Buy-at-Bulk Network Design with Protection *

Spyridon Antonakopoulos † Chandra Chekuri ‡ Bruce Shepherd § Lisa Zhang ¶

Abstract

We consider approximation algorithms for buy-at-bulk network design, with the additional constraint that demand pairs be protected against edge or node failures in the network. In practice, the most popular model used in high speed telecommunication networks for protection against failures, is the so-called 1+1 model. In this model, two edge or node-disjoint paths are provisioned for each demand pair. We obtain the first non-trivial approximation algorithms for buy-at-bulk network design in the 1+1 model for both edge and node-disjoint protection requirements. Our results are for the single-cable cost model, which is prevalent in optical networks. More specifically, we present a constant-factor approximation for the single-sink case, and an $O(\log^3 n)$ approximation for the multi-commodity case. These results are of interest for practical applications and also suggest several new challenging theoretical problems.

1 Introduction

The telecommunications industry is the inspiration for numerous network optimization problems. In this paper, we consider buy-at-bulk network design problems that arise in the design and operation of modern optical core networks [6]. These networks are characterized by the following two salient features: (i) very high capacity achieved via DWDM (Dense Wavelength Division Multiplexing) based optical transmission technology and (ii) expensive equipment exhibiting economies of scale. In such networks, each link carries enormous amounts of traffic and hence the failure of a link or a node represents an unacceptable degradation of service. Therefore, fault tolerance is an integral part of the design. Although there are a variety of ways to ensure fault tolerance, one of the most commonly used solutions in optical core networks is to set up, for each commodity, so-called *dedicated* or *1+1* protection. This amounts to reserving a pair of disjoint paths between the source and destination nodes of each commodity. The popularity of the 1+1 model comes from its operational simplicity and high restoration speed.

Disjointness may be defined in several ways, according to requirements of the commodity in question. For instance, the commonly used measures include "site-disjointness", where the two paths do not share any common nodes; edge-disjointness, where the two paths do not share any common links; and cable or fiber-disjointness, where the two paths must use distinct fibers/cables if they go through the same link. In this context, a central problem faced by network operators and equipment vendors is to build a cost-effective and bandwidth-efficient network that supports a multitude of traffic at the desired level of protection. The network operators look to utilize their network resources as efficiently as possible, and the equipment vendors seek to find innovative cost advantages to obtain a competitive edge in bidding for contracts from the network providers. We refer the reader to [6, 31, 27, 30] for in-depth descriptions of the various issues in optical network design.

We give a formal description of the optimization problem that abstracts the above problem. The input consists of an undirected edge-weighted graph $G = (V, E)$, and a set of h node pairs s_1t_1, s_2t_2, $\ldots, s_ht_h$ that represent different traffic demands. Each pair has a non-negative demand value $\mathsf{dem}(i)$ that needs to be routed between s_i and t_i and also specifies a protection requirement. In this paper we restrict our attention to the 1+1 model in which each demand requires node-disjoint protection. A feasible solution consists of a collection of path pairs $(P_1, Q_1), \ldots, (P_h, Q_h)$, where P_i and Q_i are internally node-disjoint paths between s_i and t_i and each carries a reserved bandwidth of $\mathsf{dem}(i)$. If these paths induce a requirement of b_e units of bandwidth on edge e of the network, then equipment that can support this requirement has to be purchased.

Now, let us discuss the cost model for purchasing bandwidth on the edges. In this paper, we focus on a sim-

*This work was partially supported by a basic research grant N00014-06-1-0396 from ONR to Bell Laboratories.

†Department of Computer Science, Columbia University, New York, NY 10027. Email: spyros@cs.columbia.edu. Supported by NSF grant CCF-0430946 and an Alexander S. Onassis Foundation Scholarship.

‡Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801. Email: chekuri@cs.uiuc.edu.

§Department of Mathematics and Statistics, McGill University, Montreal, Canada. Email: bruce.shepherd@mcgill.ca.

¶Bell Laboratories, Alcatel-Lucent, Murray Hill, NJ 07974. Email: ylz@research.bell-labs.com.

ple cost model, namely the single-cable cost model: bandwidth can be purchased in integer quantities of a *cable* of capacity k. The cost of purchasing a cable on edge e is c_e. Thus, the cost of purchasing a bandwidth of b_e units on edge e is $f_e(b_e) = \lceil \frac{b_e}{k} \rceil c_e$. The objective is to minimize the total cost $\sum_e f_e(b_e)$ over all possible choices of $(P_1, Q_1), \ldots, (P_h, Q_h)$. The single-cable cost function closely models DWDM networks, where each optical fiber carries the same number of wavelengths k, and each edge e has a cost c_e for deploying one copy of a fiber; the cost accounts for equipment along the edge and at the end nodes of the edge (see [6]). We give an overview of more general cost functions, namely the non-uniform and the uniform multi-cable functions, in the related work section.

Observe that, even in the single-cable setting, the buy-at-bulk problem captures, as special cases, some well-known NP-hard connectivity problems such as the minimum-cost Steiner tree and the minimum-cost Steiner forest problems. Moreover, Andrews [1] has shown that even the single-cable problem without protection constraints is hard to approximate to within an $\Omega(\log^{1/4-\epsilon} n)$ factor; this separates the approximability of the buy-at-bulk problem from those of connectivity problems. In the connectivity setting, survivability and protection constraints have long been studied and include classical problems. Jain [21] devised the important iterative rounding method that yields a 2 approximation algorithm for the survivable network design problem (SNDP), in which the goal is to find a minimum-cost subgraph that satisfies given edge connectivity requirements between each pair of nodes in a graph. In [13] this technique was extended to handle node connectivity, when the requirements are restricted to be in the set $\{0, 1, 2\}$.

Buy-at-bulk network design without protection has received substantial attention in the past decade, including some recent work on super-constant lower bounds in the simplest setting [1], and poly-logarithmic upper bounds in the most general non-uniform setting [7, 8]. On the contrary, the variant with protection has not been so far considered in the literature on approximation algorithms. One reason for this is the difficulty of the buy-at-bulk problem, even without protection constraints. Although the first approximation algorithm for the multiple-cable setting appeared in 1997 [3], the algorithm was based on a technique that was not sufficiently flexible. It is only recently that alternative algorithms [5, 7] were developed that not only handled the non-uniform cost functions, but also provided new algorithmic approaches and insights. Further, for SNDP, the iterative rounding method of Jain [21], and the earlier primal-dual approach [33], strongly rely on the structural properties on the underlying linear program, which do not hold for the buy-at-bulk problem.

Our primary motivation to study this problem arose while developing a sequence of optical network design tools at Bell Laboratories. We realized the ubiquity of the 1+1 model in practice, the lack of theoretical understanding of protected buy-at-bulk network design and a dearth of useful heuristic methods for the problem. Most algorithms used in practice are based on simple ad hoc methods combining greedy algorithms, local improvement and some enumeration. We hope this paper serves as a starting point in addressing the challenges from the theoretical point of view, as well as in providing insights that lead to more sophisticated and effective heuristics.

Results. We give approximation algorithms for buy-at-bulk network design in the 1+1 protection model for the single-cable setting. Observe that the 1+1 edge-disjoint protection problem can be reduced in a straightforward fashion to the 1+1 node-disjoint protection problem. In fact, for the edge-disjoint case our arguments can be substantially simplified; however, our focus here is on the node version, as it is the version arising more commonly in practice. We note that hardness results for the unprotected problems carry over to their protected counterparts via simple reductions.

Our first result is for the single-sink problem. This is the special case of the problem where all the pairs have one terminal node in common. In other words, the pairs are $st_1, st_2, \ldots, st_h$ and s is a common sink. We present an $O(1)$ approximation algorithm for it and also establish an $O(1)$ integrality gap for a natural linear programming relaxation.

