Approximability of Robust Network Design

Neil Olver, F. Bruce Shepherd
Department of Mathematics and Statistics, McGill University, Montreal, QC, Canada.
Email: olver@math.mcgill.ca, bruce.shepherd@mcgill.ca

Abstract

We consider robust network design problems where the set of feasible demands may be given by an arbitrary polytope or convex body more generally. This model, introduced by Ben-Ameur and Kerivin [2], generalizes the well studied virtual private network (VPN) problem. Most research in this area has focused on finding constant factor approximations for specific polytope of demands, such as the class of hose matrices used in the definition of VPN. As pointed out in [4], however, the general problem was only known to be APX-hard (based on a reduction from the Steiner tree problem). We show that the general robust design is hard to approximate to within logarithmic factors. We establish this by showing a general reduction of buy-at-bulk network design to the robust network design problem.

In the second part of the paper, we introduce a natural generalization of the VPN problem. In this model, the set of feasible demands is determined by a tree with edge capacities; a demand matrix is feasible if it can be routed on the tree. We give a constant factor approximation algorithm for this problem that achieves factor 8 in general, and 2 for the case where the tree has unit capacities.

1 Introduction

Robust network design considers the problem of designing a network, by buying bandwidth on some underlying undirected graph $G = (V, E)$, in order to support some *set* of *valid* demands (demands are specified by matrices D_{ij}). We are primarily concerned with minimizing the cost of the network which is determined by $\sum_{e \in E} c(e)x(e)$ where $x(e)$ is the bandwidth assigned to edge e, and $c(e)$ is a per-unit cost. The set of valid demands consists of some convex body, usually a polytope, which we denote by $\mathcal{U}$ (for universe). In addition, we require that the $\mathcal{U}$ is *separable*, i.e., there is a polytime algorithm to solve its separation problem. We mention that design criteria besides total cost are also possible. For instance, designing the network to have minimum maximum edge congestion, has been the focus of intense study dating back to classical work on parallel computation (e.g. [19]) and reaching to the recent breakthrough of Räcke [17]. In the following, we use the term *robust network design* problem to refer to the cost version.

Since our network must potentially satisfy multiple demand matrices, we must also define how routings are allowed to change if demands change. We can obviously achieve the cheapest network if routings are allowed to be fractional and may be recomputed for each new demand matrix; call this the *dynamic robust network design problem*. However, this is not practical in a data network setting for instance. Instead we restrict attention to the *oblivious* routing model, where each pair $i, j \in V$ must fix a path P_{ij} ahead of time; this specifies what we call the *flow template*). Subsequently, if we must support a matrix D, then for each i, j we send D_{ij} units of flow along each P_{ij}. For such a template, the capacity required on edge e, is then the maximum load put on e by one of the valid matrices in $\mathcal{U}$. Hence the optimization problem is to find a template for which the sum of these maximum loads is minimized.

While research on robust network design has focused on special classes of demand universes [7, 12, 13] and on establishing constant time approximation guarantees for these classes (some of which we describe below), as pointed out in [4] the approximability status for robust design with general polytopes has not been resolved. Our first result shows that it is hard to approximate the optimal solution to within a logarithmic factor. This follows from a reasonably broad connection to buy-at-bulk network design described in Section 2. Previously, robust design was known to be APX-hard, but our hardness result is tight to within polylog factors since an $\tilde{O}(\log n)$ factor approximation is known based on metric tree embeddings ([11] and cf. [4, 9]). Our second main result gives a constant factor approximation for a generalization of a well-studied class of demands: the hose matrices. We discuss this further now.

Positive results in robust network design have been given for relatively few demand matrix universes. One universe of central interest [7, 12, 5, 6] is the class of *hose matrices*, $\mathcal{H} = \{D_{ij} : \sum_j D_{ij} \le b(i),\ \forall\ i \in V\}$. Here, $b(i)$ is called the *marginal* for node i; in other words, it gives an upper bound on the amount of traffic allowed to terminate at this node. The demands are viewed as undirected: it is assumed $D_{ij} = D_{ji}$ and this represents a single demand between i, j. The *virtual private network* (VPN) problem is the minimum cost robust network design problem for the class $\mathcal{H}$.

