing that

$$\sigma_i d(H_i) + \sum_{v:\tau(v)<i} \beta_{i-1} d_i(v) \le O(1)\, \text{OPT}_i(T). \tag{4.2}$$

Roughly speaking, we argue that $d(H_i)$ can be charged to the optimal cost of the witnesses of the terminals of type exactly i, and moreover, the type selection rule forces terminals of type less than i to be spread out.

Thus, if we could prove that $d(P_i(v)) \le O(1)d_{i+1}(v)$, then we would be done, because we could decompose the expression for ALG in (4.1) as follows:

$$\text{ALG} \le O(1) \cdot \sum_i \left(\sigma_i d(H_i) + \sum_{v:\tau(v)<i} \beta_{i-1} d_i(v) \right). \tag{4.3}$$

Inequality (4.2) and Lemma 2.2 would imply that ALG $\le O(1)\,$OPT(T) for any HST embedding T, as desired.

However, $d(P_i(v))$ can be much larger than $d_{i+1}(v)$. This is because of the "selfishness" of the routing scheme: when a terminal w receives v's demand, it simply routes it to the terminal w' of higher type nearest to w, without any regard as to how far w' is from v. Fortunately, the fact that the incremental costs β_i are geometrically decreasing allows us to show that the total incremental cost incurred over all segments $P_i(v)$ are bounded. In particular, we show that $\sum_{v:\tau(v)\le i} \beta_i d(P_i(v)) \le O(1)\sum_i \sum_{v:\tau(v)\le i} \beta_i d_{i+1}(v)$. The rest of the proof will follow this general outline.

Bounding incremental cost. We begin by proving the above bound on the incremental cost.

LEMMA 4.1.

$$\sum_i \sum_{v:\tau(v)\le i} \beta_i d(P_i(v)) \le O(1)\sum_i \sum_{v:\tau(v)\le i} \beta_i d_{i+1}(v).$$

Proof. For terminal v, suppose its type-i segment $P_i(v)$ is non-empty: let w_i be the starting terminal of the segment. Since $P_i(v)$ starts with a terminal of type i and ends with a terminal of type at least $i+1$, so $\tau(w_i) = i$. Now since $P_i(v)$ is a shortest path in H_i from w_i to $X_{i+1}(v)$, Lemma 3.1(b) gives us $d(P_i(v)) \le 3 \cdot d(w_i, X_{i+1}(v))$. Since $d_{i+1}(v)$ is the distance from v to $X_{i+1}(v)$, $d(w_i, X_{i+1}(v)) \le d(w_i, v) + d_{i+1}(v)$. Moreover, the segments $P_{i'}(v)$ for $i' < i$ form a path from v to w_i, so $d(w_i, v) \le \sum_{i'<i} d(P_{i'}(v))$. Combining these inequalities, we get $d(P_i(v)) \le 3\left(d_{i+1}(v) + \sum_{i'<i} d(P_{i'}(v))\right)$. Unrolling the recursion, this gives $d(P_i(v)) \le \sum_{i'\le i} 3^{i-i'+1} d_{i'+1}(v)$. Hence, we get the following bound on the incremental cost of v:

$$\begin{aligned}\sum_{i\ge\tau(v)} \beta_i \cdot d(P_i(v)) &\le \sum_{i\ge\tau(v)} \beta_i \cdot \sum_{i'\le i} 3^{i-i'+1} d_{i'+1}(v)\\ &= \sum_{i\ge\tau(v)} d_{i+1}(v) \cdot \sum_{i'\ge i} 3^{i'-i+1}\beta_{i'},\end{aligned}$$

where the equality follows from rearranging the sums. But the β_is decrease geometrically by a factor of 3^2, so the last sum is dominated by the first term, and is $O(\beta_i)$. Hence the proof. □

Lemma 4.1 implies Inequality (4.3). Define $\mathrm{ALG}_i = \sigma_i d(H_i) + \sum_{v:\tau(v)<i} \beta_{i-1} d_i(v)$. The rest of this section will show that on any HST embedding T, $\mathrm{ALG}_i \le O(1)\,\mathrm{OPT}_i(T)$ for every i. Lemma 2.2 then implies that $\mathrm{ALG} \le O(1)\,\mathrm{OPT}(T)$ for any HST embedding T and so $\mathrm{ALG} \le O(\log k)\,\mathrm{OPT}$ by Corollary 2.1.