Our second result is an $O(\log^3 h)$ approximation for the multi-commodity problem. In particular, we show that that an α approximation for the single-sink problem via a natural LP relaxation yields an $O(\alpha \log^3 h)$ approximation for the multi-commodity problem, and combine this with our result for the single-sink problem. A technique developed in the recent work of Kortsarz and Nutov [23] for the unprotected buy-at-bulk problem can be applied in our setting as well, and this leads to an improved ratio of $O(\alpha \log^2 h)$. The details of this improvement are deferred to a later version.

Overview of Algorithmic Ideas. The high-level framework of our algorithms is reminiscent of familiar approaches that have been applied to buy-at-bulk network design without protection. Nevertheless, the transition to the protected setting requires some new algorithmic ideas and in the following, we give a brief overview of these.

For the single-sink problem, we take advantage of the single-cable model to start with a good lower bound on the optimal solution: we compute a minimum-cost subgraph H of G that has two node-disjoint paths from each terminal t_i to the sink s. The graph H is used in a clustering procedure to find aggregation points, called *centers*. The idea is to route the flow of each terminal t_i to two distinct centers, via node-disjoint paths. Furthermore, the centers need to receive $\Omega(k)$ flow, so that they can route to the sink independently. We remark that clustering and re-routing of flow, as above, is a natural algorithmic paradigm that has been applied in algorithms for single-sink unprotected buy-at-bulk [28, 2, 16, 18, 25]. For the unprotected case, a simple tree based clustering procedure suffices, where each cluster contains terminals with $\Theta(k)$ amount of demand, and a center can be chosen arbitrarily from the cluster.

In the protected case, in particular the node-disjoint setting, a straightforward clustering procedure as above does not guarantee that each terminal can find disjoint paths to two distinct centers. We give a clustering procedure that enables us to overcome this difficulty; a distinctive feature of this procedure is that it may create clusters that enclose an arbitrarily large (compared to k) amount of demand, but in that case the cluster is required to satisfy some special property that can be exploited. Some of the methods we employ in sending flow to two centers are inspired by the work in [10], however the node-disjointness calls for several new technical ideas.

The multi-commodity problem is considerably harder to approach directly, and here we build on the recent algorithmic paradigm developed for the unprotected non-uniform problem [19, 7]. At the high level, the algorithm uses an iterative greedy approach. In each iteration, it finds a partial solution of good *density* amongst the remaining demand pairs, where density is the ratio of the solution cost to the number of pairs connected. In [19, 7, 8] the problem of finding a partial solution of good density is effectively reduced to a single-sink problem. A key step in the reduction is to show the existence of a solution with near-optimal density that also has a *junction* structure: demand pairs connect to each other via a common junction node r, and this enables one to employ single-sink techniques by guessing r.

A similar scheme can be applied to the edge-disjoint protection problem; nevertheless, this does not suffice for the node-protected version. Indeed, even if terminals s_i and t_i individually have two node-disjoint paths to r, they may still not be 2-node-connected. To overcome that difficulty, we show that the basic junction scheme can be extended to use a *pair* of nodes (u, v), as a junction through which multiple pairs connect. This requires more intricate arguments. Furthermore, we believe this scheme offers an interesting idea for new heuristics, which should be evaluated against the current methods that are based on greedy approaches.

Related work. We briefly discuss closely related work, beginning with a review of more general cost models. The most general cost model considered in the buy-at-bulk problem is the *non-uniform* case, where each $e \in E$ has an associated concave or sub-additive function $f_e : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ and $f_e(b_e)$ is the cost of purchasing b_e units of bandwidth on e. In the *uniform* cost model, the function f_e is restricted to be identical for all edges, up to constant factors. An alternative (and equivalent) definition of this model stipulates that there exists a fixed set of cables, with capacities $u_1 < u_2 < \cdots < u_r$ and costs per unit length $c_1 < c_2 < \cdots < c_r$, such that the cost per bandwidth decreases; that is $c_1/u_1 > c_2/u_2 > \cdots > c_r/u_r$. To support b_e units of flow on edge e, one needs to purchase the cheapest combination of cables of total capacity at least b_e.

As mentioned above, the buy-at-bulk problem has so far been studied only in the unprotected setting. One of the early approximation algorithm formulations of the problem was due to Salman et al. [28]. In subsequent work, a number of variants have been considered. Even the simplest versions of buy-at-bulk network design, including the single-sink single-cable problem, are APX-hard since they generalize the Steiner tree problem.

Regarding the uniform multi-commodity problem, Awerbuch and Azar [3] showed that it is easy to solve on a tree and then reduced the problem on general graphs to one on a tree using embeddings into random tree metrics [4, 12], thus obtaining an $O(\log n)$ approximation. For the uniform single-sink problem, Andrews and Zhang [2] gave an approximation ratio that is independent of the number of nodes (but does depend on the cost function); an $O(1)$ approximation was first achieved in [16], with subsequent refinements and improvements in the ratio [18, 22]. For a special case of the multi-commodity problem called the rent-or-buy problem, an $O(1)$ approximation is known [24, 17].

For the non-uniform single-sink problem, Meyerson et al. [26] presented an $O(\log n)$ approximation. Charikar and Karagiozova [5] gave an $\exp(O(\sqrt{\log n \log\log n}))$ approximation for the non-uniform multi-commodity problem; recently, the first poly-logarithmic approximation was obtained in [19, 7], which also introduced the junction scheme that we now extend. The ratio achieved was $O(\log^4 h)$, and in [8] the same ratio was established even for the setting in which nodes have costs.

Andrews [1] showed that there is no $O\big(\log^{1/2-\epsilon} n\big)$ approximation algorithm for the non-uniform multi-commodity problem, unless NP has efficient randomized algorithms. In the uniform case, including the single-cable model, the hardness factor becomes $O\big(\log^{1/4-\epsilon} n\big)$. Moreover, for the single-sink non-uniform problem a hardness factor of $O(\log\log n)$ is known [11].

Connectivity problems have a rich history in classical combinatorial optimization, and there is vast literature on the subject. We refer to [29] for exact algorithms and classical results and [32, 20, 21, 13] for pointers to approximation algorithms. In particular, Jain [21] and Fleischer et al. [13] present 2 approximation algorithms for SNDP and the element connectivity problem (which generalizes SNDP), respectively. In [13] a 2 approximation algorithm is also obtained for the node-connectivity version of SNDP when the requirements are restriced to lie in the set $\{0, 1, 2\}$; we make use of this algorithm.

2 Single-Sink Buy-at-Bulk with Protection

An instance of the node-protected single-sink problem consists of a graph $G = (V, E)$, a sink node $s \in V$, a set of terminals $\mathcal{T} = \{t_1, t_2, \ldots, t_h\} \subseteq V \backslash s$, and a demand function $\mathsf{dem} : \mathcal{T} \to \mathbb{N}^*$. We use $k \in \mathbb{N}^*$ throughout to denote the capacity of the cable that can be installed in integral copies on any edge $e \in E$, at a cost c_e per cable. Thus, carrying bandwidth b_e on e costs $\lceil b_e/k \rceil c_e$. Our algorithm consists of three high level steps that follow the outline given in Section 1.

- **Connectivity:** Find a subgraph $H = (V_H, E_H)$ of G such that each t_i has two node-disjoint paths to s in H.
- **Clustering:** Partition the node set V_H into disjoint subsets $X_1, X_2, \cdots, X_\ell$ called *clusters*, such that for $1 \leq i \leq \ell$ the induced subgraph $H[X_i]$ is connected and $\mathsf{dem}(X_i) \geq k$, where $\mathsf{dem}(X_i) = \sum_{t_j \in X_i} \mathsf{dem}(t_j)$. Clusters exhibit additional properties that facilitate analysis.
- **Routing:** Use clusters to identify a subset $S \subset V_H$ called *centers*. For each terminal $t_j \in \mathcal{T}$, route $\mathsf{dem}(t_j)$ to two distinct centers $u_j, v_j \in S$ using node-disjoint paths, such that every center receives $\Omega(k)$ flow from terminals. Then, route the flow from each center to s directly and independently.