The VPN problem was shown to have a 2-approximation algorithm independently in [7, 12]. The algorithm works as follows. Consider taking each node $w \in V$ as a potential *hub node*. Then for each other node v, route $b(v)$ units of flow to w on a shortest path tree T centered at w; pick the cheapest of such solutions. It is clear that the resulting capacitated tree (call it the *VPN tree*) is a feasible solution. It was conjectured [15] that the optimal solution was in fact such a tree (the VPN problem would then be polytime solvable). This was resolved positively in [8], building on work in [10] (the conjecture had earlier been proved for ring networks and some other special cases [14]).

Our second result is a constant factor approximation algorithm for a new class of demand matrices called *tree demands* which generalizes the class of hose matrices. For an edge-capacitated tree T with leaf set $W \subseteq V$, we denote by $\mathcal{H}_T$ the set of matrices D_{ij} for which the demands can be routed in T. That is, routing D_{ij} units between i, j in T for each $i < j$, should not overload the capacity, denoted by b_e, of any edge $e \in T$. Note that if we take T to be a star with center node v, and the capacity of each edge iv to be $b(i)$, then the set of tree demands is just the class of hose matrices. This has another interpretation in terms of cuts. Any demand matrix D can be alternatively specified by a weighted complete graph on the terminals, with edge uv having weight D_{uv}; we call this the *demand graph*. The VPN model can be interpreted as imposing singleton cut constraints on this graph: we must be able to route all demands such that for any $u \in W$, the weight of the cut $\delta(\{u\})$ in the demand graph does not exceed its marginal $b(u)$. It is natural to study universes defined by more general cut families; each cut in a given family has a maximum capacity and a demand is valid, as long as it does not violate any of these cut constraints. Tree demands correspond exactly to the case where the cuts form a nested family. These extra cut constraints could be used to more accurately define the requirements of a network, such as a VPN, possibly yielding a cheaper final network.

In Section 3 we describe an algorithm which computes a routing template that induces a network whose cost is at most 8 times the optimal robust design for $\mathcal{H}_T$; this is improved to a tight factor 2 for the unit capacity case. In fact, the proofs imply something stronger: given the optimal network that supports each tree demand via *dynamic* fractional routing, we can find an oblivious routing which costs no more than 8 times as much (or twice in the unit capacity case).

Other related work. In the *asymmetric* hose model (which can be shown to strictly generalize the symmetric model via a simple transformation), terminals are divided into *senders* and *receivers*. In addition to hose constraints as in the symmetric version, all demand goes between senders and a receivers, i.e. $D_{ij} = 0$ if i and j are both senders or both receivers. This problem (unlike the symmetric version) is NP-hard; a randomized constant-approximation algorithm for the robust design problem for the class of asymmetric hose matrices is given in [13]. Note that while the asymmetric hose model generalizes the symmetric one, it is not comparable to the class of tree demands considered in this paper.

The original use of hose matrices was primarily motivated by an application in data networks. An operator wants to build a subnetwork for a customer with sites at several nodes i, each which may offer up to $b(i)$ units of capacity. In [18], an application to optical networking is described. They first remark that the capacitated VPN tree described above, has enough edge capacity to route all the hose matrices with an oblivious routing template that is more wasteful than the tree template. The *hub template* defines the path P_{ij} as the union of i's path to the root (or hub) w with j's path to w. (The path may be nonsimple.) The motivation for a hub template is that one may now set up cost-efficient optical fixed paths from each terminal to the hub w, avoiding expensive routing equipment. Typically several hubs are chosen to avoid a single point of failure. Intuitively, the best choice of hub(s) consists of nodes in the center of a network, but this requires that all traffic, local or not, must route to a centralized region. In [18] it is left open to compare the costs of routing architectures based on some form of "hierarchical hubbing". One possible algorithm needed for such a comparison is a simple extension of the hierarchical hubbing subroutine used in the present paper (see Section 3.2).

1.1 Model and definitions

A general instance of *polyhedral (or convex body) robust network design* consists of an undirected graph $G = (V, E)$ where each edge $e \in E$ has an associated nonnegative cost c_e. This represents the per-unit cost of bandwidth reserved on that edge. In the remainder, we use $d_G(u, v)$ (or just $d(u, v)$) to denote the length of a shortest path between u and v using the cost vector c.

Throughout we may assume that our demand matrices $D_{ij} \in \mathcal{U}$ are symmetric and in fact $D_{ij} = D_{ji}$ represents a single undirected demand between i, j. (Alternatively, one could simply work with lower triangular D.) As described above for the hose model, we generally assume that for each node v there is a marginal $b(v) \geq 0$ which is always an upper bound on the total demand which may terminal at v. We may refer to nodes with $b(v) > 0$ as *terminals*, and denote the set of terminals by W.