Charging to HST embeddings. For the following, fix an HST embedding T. Recall that for edge $e \in T$, $L(e)$ denotes the leaves below e, $T(e)$ the length of e, and $T(u,v)$ the distance between u and v in T. Also, observe that an edge e such that $r \notin L(e)$, the terminals in $L(e)$ either have to buy the edge at cost $\sigma_i T(e)$, or rent it at cost $\beta_{i-1}|L(e)|T(e)$. We record this lower bound for later.

$$\mathrm{OPT}_i(T) \ge \sum_{e\in T: r\notin L(e)} T(e)\min\{\sigma_i, \beta_{i-1}|L(e)|\}. \tag{4.4}$$

To upper-bound ALG_i, we bound both $\sigma_i d(H_i)$ and $\sum_{v:\tau(v)<i} \beta_{i-1} d_i(v)$ separately by $\mathrm{OPT}_i(T)$. In each case, we proceed by developing an appropriate charging scheme that charges to the edges of T and then arguing that the total charge received by each edge $e \in T$ is at most a constant times its contribution to the lower bound of $\mathrm{OPT}_i(T)$ in Inequality 4.4.

First, we bound $\sum_{v:\tau(v)<i} \beta_{i-1} d_i(v)$. The charging scheme is as follows: for each terminal v, charge β_{i-1} to an edge in T whose length is proportional to $d_i(v)$ and which contains v as a leaf. Then we argue that no edge is overcharged; in particular, for every edge $e \in T$, the total number of terminals that can charge e is at most σ_i/β_{i-1}. Finally, we use the fact that terminals charging to e are close together (by the bounded diameter property of HSTs), so if there were more than σ_i/β_{i-1} terminals charging e, then the one that arrived last would have been assigned a type of at least i.

LEMMA 4.2. $\sum_{v:\tau(v)<i} \beta_{i-1} d_i(v) \le O(1)\,\mathrm{OPT}_i(T)$.

Proof. Consider the following charging scheme. For each terminal v with type $\tau(v) < i$, if $d_i(v) \in [2^j, 2^{j+1})$, charge $2^{j+1}\beta_{i-1}$ to the length 2^{j-4} edge $e \in T$ whose leaves $L(e)$ contain v but not r. Such an edge must exist since otherwise $d_i(v) \le d(v,r) < 2^j$. The total charge received by the edges of T is at least $\sum_{v:\tau(v)<i} \beta_{i-1} d_i(v)$, and only edges e with $r \notin L(e)$ were charged.

Consider an edge $e \in T$ of length 2^{j-4}. Let $C(e) \subseteq L(e)$ be the set of terminals charging e. We claim that $|C(e)| \le \sigma_i/\beta_{i-1}$. Note that the total charge received by e is

$$\begin{aligned}2^{j+1}\beta_{i-1}|C(e)| &= 2^5\, T(e)\beta_{i-1}|C(e)|\\ &\le 2^5\, T(e)\min\{\sigma_i, \beta_{i-1}|L(e)|\}.\end{aligned}$$

By (4.4), this proves the lemma.

Now to prove the claim. Suppose for a contradiction, that $|C(e)| > \sigma_i/\beta_{i-1}$. Let v be the last-arriving terminal of $C(e)$. Since e is a length 2^{j-4} edge and v charged e, we have $d_i(v) \ge 2^j$ and $\tau(v) < i$. Moreover, since $C(e) \subseteq L(e)$, we have that $\mathrm{diam}(C(e)) < 2^{j-3} \le d_i(v)/2^3$. Thus, every terminal $u \in C(e)$ is within distance $d_i(v)/2^3$ from v so $C(e) \subseteq B_i(v)$. This implies that $|B_i(v)| \ge |C(e)| \ge \sigma_i/\beta_{i-1}$ so v should have been assigned a type that is at least i, contradicting the fact that $\tau(v) < i$. Therefore, $|C(e)| \le \sigma_i/\beta_{i-1}$, as desired. □

Bounding the Fixed Cost of H_i. There are three steps to the proof. The first step is to use Lemma 3.1 to charge the fixed cost to the terminals of type i. The second is to argue that since each terminal v of type i has at least σ_i/β_{i-1} witnesses nearby, v can use the cost incurred by $\mathrm{OPT}_i(T)$ in routing its witnesses to pay off the charge it accumulated. Finally, we use the fact that terminals of type i that accumulate similar charges must be far apart to argue that no witness is overcharged.