2.1 Connectivity

We apply the 2 approximation algorithm from [13] for the node-connectivity version of the survivable network design problem on G, with a connectivity requirement of 2 between s and t_i, for each $t_i \in \mathcal{T}$, and 0 for every other pair of nodes. Let $H = (V_H, E_H)$ be the subgraph returned. We install one cable on each edge of H and hence $\mathsf{cost}(H) \leq 2\,\mathsf{cost}(\mathrm{OPT}_{\mathrm{ss}})$, where $\mathrm{OPT}_{\mathrm{ss}}$ is the optimal solution to the node-protected single-sink problem.

For simplicity, we henceforth assume that H is 2-node-connected, because the clustering and routing procedures can be applied to each 2-node-connected component of H separately.

2.2 Clustering

We describe an algorithm to partition H (in fact, any 2-node-connected node-weighted graph) into clusters, as mentioned earlier. A cluster X is called *small* if $\mathsf{dem}(X) < k$; *normal* if $k \leq \mathsf{dem}(X) \leq 2k$; and *jumbo* if $\mathsf{dem}(X) > 2k$. Ideally, we would like to partition V_H so that all clusters are normal. However, this is not always possible. Instead, we allow jumbo clusters in the partition, as long as they possess certain structural properties. In particular, a

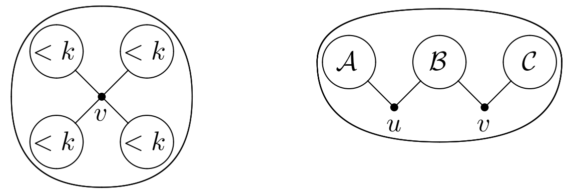

Figure 1. A typical star-like cluster with 4 components (left) and a typical twin cluster (right). By definition, $\mathsf{dem}(\mathcal{A}) + \mathsf{dem}(\mathcal{B}) < 2k$ **and** $\mathsf{dem}(\mathcal{B}) + \mathsf{dem}(\mathcal{C}) < 2k$**.**

jumbo cluster X is *star-like* if and only if there exists a *special* node $v \in X$ such that each connected component of $H[X \backslash v]$ contains $< k$ demand. Similarly, X is *twin* if and only if there exist two special nodes $u, v \in X$ such that for all $w \in \{u, v\}$, one component of $H[X \backslash w]$ contains $< 2k$ demand, while all other components contain $< k$ demand. Figure 1 provides a visualization of the above definitions.

Lemma 2.1. *In polynomial time, the node set V_H can be partitioned into clusters $X_1, X_2, \ldots, X_\ell$, such that for $1 \leq i \leq \ell$: (a) the induced subgraph $H[X_i]$ is connected; (b) $\mathsf{dem}(X_i) \geq k$; and (c) if $\mathsf{dem}(X_i) > 2k$, then X_i is either star-like or twin.*

Proof. We need a few more definitions. Given some partition of V_H into clusters $X_1, X_2, \ldots, X_\ell$, we say that $v \in X_i$ is a *border node* if and only if there exists a $u \in X_j$, $i \neq j$, such that the edge $uv \in E_H$. In that case, X_i and X_j are *neighboring* clusters and v is a *neighbor* of X_j. Moreover, a node v is called *critical* if and only if: (i) it belongs to a cluster X with $\mathsf{dem}(X) \geq 2k$, and (ii) at least one of the connected components of $H[X \backslash v]$ contains $< k$ demand; and (iii) it is a neighbor of one or more small clusters. We say that these clusters *contend* for the critical node. Furthermore, a component of $H[X \backslash v]$ that has $\geq k$ demand is called *self-sufficient*. Note that a non-critical node may become critical and vice-versa several times during the clustering procedure, which is presented below.

Initially, each node of V_H is in a cluster of its own. Henceforth, consider all small clusters simultaneously. Apply the following transformations, when feasible, in order of priority. Repeat until there are no small clusters left.

1. If the total demand in two neighboring clusters is at most $2k$, merge them.
2. If a small cluster has a neighbor node that is not critical, move the node to that cluster.
3. Consider a cluster X with a critical node v. Separate any self-sufficient components of $H[X \backslash v]$ into new clusters. Keep v and all other components together.

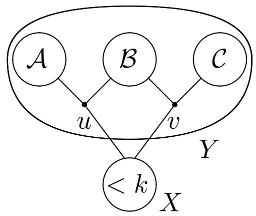

Figure 2. An example for which transformation 5 is necessary. Since no other transformation applies, $\mathsf{dem}(\mathcal{A}) + \mathsf{dem}(\mathcal{B}) < k$ **and** $\mathsf{dem}(\mathcal{B}) + \mathsf{dem}(\mathcal{C}) < k$**. Compare with Figure 1.**

If the remaining cluster is small, merge it immediately with a small cluster contending for v (like in transformation 1).

4. If there exists a small cluster X that contends for only one critical node v of cluster Y, then create a jumbo cluster by merging X and Y into a new cluster Z. The connected components of $H[Z\backslash v]$ are $H[X]$ and the components of $H[Y\backslash v]$, none of which is self-sufficient. Therefore, Z is star-like.

5. If there exists a small cluster X that contends for several critical nodes of cluster Y (e.g. as in Figure 2), then create a jumbo cluster by merging X and Y into a new cluster Z.

   Let $U \subseteq Y$ be the set of critical nodes contended for by X. We say that $w \in U$ is *interesting* if and only if all other elements of U belong to only one component of $H[Y\backslash w]$, denoted W. After the merger, $H[V_W \cup X]$ is a connected component of $H[Z\backslash w]$ containing $< 2k$ demand. All other components of $H[Z\backslash w]$ are also components of $H[Y\backslash w]$ and hence contain $< k$ demand. Note that there at least *two* interesting critical nodes in U, so Z is a twin cluster.

A twin cluster created at some point in the procedure may later become star-like or normal due to transformations 2 and/or 3. Similarly, a star-like cluster may become normal. Obviously, when the procedure terminates, there are only normal, star-like and twin clusters in the partition.

It remains to sketch why the algorithm terminates in polynomial time. Let n_1, n_2, n_3 be the combined number of normal and jumbo clusters, the number of small clusters and the number of nodes in small clusters, respectively, at any given moment. Clearly, all three numbers are bounded by $|V_H|$, since every cluster contains at least one node. Observe that if any of the above transformations is applied, n_1 does not decrease and n_2 does not increase. Furthermore, if both n_1 and n_2 are left unchanged, n_3 strictly increases. Consequently, there can be no more than $|V_H|$ consecutive transformations in which both n_1 and n_2 remain constant, which implies that the total number of transformations is $O\big(|V_H|^2\big)$. Since each transformation can be implemented in polynomial time, thus completing the proof. $\square$

2.3 Routing

We now describe a scheme to implement the routing step of the algorithm using the clustering of H. The routing involves several phases and the analysis goes hand-in-hand with each phase. In this section, an edge e of H is called *intra-cluster* if there exists a cluster X such that e is an edge of $H[X]$. Otherwise, e is an *inter-cluster* edge.

Phase 1: We process each cluster X_i separately. First of all, take a spanning tree of $H[X_i]$ and find its balanced separator, with respect to the amount of demand. Call this node the *center* of X_i and denote it by v_i. If X_i is star-like, its special node is an obvious choice for v_i.