2 An equivalence between robust and buy-at-bulk network design

In this section, we show that a large class of uniform buy-at-bulk network design problems can be simulated by a robust network design problem over a separable polytope of demand matrices. This will imply that it is hard to approximate the robust network design problem for a general polytope $\mathcal{P}$ within a factor of $\Omega(\log^{1/4-\epsilon} n)$ for any $\epsilon > 0$, assuming that $NP \not\subset ZPTIME(n^{\text{polylog}\, n})$. We are given an undirected graph G with nonnegative edge lengths c_e, as well as a single nonnegative, increasing and concave price function f, with $f(0) = 0$. A number of demand pairs $s_1t_1, s_2t_2, \ldots, s_kt_k$ are also given. A solution must reserve enough capacity on each edge so that all the demand pairs may route simultaneously along selected paths P_i between s_i, t_i. The cost of an edge e in the solution, however, is given by $c_e f(x_e)$, where x_e is the number of P_i's containing e. Since f is concave, buying a large capacity on a single edge may be much cheaper than buying small capacities on many edges.

Andrews [1] showed that even on undirected graphs, this uniform buy-at-bulk problem is hard to approximate; in particular, it cannot be approximated within a ratio of $\Omega(\log^{1/4-\epsilon} n)$ for any $\epsilon > 0$, unless $NP \subset ZPTIME(n^{\text{polylog}\, n})$.

We begin with an instance of uniform buy-at-bulk. From this, we construct an instance of robust network design with a polytope that can be described very simply, and separated in polynomial time.

Let Π be the set of permutation of the integers 1 through k, and let π be any such permutation. For notational convenience, we also define $\pi(0) = 0$. Define the demand matrix D^π by

$$D^\pi_{uv} = \begin{cases} f(\pi(i)) - f(\pi(i) - 1) & \text{if } \{u, v\} = \{s_i, t_i\} \text{ for some } 1 \le i \le k \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

Now define the polytope $\mathcal{P}$ as

$$\mathcal{P} := \text{conv}\{D^\pi : \pi \in \Pi\}. \tag{2}$$

Theorem 2.1. *The buy-at-bulk problem on graph G with lengths c_e and cost function $f(\cdot)$, has the same optimum as the robust optimization problem on the same instance where $\mathcal{P}$ is used for the demand polytope. In addition, the optimal routings are the same.*

Proof. Consider an arbitrary solution template given by s_it_i paths P_i for each $1 \le i \le k$. Let ℓ_e be the number of demand pairs which use edge e on their path. Then for any edge e, the cost of this edge in the buy-at-bulk instance is $c_e f(\ell_e)$. In the robust instance, the required capacity u_e is

$$\begin{aligned} u_e &:= \max_{D \in \mathcal{P}} \sum_{i : e \in P_i} D_{s_i t_i} \\ &= \max_{\pi \in \Pi} \sum_{i=1}^{k} \mathbf{1}_{e \in P_i} (f(\pi(i)) - f(\pi(i) - 1)), \end{aligned}$$

Since the maximum occurs at a vertex of the polytope. But since f is concave, the differences $f(j) - f(j-1)$ decrease as j increases. So we have that

$$\sum_{i=1}^{k} \mathbf{1}_{e \in P_i} (f(\pi(i)) - f(\pi(i) - 1)) \le \sum_{i=1}^{\ell_e} f(i) - f(i-1) = f(\ell_e).$$

In fact we have equality, from any permutation π that maps $\{i : e \in P_i\}$ to $\{1, 2, \ldots, \ell_e\}$. Thus the amount paid for the reservation of edge e is $c_e u_e = c_e f(\ell_e)$, exactly the cost in the buy-at-bulk instance. $\square$

It remains to show that this choice of $\mathcal{P}$ can be separated:

Claim 1. *The polytope $\mathcal{P}$ defined in* (2) *is separable in polynomial time.*