LEMMA 4.3. *$\sigma_i d(H_i) \le O(1)\,\mathrm{OPT}_i(T)$.*

Proof. The layer H_i is an MLAST whose set of sources is R_i (the terminals of type exactly i) and sinks $S_i = X_{i+1}$, the terminals of higher type. Let $\mathrm{class}_i(v)$ be the class assigned by this MLAST algorithm to terminal v. By Lemma 3.1, we have $\sigma_i d(H_i) \le O(1)\sum_{v\in R_i}\sigma_i 2^{\mathrm{class}_i(v)}$. Define $R_i(j) = \{v \in R_i : \mathrm{class}_i(v) = j\}$. Let E_j be the set of length 2^{j-4} edges of T. To prove the lemma, we will show that for each j,

$$\sigma_i|R_i(j)| \le \sum_{e\in E_j : r\notin L(e)} \min\{\sigma_i, \beta_{i-1}|L(e)|\}. \tag{4.5}$$

This would then imply that

$$\begin{aligned}
&\sigma_i d(H_i)\\
&\le O(1)\sum_{v\in R_i}\sigma_i 2^{\mathrm{class}_i(v)}\\
&\le O(1)\sum_j 2^j\sigma_i|R_i(j)|\\
&\le O(1)\sum_j 2^{j-4}\sum_{e\in E_j:r\notin L(e)}\min\{\sigma_i,\beta_{i-1}|L(e)|\}.
\end{aligned}$$

But $T(e) = 2^{j-4}$ for edges $e \in E_j$, so (4.4) bounds the cost by $O(1)\,\mathrm{OPT}_i(T)$ to complete the proof.

Now to prove (4.5). We will show that for each $v \in R_i(j)$ (i.e., having type i, and class j in the MLAST H_i corresponding to type-i terminals), every terminal u in its ball $B_i(v)$ has to be routed on some level-j edge $e \in E_j$ with $r \notin L(e)$, and that the level-j edges used by $B_i(v)$ is disjoint from the edges used by $B_i(v')$ for any other $v' \in R_i(j)$. More formally, for each terminal v, define $e_j(v)$ to be the unique edge of E_j such that $v \in L(e_j(v))$ and define $E_j(v) := \{e_j(u) : u \in B_i(v)\}$. We need the following claims.

CLAIM 4.1. *For every $v \in R_i(j)$ and $u \in B_i(v)$, we have $d(u,v) \le 2^{j-3}$.*

Proof. By definition of $B_i(v)$, we have the following bound: $d(u,v) \le d_i(v)/2^3 = d(v, X_i(v))/2^3$. Now observe that $X_i(v) = R_i(v) \cup S_i(v)$, where $R_i(v)$ and $S_i(v)$ are the sources and sinks that arrived before v in the MLAST for type i. By Lemma 3.1(d), $d(v, X_i(v)) \le 2^{\mathrm{class}_i(v)} = 2^j$. Combining this with the above bound on $d(u,v)$ gives us $d(u,v) \le 2^{j-3}$, as desired. □

CLAIM 4.2. *For every $v \in R_i(j)$, we have $\bigcup_{e\in E_j(v):r\notin L(e)} L(e) \supseteq B_i(v)$. Moreover, $E_j(v) \cap E_j(v') = \emptyset$ for distinct $v, v' \in R_i(j)$.*

Proof. To prove the first part of the claim, observe that by definition of $E_j(v)$, we have $\bigcup_{e\in E_j(v)} L(e) \supseteq B_i(v)$. Thus, it suffices to show that for every $v \in R_i(j)$ and each terminal $u \in B_i(v)$, the edge $e_j(v)$ does not contain r as a leaf. Suppose, towards a contradiction, that there exists $u \in B_i(v)$ such that $r \in L(e_j(u))$. Since $e_j(u)$ has length 2^{j-4}, we have $\mathrm{diam}(L(e_j(u)) \le 2^{j-3}$; moreover, $r, u \in L(e_j(u))$ and so $d(u,r) \le 2^{j-3}$. Now, Claim 4.1 implies that $d(u,v) \le 2^{j-3}$. Therefore, $d(v,r) \le d(u,v) + d(u,r) < 2^j$. However, this contradicts Lemma 3.1(d), which implies that $d(v,r) \ge 2^j$. Thus, $r \notin L(e_j(u))$.