Proposition 2.2. *For each terminal $t_j \in X_i\backslash v_i$, there exist two node-disjoint paths $P_1(t_j)$ and $P_2(t_j)$ using edges of $H[X_i]$, both starting from t_j, such that $P_1(t_j)$ ends at v_i and $P_2(t_j)$ ends at a border node $b(t_j)$ of X_i.*

Proof. Create a graph $\mathcal{G}$ from H, by contracting all nodes of $V_H\backslash X_i$ into v^*. Since $H[X_i]$ is connected, v^* is not a cut vertex of $\mathcal{G}$. Furthermore, if $u \in X_i$ is a cut vertex of $\mathcal{G}$, then it is also a cut vertex of H, contradicting H's biconnectivity. Therefore, $\mathcal{G}$ has no cut vertices, i.e. it is biconnected. Hence, there exist two node-disjoint paths $P_1(t_j)$, $P_2(t_j)$ from t_j to v_i and v^*, respectively, which can be found by solving a min-cost flow problem. After deleting the last edge of $P_2(t_j)$, these two paths satisfy all required properties. $\square$

Send flow equal to $\mathsf{dem}(t_j)$ along each of $P_1(t_j)$, $P_2(t_j)$. Then, extend $P_2(t_j)$ by adding an inter-cluster edge $(b(t_j), b'(t_j))$, where $b'(t_j)$ belongs to some other cluster $X_{i'}$. We refer to $b'(t_j)$ as the *entry point* of t_j to $X_{i'}$. We upper bound the total flow on any edge induced by this routing phase. If X_i is normal, then the total flow carried on an edge e is at most $2k$, because for every terminal $t_j \in X_i$ at most one of $P_1(t_j)$ and $P_2(t_j)$ passes through e. If X_i is star-like, the maximum flow per edge is $< k$, since the paths originating in one component of $H[X_i\backslash v_i]$ have no common edges with paths originating in another component. Finally, if X_i is twin with special nodes u, v, the maximum flow per edge is $< 3k$. However, note that the terminals contained in each small component of $H[X_i\backslash u]$ and $H[X_i\backslash v]$ send flow to a border node within that same component. Hence, regardless of how other terminals of X_i are routed, no more than $2k$ flow from this cluster passes through any single border node. The next lemma follows from the above.

Lemma 2.3. *In Phase 1, every intra-cluster edge of H carries at most $3k$ flow and every inter-cluster edge carries at most $4k$ flow ($2k$ from each cluster its endpoints belong to).*

Phase 2: Again, we examine each X_i individually, but now the focus is on *foreign* flow, i.e. flow that arrives in X_i from other clusters. Consider a spanning tree T of $H[X_i]$ and root it at v_i. We process the nodes in T in a bottom up fashion starting from the leaves; a node v is processed only after all its descendants in T have been processed.

When processing a node $w \not\equiv v_i$, let S be the set of terminals that send foreign flow to w. If that flow does not exceed $4k$, it is forwarded to w's parent $p(w)$, which means that the P_2 paths of the terminals in S are extended up to $p(w)$. Naturally, this flow is taken into consideration when processing $p(w)$.

Otherwise, consider the P_2 paths of the terminals in S, and in particular the path segments between the entry points to X_i and w. These segments define a tree T_w, which is a subgraph of T. Find the balanced separator x of T_w, assuming that the weight of a node equals the total demand of the terminals in S for which it is an entry point. If $x \not\equiv w$, re-route the paths along edges of T_w so that they all end up in x instead, which becomes an *auxiliary center*. Consequently, some paths are extended and others are contracted. Observe, though, that the total flow on each of the edges involved cannot increase, and in fact may decrease.

Finally, v_i is processed last. If it receives $< 4k$ foreign flow, we are done, else the aforementioned procedure is applied. Note that the auxiliary center thus created may coincide with v_i, but we treat them as separate entities to simplify the subsequent analysis.

Lemma 2.4. *In Phase 2, each intra-cluster edge of H carries at most $4k$ foreign flow.*

Phase 3: For a cluster X_i with center v_i, denote by $f(v_i)$ the total flow accumulated in v_i. Note that $f(v_i)$ includes flow coming from terminals of $X_i \backslash v_i$, foreign flow (which cannot exceed $4k$, see above), plus $\mathsf{dem}(v_i)$ if v_i is itself a terminal. Likewise, for an auxiliary center x we define $f(x)$ as the total foreign flow accumulated in x. Moreover, let $g(\cdot) = k\lceil f(\cdot)/k\rceil$, i.e. g is simply f rounded up to the nearest multiple of k.

Consider an instance I' of the node-protected single-sink problem on the graph G, with sink s and the cluster centers and auxiliary centers as terminals. The demand of a terminal t is given by $g(t)$, which is $2k$ or a larger multiple of k. In this case, the optimal solution to the problem can be found in polynomial time: for each terminal t, find the shortest cycle containing t and s, and then route flow along the two paths from t to s induced by the cycle. The network $\textsc{opt}_{\text{ss}}{}'$ built in this fashion has the minimum possible total *volume* (capacity $\times$ length, summed over all edges), so its cost is optimal even when splittable flows are allowed. The following lemma captures the cost of the routing.

Lemma 2.5. $\mathsf{cost}(\textsc{opt}_{\text{ss}}{}') < 21\,\mathsf{cost}(H) + 15\,\mathsf{cost}(\textsc{opt}_{\text{ss}})$.

Proof. We shall construct a hypothetical feasible solution $\textsc{sol}'$ to I', with cost not exceeding $21\,\mathsf{cost}(H) + 15\,\mathsf{cost}(\textsc{opt}_{\text{ss}})$. In $\textsc{sol}'$, each terminal sends flow to one or more of $t_1, t_2, \ldots, t_h$ (i.e. the terminals of the original instance), which is then routed to s. Let us examine cluster centers and auxiliary centers separately.

Suppose that X_i is a normal cluster. The center node v_i sends $2g(v_i)$ flow to terminals of X_i, so that no more than half the amount of that flow goes through the same node (except v_i and s), and the flow each terminal $t_j \in X_i$ receives is proportional to $\mathsf{dem}(t_j)$. Of course, if v_i is itself a terminal, it absorbs its own share of flow. Since v_i is the balanced separator of a spanning tree of $H[X_i]$, this routing can be achieved using edges of that tree, with the flow on any edge not exceeding $g(v_i) \leq 6k$. Furthermore, each terminal t_j receives at most $12\,\mathsf{dem}(t_j)$ flow, because $2g(v_i)/\,\mathsf{dem}(X_i) < 12$.

In case X_i is a twin cluster, $g(v_i) \leq 7k$ and $2g(v_i)/\,\mathsf{dem}(X_i) < 7$. Therefore, each terminal t_j receives at most $7\,\mathsf{dem}(t_j)$ flow and the flow on any edge is $\leq 7k$. If X_i is a star-like cluster, a similar argument applies. Note that $2g(v_i)/\,\mathsf{dem}(X_i) \leq 7$, so each terminal t_j receives at most $7\,\mathsf{dem}(t_j)$ flow. However, since every component of $H[X_i \backslash v_i]$ originally contained $< k$ demand, the flow on any edge is $< 7k$.

The last step is to route the flow that has accumulated at $t_1, t_2, \ldots, t_h$ to s. It is easy to see that this can be done with cost at most $12\,\mathsf{cost}(\textsc{opt}_{\text{ss}})$. Thus, we derive that:

Proposition 2.6. *The partial cost of $\textsc{sol}'$ due to cluster centers is $< 7\,\mathsf{cost}(H) + 12\,\mathsf{cost}(\textsc{opt}_{\text{ss}})$.*

Now, consider an auxiliary center x in some cluster X_i. Recall from the description of Phase 2 the definitions of S and T_w. In that phase, each edge e of T_w carried up to $\min\{4k, f(x)/2\}$ foreign flow to x. In $\textsc{sol}'$, the flow on e from x is $2g(x)/f(x)$ times as much, which turns out to be $< 9k$. Furthermore, take an intra-cluster edge e that carried flow destined for one or more auxiliary centers, located in clusters other than X_i. This flow could not exceed $2k$, and because $2g(x)/f(x) < \frac{5}{2}$ for any x, the flow on e in $\textsc{sol}'$ is $< 5k$. Hence, the total flow on an intra-cluster edge due to auxiliary centers does not exceed $9k + 5k = 14k$.

On the other hand, if e is an inter-cluster edge, it carried at most $4k$ flow to auxiliary centers, so in $\textsc{sol}'$ it has $< 10k$ flow. Finally, routing the flow to s has cost $< \lceil \frac{5}{2} \rceil\,\mathsf{cost}(\textsc{opt}_{\text{ss}}) = 3\,\mathsf{cost}(\textsc{opt}_{\text{ss}})$.