Proof. Let D be an arbitrary demand matrix. If $D_{uv} \neq 0$ for some pair u, v that is not a source-sink pair s_i, t_i, then D is clearly not in $\mathcal{P}$. Otherwise, let the vector $\mathbf{d}$ be defined by $\mathbf{d}_i = D_{s_i t_i}$; this contains all the information in D. Similarly, define $\mathbf{d}^\pi = D^\pi_{s_i t_i}$. The matrix D is in $\mathcal{P}$ if and only if $D = \sum_{\pi \in \Pi} w_\pi D^\pi$, for some nonnegative weights w_π that sum to 1. Equivalently we require

$$\mathbf{d} = \sum_{\pi \in \Pi} w_\pi \mathbf{d}^\pi.$$

But $\mathbf{d}^\pi = P^\pi \mathbf{d}^1$, where P^π is the permutation matrix associated with π, and $\mathbf{d}^1$ is the demand vector associated with the identity permutation. Hence $D \in \mathcal{P}$ if and only if $\mathbf{d}$ is a convex combination of elements in $\{P^\pi \mathbf{d}^1 : \pi \in \Pi\}$. This is equivalent to saying that

$$\mathbf{d} = M \mathbf{d}^1$$

for some doubly stochastic matrix M. This can easily be written as a linear program as follows. Consider a complete bipartite graph with bipartition $X \cup Y$ where nodes in X and Y correspond to the terminal pairs s_it_i. Each $i \in X$ has an associated value $\mathbf{d}_i$, and on Y this node has a value $\mathbf{d}^1_i$. Finding M is equivalent to finding a fractional perfect matching $(x_{ij} : i \in X, j \in Y)$ in this graph such that for each i, $\sum_j x_{ij} \mathbf{d}^1_j = \mathbf{d}_i$. So $\mathcal{P}$ is indeed separable. $\square$

3 Tree demands

3.1 The demand model

As the general robust network design problem is hard to approximate to within logarithmic factors, it is interesting to examine the universes of demand matrices for which which constant approximations are possible. The most prominent examples to date consisted of the universe of hose matrices (exactly solvable), and more generally the class of asymmetric hose matrices (5.55-approximable) described in the introduction [13]. In this section, we describe an 8-approximation for the class of tree-demand matrices. This class includes the hose matrices but is noncomparable to the class of asymmetric hose matrices. Throughout we use OPT denote the cost of an optimal solution for an instance of this problem.

Let T be a tree whose leaves are indexed by the terminals W. We sometimes abuse notation and do not distinguish between the leaves of T and the terminals W. Each edge e of T has an associated capacity b_e.

Definition 1. *A demand matrix D_{ij} whose rows and columns are indexed by W is called a T-*demand *if it can be routed on T without violating the capacities on the edges of T. The set of T-demands defines a polytope that we denote by $\mathcal{H}_T$.*

The *tree demand problem* (for a given T) is defined as the robust network design problem induced by G and the universe $\mathcal{H}_T$. Thus, we seek an oblivious routing for the terminals which minimizes the total capacity cost required to support all T-demands. Notice that the special case where T is a star is precisely a VPN instance, with the marginal on terminal v given by the capacity of the edge in the star between v and the root.

In this section, we provide an approximation algorithm for the tree demand problem. It achieves an approximation ratio of 2 if the edges of T have unit capacity, and a ratio of 8 for arbitrary capacities. We do not show that these bounds are tight however; it is possible that the algorithm described is in fact optimal. This is the case if the tree is a star by the VPN Theorem [8].

An integral part of the algorithm is a facility-location type subroutine which places hubs in the network so that subsets of terminals have low-cost routings to their local hubs. We describe this problem next.

3.2 A hierarchical hubbing algorithm

In this section, we describe an exact algorithm for the following *hierarchical hubbing problem* which is very similar to the *zero-extension problem* [16, 3] on a tree. Given our capacitated tree T, find a mapping $h : V(T) \to V(G)$ such that $h(v) = v$ for each leaf v and subject to this, we wish to minimize $\sum_{e=uv \in T} b_e d_G(h(u), h(v))$. A solution to this problem also defines an oblivious *hierarchical routing template* where P_{uv} (possibly nonsimple) is defined as the union of the shortest paths between nodes $h(z), h(w)$ over all edges $e = zw$ on the unique u-v path in T.

Recall that in the zero-extension problem we are given a set of terminals W within a weighted graph (T in our case) and a metric on W. We wish to assign all nodes of T to terminals so as to minimize the sum over all edges of T, the product of the edges weight, and the distance between the terminals to which its endpoints are assigned. If $W = V(G)$, then the hubbing problem is just zero-extension on the tree T using the metric from G.