To prove the second part of the claim, suppose towards a contradiction, that there exist $v, v' \in R_i(j)$ and $u \in B_i(v)$ and $u' \in B_i(v')$ such that $e_j(u) = e_j(u')$. By triangle inequality, $d(v,v') \le d(u,v) + d(u,u') + d(u',v')$. Claim 4.1 implies that $d(u,v), d(u',v') \le 2^{j-3}$. Since u, u' are leaves of the same edge of length 2^{j-4}, we get $d(u,u') \le 2^{j-3}$. Therefore, $d(v,v') < 2^j$. On the other hand, Lemma 3.1(c) says that $d(v,v') \ge 2^j$. This gives us our desired contradiction. □

With these claims in hand, we have

$$\begin{aligned}
&\sum_{e\in E_j:r\notin L(e)}\min\{\sigma_i,\beta_{i-1}|L(e)|\}\\
&\ge \sum_{v\in R_i(j)}\sum_{e\in E_j(v):r\notin L(e)}\min\{\sigma_i,\beta_{i-1}|L(e)|\}\\
&\ge \sum_{v\in R_i(j)}\min\{\sigma_i,\beta_{i-1}\sum_{e\in E_j(v):r\notin L(e)}|L(e)|\}\\
&\ge \sum_{v\in R_i(j)}\min\{\sigma_i,\beta_{i-1}|B_i(v)|\}\\
&\ge \sum_{v\in R_i(j)}\sigma_i = \sigma_i|R_i(j)|,
\end{aligned}$$

where the first inequality follows from the second part of Claim 4.2, the third inequality from the first part, and the last inequality from the fact that $|B_i(v)| \ge \sigma_i/\beta_{i-1}$. □

Having proved these two lemmas, we have that for every HST embedding T and every i, we have

$\mathrm{ALG}_i \le O(1)\,\mathrm{OPT}_i(T)$. Summing over all i and using Lemma 2.2, we have $\mathrm{ALG} \le O(1)\,\mathrm{OPT}(T)$ for any HST embedding T, and so $\mathrm{ALG} \le O(\log k)\,\mathrm{OPT}$. This proves Theorem 1.1.

5 Conclusions

We defer the results on oblivious buy-at-bulk to the full version. It also contains our results for online LASTs and online low-stretch spanners, which are used in these results. Several open questions remain: can we do better than $O(\log^2 k)$ for the oblivious randomized case, and $O(\log^{2.5} k)$ for the oblivious deterministic case? Can we get online constructions of tree embeddings with stretch better than $O(\log k \log \Delta)$?

References

[AA97] Baruch Awerbuch and Yossi Azar. Buy-at-bulk network design. In *38th Annual Symposium on Foundations of Computer Science, FOCS '97, Miami Beach, Florida, USA, October 19-22, 1997*, pages 542–547, 1997.

[AAB04] Baruch Awerbuch, Yossi Azar, and Yair Bartal. On-line generalized steiner problem. *Theor. Comput. Sci.*, 324(2-3):313–324, 2004.

[AKPW95] Noga Alon, Richard M. Karp, David Peleg, and Douglas West. A graph-theoretic game and its application to the k-server problem. *SIAM J. Comput.*, 24(1):78–100, 1995.

[And04] Matthew Andrews. Hardness of buy-at-bulk network design. In *45th Symposium on Foundations of Computer Science (FOCS 2004), 17-19 October 2004, Rome, Italy, Proceedings*, pages 115–124, 2004.

[Bar96] Yair Bartal. Probabilistic approximations of metric spaces and its algorithmic applications. In *37th Annual Symposium on Foundations of Computer Science, FOCS '96, Burlington, Vermont, USA, 14-16 October, 1996*, pages 184–193, 1996.

[Bar98] Yair Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing, Dallas, Texas, USA, May 23-26, 1998*, pages 161–168, 1998.