Proposition 2.7. *The partial cost of $\textsc{sol}'$ due to auxiliary centers is $< 14\,\mathsf{cost}(H) + 3\,\mathsf{cost}(\textsc{opt}_{\text{ss}})$.*

Propositions 2.6 and 2.7 finish the proof. $\square$

Lemmas 2.3, 2.4 and 2.5 imply that the total cost of our solution is at most $28\,\mathsf{cost}(H) + 15\,\mathsf{cost}(\textsc{opt}_{\text{ss}}) \leq 71\,\mathsf{cost}(\textsc{opt}_{\text{ss}})$. It is also relatively straightforward to verify its feasibility. Therefore,

Theorem 2.8. *The node-protected single-sink single-cable buy-at-bulk problem is $O(1)$ approximable.*

2.4 LP Relaxation and its Integrality Gap

So far, we have evaluated buy-at-bulk solutions under a *cable capacity* cost model. In other words, the cost of using edge e is $c_e\lceil b_e/k\rceil$, where k is the cable capacity, b_e is the flow on e and c_e is the cable cost for e. We now compare this model against a *fixed + incremental* cost model (FI), otherwise known as the *cost-distance* model [2, 26]. In the FI model, each edge e has a fixed cost c_e and an incremental cost ℓ_e; the cost of purchasing bandwidth b_e on e is given by $f_e(b_e) = c_e + \ell_e \cdot b_e$. When restricted to the single-cable case, the FI model specializes to having $\ell_e = c_e/k$ for each e; in other words, $f_e(b_e) = c_e(1+b_e/k)$. Since $c_e\lceil b_e/k\rceil \leq c_e(1 + b_e/k) \leq 2c_e\lceil b_e/k\rceil$, the cost of a solution under the single-cable FI model is within a factor of 2 of that under the cable capacity model.

Let us formulate a linear programming relaxation for the protected single-sink buy-at-bulk problem under the single-cable FI model. In the formulation, $x(e)$ is a variable that indicates whether or not edge e is in the solution; $\mathcal{Q}_i$ is the collection of simple cycles containing the sink s and the terminal $t_i \in \mathcal{T}$; $f(q)$ is a variable indicating whether flow from t_i is carried to s using the node-disjoint paths on the cycle $q \in \mathcal{Q}_i$; finally $\ell_q = \sum_{e\in q} \ell_e$ is the total length of the edges in q, where the edge length ℓ_e equals the incremental cost c_e/k per unit flow. Observe that the first term in the objective function (1a) represents the fixed cost, which depends only on which edges are used in the network, while the second term is the incremental cost, that is proportional to the flow carried by these edges.

$$\begin{aligned}
\mathsf{LP1}: \quad & \min \sum_{e\in E} c_e x(e) + \sum_{i=1}^{h} \mathsf{dem}(t_i) \sum_{q\in\mathcal{Q}_i} \ell_q f(q) && && \text{(1a)}\\
& \text{s.t.} \sum_{\substack{q\in\mathcal{Q}_i\\ q\ni e}} f(q) \leq x(e) && \forall e \in E,\ 1 \leq i \leq h && \text{(1b)}\\
& \sum_{q\in\mathcal{Q}_i} f(q) \geq 1 && \forall 1 \leq i \leq h && \text{(1c)}\\
& x(e), f(q) \geq 0 && \forall e \in E,\ q \in \textstyle\bigcup_i \mathcal{Q}_i && \text{(1d)}
\end{aligned}$$

Lemma 2.9. *The linear program $\mathsf{LP1}$ has an integrality gap of $O(1)$ for the single-cable FI cost model.*

Proof. Denote by $\textsc{opt}_{\mathsf{LP1}}$ the optimal solution to $\mathsf{LP1}$. Using the same arguments as in Propositions 2.6 and 2.7, but replacing $\textsc{opt}_{\text{ss}}$ with $\textsc{opt}_{\mathsf{LP1}}$, we can show that the FI cost of $\textsc{sol}'$ is bounded by (a) the FI cost of using the edges of H, with up to $21k$ units of flow on each edge, plus (b) the FI cost of using the edges of $\textsc{opt}_{\mathsf{LP1}}$, such that each edge carries up to 15 times the flow it carries in $\textsc{opt}_{\mathsf{LP1}}$. In other words, $\mathsf{cost}_{\text{FI}}(\textsc{sol}') < 22\,\mathsf{cost}(H) + 15\,\mathsf{cost}_{\text{FI}}(\textsc{opt}_{\mathsf{LP1}})$. Moreover, in the first two routing phases the algorithm uses edges of H, with up to $7k$ units of flow on each edge, so the corresponding FI cost is $< 8\,\mathsf{cost}(H)$. Thus, $\mathsf{cost}_{\text{FI}}(\textsc{sol}) < 30\,\mathsf{cost}(H) + 15\,\mathsf{cost}_{\text{FI}}(\textsc{opt}_{\mathsf{LP1}})$.

It remains to bound $\mathsf{cost}(H)$ in terms of $\mathsf{cost}_{\text{FI}}(\textsc{opt}_{\mathsf{LP1}})$. Recall that H is obtained by iterative rounding of the optimal solution to the following LP formulation of the node-connectivity version of the survivable network design problem [13].

$$\begin{aligned}
\mathsf{LP2}: \quad & \min \sum_{e\in E} c_e x(e) && && \text{(2a)}\\
& \text{s.t.} \sum_{e\in\delta(S,S')} x(e) \geq 2 - |V\backslash(S\cup S')| && \forall S, S' \subseteq V,\ S\cap S' = \emptyset,\ s\in S,\ \mathcal{T}\cap S' \neq \emptyset && \text{(2b)}\\
& 0 \leq x(e) \leq 1 && \forall e \in E && \text{(2c)}
\end{aligned}$$

Since constraints (1b)-(1d) imply (2b) and (2c), the value of the optimal solution to $\mathsf{LP2}$ is clearly a lower bound on $\mathsf{cost}_{\text{FI}}(\textsc{opt}_{\mathsf{LP1}})$, and $\mathsf{cost}(H)$ is at most twice that. Hence, $\mathsf{cost}(H) \leq 2\,\mathsf{cost}_{\text{FI}}(\textsc{opt}_{\mathsf{LP1}})$ and $\mathsf{cost}_{\text{FI}}(\textsc{sol}) < 75\,\mathsf{cost}_{\text{FI}}(\textsc{opt}_{\mathsf{LP1}})$. $\square$

3 From Single-Sink to Multi-Commodity

In this section we consider the node-protected multi-commodity buy-at-bulk problem. We establish that an α approximation for the single-sink problem implies an $O(\alpha \log^2 h \log D)$ approximation for the multi-commodity problem via a natural LP relaxation, where $D = \sum_i \mathsf{dem}(i)$. Note that this result applies to the general FI model, i.e. without the single-cable restriction $\ell_e = c_e/k$ that was introduced in Section 2.4. This is significant because the general FI model is essentially equivalent to the non-uniform model. However, in the single-cable model that is of interest here, the dependence on D can be removed and the ratio becomes $O(\alpha \log^3 h)$. The results in Section 2.4 imply that $\alpha = O(1)$ for the single-cable model, and thus we obtain an $O(\log^3 h)$ approximation for the multi-commodity single-cable problem. To simplify the exposition, throughout this section we assume unit demands ($\mathsf{dem}(i) = 1$ for $1 \leq i \leq h$) and prove the ratio of

$O(\log^3 h)$ in this setting. The extension to the general case of arbitrary demands is omitted.

We use the algorithmic paradigm from [7], as outlined in Section 1. The main technical ingredient is an extension of the *junction tree* concept from [7]. We define a structure which we call a *junction-structure*, more precisely a two-node junction-structure, as shown below.

To begin with, let us formulate the objective function for the multi-commodity problem in the general FI model. Recall that c_e and ℓ_e are the fixed and incremental cost (= length) of e. Given two nodes a, b and a subgraph H of G, we let $\ell 2_H(a, b)$ be the minimum-length cycle of H containing a and b. Note that this is the same as the minimum length of two node-disjoint paths between a and b in H. Then the objective function is to find $E' \subseteq E$ that minimizes $\sum_{e \in E'} c_e + \sum_{s_i t_i \in \mathcal{T}} \ell 2_{G[E']}(s_i, t_i)$.