The hubbing problem is also a natural extension of the algorithm for the VPN problem. In the case where T is a star the mapping yields a tree in G. In fact, it yields the cheapest *shortest path tree*, where each terminal routes to a hub node r along a shortest path.

Given a mapping for the hubbing problem, we obtain a natural oblivious routing template. For any pair $i, j \in W$, look at the path in T between the leaf nodes i and j. This path $i =$

$x_1, x_2, \ldots, x_t = j$ can now be mapped into a (not necessarily simple) path between i and j in G, by mapping the edges $x_1x_2, x_2x_3, \ldots, x_{t-1}x_t$ via h to paths from i to j via $x_2, x_3, \ldots, x_{t-1}$. This motivates the name "hierarchical hubbing".

Lemma 3.1. *An optimal hierarchical hubbing solution can be found in polynomial time.*

Proof. It is clear that the solution should map an edge $uv \in T$ to a shortest path between $h(u)$ and $h(v)$. So the optimal hierarchical hub routing is determined by the map h on the vertices of T, i.e., by the positions of the hierarchical hubs. For any subtree S of T, and any node $v \in V$, let $C(S, v)$ be the cost of an optimal hierarchical hubbing solution for S, but with the root of S mapped to node v. For $S = \{i\}$ a leaf of T, define $C(\{i\}, i) = 0$ and $C(\{i\}, v) = \infty$ if $v \neq i$, i.e., mapping i to v is not valid.

We calculate these costs using dynamic programming. Let s be a node of T, and S the subtree rooted at s. Label the children of s as $s_1, s_2, \ldots, s_k$, and let S_i be the subtree rooted at s_i. Let e_i denote the edge from s to s_i. Suppose we know $C(S_i, w)$ for $1 \leq i \leq k$ and all nodes $w \in V$. We wish to calculate $C(S, v)$ for some $v \in V$. But the optimal location of the hub represented by s_i is clearly the vertex w_i that minimizes $C(S_i, w_i) + b_{e_i} d(v, w_i)$. Then $C(S, v) = \sum_{i=1}^{k} C(S_i, w_i) + b_{e_i} d(v, w_i)$. This clearly yields a polynomial algorithm via dynamic programming. $\square$

We mention two extensions of the problem, *hub-constrained* and *leaf-constrained* hierarchical hubbing, which relate to the optical design problem discussed in the introduction. In the hub-constrained version, each edge uv of T has an associated bound $U(u, v)$ which gives the maximum allowed distance between $h(u)$ and $h(v)$. In the leaf-constrained version, we think of T as being rooted at some node r, and for each leaf u and node v on the path from u to r, we require that $d(h(u), h(v)) \leq U(u, v)$. The algorithm described above can be easily modified to find the optimal solution subject to these extra constraints.

3.3 Overview of the analysis

In the next section, we define a class of demand matrices D^ℓ with the following properties. Each D^ℓ is associated with a so-called *connected labelling* of T, which in turn has an associated oblivious template $\mathcal{P}^\ell$. Moreover, if G has enough capacity to route a given D^ℓ, then it also has enough capacity to support all demands in $\mathcal{H}_T$ via the template $\mathcal{P}^\ell$. Unfortunately, it is not the case that every $D^\ell \in \mathcal{H}_T$. Instead, we define a distribution over this class of matrices such that $\bar{D} := \mathbb{E}(D^\ell)$ lies in $\alpha\mathcal{H}_T$ for some constant α. We show that it is actually enough to route $\bar{D}$ in G. It follows that for some ℓ, the cost of routing D^ℓ is within a constant factor of the optimal robust network. Finding such a D^ℓ may be hard in general; instead, we show that the cost of routing any D^ℓ is at least the cost of an optimal hierarchical hub routing, which we can find in polynomial time. Since the hierarchical hub routing is a feasible solution to the tree demand problem, this gives an α approximation; we will demonstrate a distribution that yields $\alpha = 8$.

3.4 Connected labellings and hub routings

Definition 2. *A* connected labelling *of a tree T is a function $\ell : V(T) \to W$ satisfying the properties that $\ell(w) = w$ for all $w \in W$, and $\ell^{-1}(w)$ connected for all $w \in W$.*

A connected labelling ℓ induces a demand matrix D^ℓ in a very natural way. Simply contract each set $\ell^{-1}(w)$ to obtain a new tree T^ℓ, with $V(T^\ell) = W$ (see Figure 1). The edges of T^ℓ determine the nonzero demands—if $uv \notin T^\ell$, then $D^\ell_{uv} = 0$. Now consider $uv \in T^\ell$. There is a unique edge $e \in T$ that connects the components $\ell^{-1}(u)$ and $\ell^{-1}(v)$. Define $D^\ell_{uv} = b_e$.