[Bar04] Yair Bartal. Graph decomposition lemmas and their role in metric embedding methods. In *Algorithms - ESA 2004, 12th Annual European Symposium, Bergen, Norway, September 14-17, 2004, Proceedings*, pages 89–97, 2004.

[BC97] Piotr Berman and Chris Coulston. On-line algorithms for steiner tree problems (extended abstract). In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 344–353, 1997.

[Bol04] Béla Bollobas. *Extremal Graph Theory.* Dover Publications, Incorporated, 2004.

[ECKP15] Alina Ene, Deeparnab Chakrabarty, Ravishankar Krishnaswamy, and Debmalya Panigrahi. Online buy-at-bulk network design. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 545–562, 2015.

[FRT04] Jittat Fakcharoenphol, Satish Rao, and Kunal Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *J. Comput. Syst. Sci.*, 69(3):485–497, 2004.

[GE05] Ashish Goel and Deborah Estrin. Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk. *Algorithmica*, 43(1-2):5–15, 2005.

[GHR06] Anupam Gupta, Mohammad Taghi Hajiaghayi, and Harald Räcke. Oblivious network design. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 970–979, 2006.

[GI06] Fabrizio Grandoni and Giuseppe F. Italiano. Improved approximation for single-sink buy-at-bulk. In *Algorithms and Computation, 17th International Symposium, ISAAC 2006, Kolkata, India, December 18-20, 2006, Proceedings*, pages 111–120, 2006.

[GKK+01] Naveen Garg, Rohit Khandekar, Goran Konjevod, R. Ravi, F. Sibel Salman, and Amitabh Sinha II. On the integrality gap of a natural formulation of the single-sink buy-at-bulk network design problem. In *Integer Programming and Combinatorial Optimization, 8th International IPCO Conference, Utrecht, The Netherlands, June 13-15, 2001, Proceedings*, pages 170–184, 2001.

[GKR03] Anupam Gupta, Amit Kumar, and Tim Roughgarden. Simpler and better approximation algorithms for network design. In *35th STOC*, pages 365–372, 2003.

[GMM09] Sudipto Guha, Adam Meyerson, and Kamesh Munagala. A constant factor approximation for the single sink edge installation problem. *SIAM J. Comput.*, 38(6):2426–2442, 2009.

[GP12] Ashish Goel and Ian Post. One tree suffices: A simultaneous o(1)-approximation for single-sink buy-at-bulk. *Theory of Computing*, 8(1):351–368, 2012.

[GRS11] Fabrizio Grandoni, Thomas Rothvoss, and Laura Sanità. From uncertainty to nonlinearity: Solving virtual private network via single-sink buy-at-bulk. *Mathematics of Operations Research*, 36(2):185–204, 2011.

[IW91] Makoto Imase and Bernard M. Waxman. Dynamic steiner tree problem. *SIAM J. Discrete Math.*, 4(3):369–384, 1991.

[JR09] Raja Jothi and Balaji Raghavachari. Improved approximation algorithms for the single-sink buy-at-bulk network design problems. *J. Discrete Algorithms*, 7(2):249–255, 2009.

[KRY95] Samir Khuller, Balaji Raghavachari, and Neal E. Young. Balancing minimum spanning trees and shortest-path trees. *Algorithmica*, 14(4):305–321, 1995.

[LPS88] Alexander Lubotzky, Ralph Phillips, and Peter

Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[MP98] Yishay Mansour and David Peleg. An approximation algorithm for minimum-cost network design. In *Robust Communication Networks: Interconnection and Survivability, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, November 18-20, 1998*, pages 97–106, 1998.

[SCRS97] F. Sibel Salman, Joseph Cheriyan, R. Ravi, and S. Subramanian. Buy-at-bulk network design: Approximating the single-sink edge installation problem. In *Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, 5-7 January 1997, New Orleans, Louisiana.*, pages 619–628, 1997.

[Tal02] Kunal Talwar. The single-sink buy-at-bulk LP has constant integrality gap. In *Integer Programming and Combinatorial Optimization, 9th International IPCO Conference, Cambridge, MA, USA, May 27-29, 2002, Proceedings*, pages 475–486, 2002.

[Umb15] Seeun Umboh. Online network design algorithms via hierarchical decompositions. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1373–1387, 2015.