A *two-node junction* is a pair of nodes (u, v) with $u \neq v$. We say that a node x is *two-connected to a junction* (u, v) in a graph H if there exist paths P and Q in H that connect x to u and x to v, respectively, and are node-disjoint (except at x). Denote by $\ell 2_H(x, (u, v))$ the minimum total length of two such paths. The following is straightforward to verify.

Proposition 3.1. *Let H be a graph in which s_i and t_i are two-connected to a junction (u, v) and there exists a cycle containing u and v. Then there is a cycle in H containing s_i and t_i of length at most $\ell 2_H(s_i, (u, v)) + \ell 2_H(t_i, (u, v)) + \ell 2_H(u, v)$.*

Given a subset A of the demands, a *junction-structure for A* rooted at a two-node junction (u, v) is a subgraph $H(u, v)$ of G satisfying the requirements of Proposition 3.1 for every s_i and t_i such that $s_i t_i \in A$. Hence, we can connect the pairs in A using edges of $H(u, v)$, with cost no more than

$$\sum_{e \in E(H(u,v))} c_e + \sum_{s_i t_i \in A} \big(\ell 2_{H(u,v)}(s_i, (u, v)) + \\ + \ell 2_{H(u,v)}(t_i, (u, v)) + \ell 2_{H(u,v)}(u, v) \big) . \quad (3)$$

Quantity (3) is called – somewhat abusively – the *cost* of junction-structure $H(u, v)$.

Given a multi-commodity instance with unit demands, let OPT_{MC} be the optimal solution. We first show the existence of a junction-structure of density $O\big(\frac{\log h}{h}\big) \operatorname{cost}(\text{OPT}_{\text{MC}})$, where density is defined to be the ratio of the cost of the junction-structure to the number of demand pairs connected by it. Although this existence proof builds upon the ideas in [7], to ensure node-disjointness we need a more sophisticated argument in Lemma 3.4. Using the $O(1)$ integrality gap of the single-sink problem, we further show how to find a junction-structure whose density is at most $O(\log h)$ times the optimal density, namely a structure of density at most $O\big(\frac{\log^2 h}{h}\big) \operatorname{cost}(\text{OPT}_{\text{MC}})$. We now remove the demands whose source-destination nodes are connected and recurse on the remaining ones. This gives us an approximation ratio of $O(\log^3 h)$ for the protected multi-commodity buy-at-bulk problem.

3.1 Existence of Low-Density Junction-Structure

To show the existence of a junction-structure with low density, we assume knowledge of an optimal solution E^* to the given multi-commodity instance and find a low-density junction-structure from E^*. Let $G^* = G[E^*]$ be the graph induced on the edge set E^*. Let $L = \sum_i \ell 2_{G^*}(s_i, t_i)/h$ be the *average length* of the demand pairs in the optimal solution. A demand $s_i t_i$ is *short* if $\ell 2_{G^*}(s_i, t_i)$ is at most $2L$. By Markov's inequality, most demands are short:

Proposition 3.2. *At least $h/2$ demands are short.*

From now on we focus on these short demands only. For each demand pair $s_i t_i$, we fix a shortest cycle Q_i through s_i and t_i in G^*. Subsequently, we present an algorithm that decomposes G^* into connected *edge-disjoint*[1] induced subgraphs $G_1^* = G[E_1^*], G_2^* = G[E_2^*], \ldots, G_f^* = G[E_f^*]$. For a subgraph H of G^*, we define a ball $B_H((u, v), r)$ with center (u, v) and radius r to contain vertices $x \in V(H)$ for which $\ell 2_H(x, (u, v)) \leq r$. We abuse notation and use $B_H((u, v), r)$ also to denote the induced subgraph. A demand pair $s_i t_i$ is *captured* by a ball $B_H((u, v), r)$ if both s_i and t_i are contained in the ball. A pair $s_i t_i$ *intersects* $B_H((u, v), r)$ if it is not captured by the ball, but the ball contains some edge in the cycle Q_i.

We choose a short demand pair uv as center (u, v) and define a sequence of radii $r_j = 2L \cdot j$ for $j \geq 1$. We begin with the ball $B_{G^*}((u, v), r_1)$; if the number of captured demands is at least the number of intersected demands, we make the ball the first component G_1^*. Otherwise, the number of captured demands is fewer than the number of intersected demands. In this case, we consider progressively larger balls, of radii r_2, r_3, and so on. Let J be the smallest index such that the number of demands captured by $B_{G^*}((u, v), r_J)$ is fewer than the number of demands intersected by the same ball. Then, $B_{G^*}((u, v), r_J)$ becomes the first component G_1^*. We remove all *edges* in $B_{G^*}((u, v), r_J)$ from G^* and all demands that are either captured or intersected by that ball; these intersected demands are henceforth considered *lost*. We recurse on the residual of G^* and the remaining demands to create components $G_2^*, G_3^*, \ldots, G_f^*$, until no demands are left. We let T_i be the set of demands captured by the component G_i^*, and let (u_i, v_i) denote the center we have arbitrarily chosen for G_i^*. Since lost demands are fewer than captured demands, the following lemma also holds.

[1] This is in contrast to the 1+1 edge-protection case where the subgraphs can be chosen to be node-disjoint. The node-disjoint property is relevant for the buy-at-bulk problem with node costs [8].

Lemma 3.3. *The total number of demands that are captured by $G_1^*, G_2^*, \ldots, G_f^*$ is at least $h/4$.*

We show below that one of the components corresponds to a low-density junction-structure. The construction of the ball-growing algorithm has the following property.

Lemma 3.4. *Any demand intersected by the ball $B_{G^*}((u,v),r_j)$ is captured by $B_{G^*}((u,v),r_{j+1})$.*

Proof. Assume that $B_{G^*}((u,v),r_j)$ intersects the demand pair $s_i t_i$. We establish that $\ell 2_{G^*}(s_i,(u,v)) \leq r_{j+1}$. (The proof for t_i is identical.) Start from s_i and move in one direction along Q_i. Denote by x the first node in $B_{G^*}((u,v),r_j)$ thus encountered, and by $P_{s_i x}$ the segment of Q_i traversed. Then, go back to s_i and move in the opposite direction along Q_i. Denote by y the first node in $B_{G^*}((u,v),r_j)$ encountered, and by $P_{s_i y}$ the segment of Q_i traversed. Note that, x and y are distinct; since Q_i and $B_{G^*}((u,v),r_j)$ share an edge, at least two nodes on Q_i are in $B_{G^*}((u,v),r_j)$. Also, by construction, $P_{s_i x}$ and $P_{s_i y}$ are node-disjoint. Let P_{xu} and P_{xv} be the two node-disjoint paths from x to u and v, respectively, whose combined length is at most r_j. Similarly, let P_{yu} and P_{yv} be the two node-disjoint paths from y to u and v, respectively, whose combined length is at most r_j. By the choice of x and y, the paths P_{xu}, P_{xv}, P_{yu} and P_{yv} are node-disjoint from $P_{s_i x}$ and $P_{s_i y}$, other than at x and y. We distinguish the following two cases.

Case 1: $x \notin P_{yu} \cup P_{yv}$ and $y \notin P_{xu} \cup P_{xv}$ (see Figure 3, top). Let F be the subgraph induced on $Q_i \cup P_{xu} \cup P_{xv} \cup P_{yu} \cup P_{yv}$. Add a dummy node α to F, such that α is adjacent to u and v only, via zero-length edges. Then, create a single-sink min-cost flow problem on F, where two units of flow are sent from s_i to α. Assume that each node except s_i and α has unit capacity and zero cost, and each edge has cost equal to its length.

Consider the following fractional solution: s_i sends one unit of flow to x along $P_{s_i x}$ and one unit to y along $P_{s_i y}$; x sends $1/2$ units of flow to u along P_{xu} and $1/2$ units to v along P_{xv}; and y sends $1/2$ units of flow to u along P_{yu} and $1/2$ units to v along P_{yv}. It is easy to see that node capacities are not exceeded and the routing cost is at most

$$\frac{1}{2}\ell(P_{xu} \cup P_{xv}) + \frac{1}{2}\ell(P_{yu} \cup P_{yv}) + \ell(Q_i) \leq \leq \frac{1}{2}r_j + \frac{1}{2}r_j + 2L = r_{j+1} \ . \quad (4)$$

The integrality of single-sink min-cost flow implies the existence of two node-disjoint integral paths between s_i and α, of total cost no more than the fractional cost given in (4).

Case 2: Otherwise, without loss of generality, suppose that P_{yu} goes through x (see Figure 3, right); the other subcases are symmetrical. P_{yv} and the segment of P_{yu} between x and u yield two node-disjoint paths from x to u and

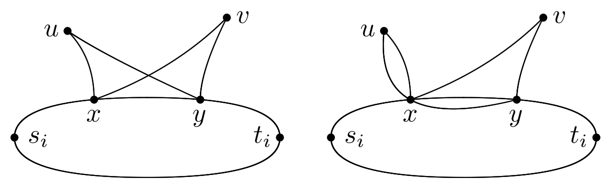

Figure 3. Existence of two node-disjoint paths from s_i (symmetrically t_i) to u and v.

from y to v, whose combined length is less than r_j. Consequently, there exist two node disjoint paths from s_i to u and v, whose total length is at most $\ell(P_{yu}) + \ell(P_{yv}) + \ell(Q_i) \leq r_j + 2L = r_{j+1}$. □

Lemma 3.5. *For $1 \leq i \leq f$ and every node x in G_i^*, $\ell 2_{G_i^*}(x,(u_i,v_i)) \leq 2L \cdot (1 + \log h)$. In particular, for each demand st captured by G_i^*, $\ell 2_{G_i^*}(s,(u_i,v_i)) + \ell 2_{G_i^*}(t,(u_i,v_i)) + \ell 2_{G_i^*}(u_i,v_i) \leq 2L(3 + 2\log h)$.*

Proof. Let H be the residual graph of G^* after the first $i-1$ components $G_1^*, \ldots, G_{i-1}^*$ are created and their edges removed. From Lemma 3.4 and the construction of the algorithm, every time the ball grows from $B_H((u_i,v_i),r_j)$ to $B_H((u_i,v_i),r_{j+1})$ the number of demands captured by $B_H((u_i,v_i),r_{j+1})$ is at least twice the number captured by $B_H((u_i,v_i),r_j)$. Since the total number of pairs is h, the number of times the radius is increased is at most $\lceil \log h \rceil$. Thus, G_i^* has radius at most $2L \cdot (1 + \log h)$. Clearly, then,

$$\ell 2_{G_i^*}(s,(u_i,v_i)) + \ell 2_{G_i^*}(t,(u_i,v_i)) + \ell 2_{G_i^*}(u_i,v_i) \leq \leq 2 \cdot 2L(1 + \log h) + 2L = 2L(3 + 2\log h) \ . \quad \square$$

Lemmata 3.3 and 3.5, and the fact that the subgraphs $G_1^*, \ldots, G_f^*$ are edge-disjoint, when combined with a simple averaging argument lead to the following theorem.

Theorem 3.6. *Given a multi-commodity instance of the protected buy-at-bulk problem, there exists a junction-structure of density $O\left(\frac{\log h}{h}\right) \mathsf{cost}(\mathrm{OPT}_{\mathrm{MC}})$.*

3.2 Finding a Low-Density Junction-Structure

Using the single-sink single-cable approximation algorithm as a subroutine, an $O(\log h)$ approximation to the minimum-density junction-structure can be derived. This is a consequence of the theorem below.

Theorem 3.7. *There is an $O(\log h)$ approximation for the min-density protected single-sink single-cable problem.*

Proof. We closely follow the argument used in [7]. First, let us formulate a linear programming relaxation for the density version of the protected single-sink problem. In other words, the objective is to minimize the ratio of the cost of the network to the number of terminals it connects to the sink. For each terminal t_i, we introduce a variable y_i that indicates whether or not t_i is connected to s in the solution. By normalizing the sum $\sum_{i=1}^h y_i$ to 1, we ensure that the objective function represents the density of the solution.

$$\mathsf{LP}3: \quad \min \quad \sum_{e\in E} c_e x(e) + \sum_{i=1}^{h} \sum_{q\in\mathcal{Q}_i} \ell(q) f(q) \qquad (5a)$$

$$\text{s.t.} \quad \sum_{\substack{q\in\mathcal{Q}_i \\ q\ni e}} f(q) \le x(e) \qquad \forall e\in E,\ 1\le i\le h \qquad (5b)$$

$$\sum_{q\in\mathcal{Q}_i} f(q) \ge y_i \qquad \forall 1\le i\le h \qquad (5c)$$

$$\sum_{i=1}^{h} y_i = 1 \qquad (5d)$$

$$x(e), f(q), y_i \ge 0 \qquad \forall e\in E,\ q\in \bigcup_i \mathcal{Q}_i,\ 1\le i\le h \qquad (5e)$$

Lemma 3.8. *The linear program* LP3 *is a valid relaxation for the min-density protected single-sink problem. It can be solved optimally in polynomial time.*

We then partition the demands into groups $\mathcal{T}_a$, where $a \in [0..\lceil \log h \rceil]$, depending on the values of the corresponding y_i variables in the optimal solution to LP3, which can be computed efficiently. Let $\mathcal{T}_a = \{t_i \mid y_{\max}/2^{a+1} \le y_i < y_{\max}/2^a\}$, where $y_{\max} = \max_i y_i$. Among these groups there exists one group $\mathcal{T}_b$ such that $\sum_{t_i\in\mathcal{T}_b} y_i = \Omega(1/\log h)$. Moreover, $2^b/(y_{\max}|\mathcal{T}_b|) = O(\log h)$.

Finally, we solve the protected single-sink problem on $\mathcal{T}_b$ using the approximation algorithm from Section 2, and claim that the resulting solution is an $O(\log h)$ approximation to the min-density protected single-sink problem. Indeed, let α be the value of the optimal solution to LP3. For terminals in $\mathcal{T}_b$, we may obtain a feasible solution to LP1 by scaling up the solution to LP3 by a factor of $\beta = 2^{b+1}/y_{\max}$. The cost of this solution is at most $\alpha\beta$. By the proof of Lemma 2.9, the algorithm of Section 2 can yield an integral solution, for terminals in $\mathcal{T}_b$, of value $O(\alpha\beta)$. Its density is $O(\alpha\beta)/|\mathcal{T}_b|$, which is $O(\log h)\alpha$, by the choice of b. Since α is a lower bound on the density of the optimal solution, the proof is complete. □

We now describe how to approximate the minimum-density junction-structure. Once again, the ideas are similar to those in [7], but the details are more elaborate. The step of guessing the junction (u, v) of a min-density junction-structure is implemented, as is standard, by trying each possible pairs of nodes as a candidate junction, and keeping the best result.

Then, we relax this problem to an LP very similar to that for the min-density single-sink problem. Create a new graph G' by adding an artificial sink node r to the graph G and connecting it to u and v via edges ur, vr such that $\ell_{ur} = \ell 2_G(u,v)$, $c_{ur} = k \cdot \ell_{ur}$, and $c_{vr} = \ell_{vr} = 0$. Assume, without loss of generality, that each node in the original graph G is the endpoint of at most one demand pair in $\mathcal{T}$. Consider LP3 on G', with r as sink and $\mathcal{T}' = \{s_1, t_1, s_2, t_2, \ldots, s_h, t_h\} = \{t'_1, t'_2, \ldots, t'_{2h}\}$ as the set of terminals. Moreover, place an additional set of constraints in LP3:

$$y_i = y_j \qquad \forall i, j \text{ such that } \exists p \text{ with } t'_i = s_p \text{ and } t'_j = t_p$$

Suppose that the minimum-density junction-structure OPT^* has density γ^*. It is straightforward to convert OPT^* to a feasible solution of this new linear program, with density between $\frac{1}{2}\gamma^*$ and γ^*; it may not be exactly $\frac{1}{2}\gamma^*$, because the fixed cost of some junction-structure edges may be double-counted in the objective function of the LP.

Apply the algorithm from Theorem 3.7 to the optimal solution of the modified LP3 above. Observe that the rounding procedure ensures that for any i, j such that $y_i = y_j$, either both t'_i and t'_j are connected to r, or neither is. Thus, the resulting solution SOL^* to the single-sink density problem can be converted back to a junction-structure. By Theorem 3.7, SOL^* has density $O(\log h)\gamma^*$, so the corresponding junction-structure also has density $O(\log h)\gamma^*$. Combined with Theorem 3.6, this yields

Theorem 3.9. *Given a multi-commodity instance of the protected single-cable buy-at-bulk problem, there is a polynomial time algorithm that finds a junction-structure with density* $O\left(\frac{\log^2 h}{h}\right) \mathsf{cost}(\text{OPT}_{\text{MC}})$.

As mentioned before, we use an iterative greedy algorithm, similar to the classic one for set cover. In each iteration, we use Theorem 3.9 to find an approximate junction structure in the residual instance and remove the demand pairs that are connected by the structure to obtain the residual instance for the next iteration. Hence,

Theorem 3.10. *The node-protected multi-commodity single-cable buy-at-bulk problem can be approximated by a factor of* $O(\log^3 h)$.

Recall that in this section all demands have $\mathsf{dem}(i) = 1$. In the single-cable setting, Theorem 3.10 holds for arbitrary demands as well; we omit the details.

Remark. The reduction from the min-density junction structure problem to the single-sink one can be extended to the general FI model. Consequently, if a bound on the integrality gap of LP1 for either the uniform or the non-uniform cost model were known, via a constructive result analogous to Lemma 2.9, we could generalize Theorems 3.7 and 3.9 – and, ultimately, Theorem 3.10 – accordingly.

4 Conclusions

Our main contributions are the formal introduction of the protected buy-at-bulk network design problem and the first approximation algorithms for it in the single-cable setting. We remark that for edge-disjoint protection, the ideas in Section 3 simplify and hold for connectivity requirements larger than two. One important question is the approximability of the protected buy-at-bulk problem in the uniform multiple-cable and non-uniform cost models. Our results in Section 3 pertaining to the general FI model indicate that it is sufficient to focus on the single-sink version of the problem. We note that even the case of two cables is of interest, as current techniques appear inadequate for tackling it. From the practical perspective, we hope the concepts of clustering and junction structures may inspire the design of more effective heuristics for the design of real-world DWDM networks.

References

[1] M. Andrews. Hardness of buy-at-bulk network design. *Proc. of IEEE FOCS*, 115–124, 2004.

[2] M. Andrews and L. Zhang. The access network design problem. *Proc. of IEEE FOCS*, 40–49, 1998.

[3] B. Awerbuch and Y. Azar. Buy-at-bulk network design. *Proc. of IEEE FOCS*, 542–547, 1997.

[4] Y. Bartal. On approximating arbitrary metrics by tree metrics. *Proc. of ACM STOC*, 161–168, 1998.

[5] M. Charikar and A. Karagiozova. On non-uniform multicommodity buy-at-bulk network design. *Proc. of ACM STOC*, 176–182, 2005.

[6] C. Chekuri, P. Claisse, R. Essiambre, S. Fortune, D. Kilper, W. Lee, K. Nithi, I. Saniee, B. Shepherd, C. White, G. Wilfong, and L. Zhang. Design tools for transparent optical networks. *Bell Labs Technical Journal*, 11(2):129–143, 2006.

[7] C. Chekuri, M. Hajiaghayi, G. Kortsarz, and M. Salavatipour. Approximation algorithms for non-uniform buy-at-bulk network design problems. *Proc. of IEEE FOCS*, 677–686, 2006.

[8] C. Chekuri, M. Hajiaghayi, G. Kortsarz, and M. Salavatipour. Approximation algorithms for node-weighted buy-at-bulk network design. *Proc. of ACM-SIAM SODA*, 1265–1274, 2007.

[9] C. Chekuri, S. Khanna, and J.S. Naor. A deterministic algorithm for the cost-distance problem. *Proc. of ACM-SIAM SODA*, 232-233, 2001.

[10] C. Chekuri, S. Khanna, and F.B. Shepherd. The all-or-nothing multicommodity flow problem. *Proc. of ACM STOC*, 156–165, 2004.

[11] J. Chuzhoy, A. Gupta, J. Naor, and A. Sinha. On the approximability of some network design problems. *Proc. of ACM-SIAM SODA*, 943–951, 2005.

[12] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *JCSS*, 69:485–497, 2004.

[13] L. Fleischer, K. Jain, and D.P. Williamson. An iterative rounding 2-approximation algorithm for the element connectivity problem. *JCSS*, 72(5):838–867, 2006.

[14] A. Goel and D. Estrin. Simultaneous optimization for concave costs: Single sink aggregation or single source buy-at-bulk. *Proc. of ACM-SIAM SODA*, 499–505, 2003.

[15] M. Goemans and D. Williamson. A general approximation technique for constrained forest problems. *SIAM J. Comput.*, 24(2):296–317, 1995.

[16] S. Guha, A. Meyerson, and K. Munagala. A constant factor approximation for the single sink edge installation problem. *Proc. of ACM STOC*, 383–388, 2001.

[17] A. Gupta, A. Kumar, M. Pál, and T. Roughgarden. Approximation via cost-sharing: A simple approximation algorithm for the multicommodity rent-or-buy problem. *Proc. of IEEE FOCS*, 606–615, 2003.

[18] A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximations for network design. *Proc. of ACM STOC*, 365–372, 2003.

[19] M. Hajiaghayi, G. Kortsarz, and M. Salvatipour. Polylogarithmic approximation algorithm for non-uniform multicommodity buy-at-bulk. ECCC Report TR06-007, 2005.

[20] D. Hochbaum. *Approximation algorithms for NP-hard problems*. PWS Publishing Company, 1997.

[21] K. Jain. A factor 2 approximation algorithm for the generalized steiner network problem. *Combinatorica*, 21(1):39–60, 2001.

[22] R. Jothi and B. Raghavachari. Improved approximation algorithms for the single-sink buy-at-bulk network design problem. In *Proc. of SWAT*, 336–348, 2004.

[23] G. Kortsarz and Z. Nutov. Approximating some network design problems with node costs. Manuscript, April 2007.

[24] A. Kumar, A. Gupta, and T. Roughgarden. A constant-factor approximation algorithm for the multicommodity rent or buy problem. *Proc. of IEEE FOCS*, 333–342, 2002.

[25] A. McGregor and B. Shepherd. Island hopping and path colouring, with applications to WDM network design. *Proc. of ACM-SIAM SODA*, 864–873, 2007.

[26] A. Meyerson, K. Munagala, and S. Plotkin. Cost-distance: Two metric network design. *Proc. of IEEE FOCS*, 383–388, 2000.

[27] R. Ramaswami and K.N. Sivarajan. Optical Networks: A Practical Perspective. Morgan Kaufmann, 2nd edition, 2001.

[28] F.S. Salman, J. Cheriyan, R. Ravi, and S. Subramanian. Buy at bulk network design: Approximating the single-sink edge installation problem. *SIAM J. on Optimization*, 11(3):595–610, 2000.

[29] A. Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Springer-Verlag, Berlin Heidelberg, 2003.

[30] A. Somani. *Survivability and Traffic Grooming in WDM Optical Networks*. Cambridge University Press, 2006.

[31] T.E. Stern, K. Bala. *Multiwavelength Optical Networks: A Layered Approach*. Addison-Wesley, 1999.

[32] V. Vazirani. *Approximation algorithms*. Springer, 1994.

[33] D.P. Williamson, M.X. Goemans, M. Mihail and V.V. Vazirani. A primal-dual approximation algorithm for generalized Steiner network problems. *Combinatorica*, 15:534–454, December 1995.