The optimal solution to route just the single demand matrix D^ℓ simply consists of routing on shortest paths. This has a cost of $C^*(D^\ell) = \sum_{u,v \in W} D^\ell_{uv} d(u,v)$. This has an alternative interpretation that connects to hierarchical hubbing. Recall that the hierarchical hubbing algorithm found a mapping $h : V(T) \to V(G)$, taking leaves to respective terminals, and minimizing the cost $\sum_{uv \in E(T)} b_{uv} d(h(u), h(v))$. This means that the optimal solution for the single matrix D^ℓ is exactly a hierarchical hubbing solution where we enforce $h(u) = \ell(u)$ for each node $u \in V(T)$. It follows that:

Lemma 3.2. *For any connected labelling ℓ, the hierarchical hubbing solution for T costs no more than the optimal routing for D^ℓ.*

Let $\mathcal{Q}$ be any routing of D^ℓ (although we could assume a shortest path routing) and let $\mathbf{u}$ be the edge capacity (i.e. induced edge load) vector associated with this static routing. We define a routing template as follows. For any given pair u, v of terminals, consider the path between u and v in T^ℓ; let it be $v_0 v_1 \cdots v_m$, where $v_0 = u$ and $v_m = v$. Then for each edge $v_i v_{i+1}$ of this path, there is an associated route $Q_{v_i v_{i+1}}$ in $\mathcal{Q}$. We define P_{uv} to be a simple u-v path contained in the union $Q_{vv_1} \cup Q_{v_1 v_2} \cup \cdots \cup Q_{v_{m-1} v}$, and take $\mathcal{P}^\ell$ to be the routing template given by the P_{uv}'s.

Lemma 3.3. *The capacities $\mathbf{u}$ are enough to support the routing of any $D \in \mathcal{H}_T$ via $\mathcal{P}^\ell$.*

Proof. Let D be any T-demand, and let f be any edge of G. Let E' be the set of edges $e = zw$ in T^ℓ, such that Q_e contains f. Note that since T^ℓ was obtained from T by contracting edges, we can think of e as an edge in T also. A pair u, v uses path Q_e as part of their routing P_{uv} if e separates v and w in T. Let $S(e)$ denote the set of such terminal pairs. Then the total load induced on edge f by demand D via $\mathcal{P}^\ell$ is at most $\sum_{e \in E'} \sum_{uv \in S(e)} D_{uv} \leq \sum_{e \in E'} b_e$. The last inequality follows by definition of a tree demand: the total demand from D across any edge $e \in T$ cannot exceed b_e. Since $D^\ell_{zw} = b_{zw}$ for each edge in $zw \in T^\ell$, the total load does not exceed $\sum_{zw \in E'} D^\ell_{zw} \leq u_f$ as required. □

3.5 Distributions over connected labellings

For any connected labelling ℓ, D^ℓ induces a load on edges in the original T. For edge $e = uv \in T$, this is $\sum_{uv \in S(e)} D^\ell_{uv}$, where recall $S(e)$ is the set of terminal pairs separated by e in T. If $e \in T^\ell$, the only pair in $S(e)$ with nonzero demand in D^ℓ is between $\ell^{-1}(u)$ and $\ell^{-1}(v)$, and this gives a load of b_e. For other edges, the load may generally exceed the edge's capacity b_e, and so D^ℓ may not be a valid T-demand. But suppose we manage to find a distribution so that the *expected load* across on any edge of T exceeds its capacity only by a constant factor α. Then consider the demand matrix $\bar{D}$ obtained by averaging the demand matrices D^ℓ over this distribution, i.e. the demand matrix given by $\bar{D}_{ij} = \mathbb{E}(D^\ell_{ij})$. The demand $\bar{D}/\alpha$ does not exceed any edge capacity, and so is a feasible T-demand. Thus the cost to optimally route the single matrix $\bar{D}/\alpha$ (which we denote by $C^*(\bar{D}/\alpha)$) is a lower bound on the cost of $\textsc{opt}$, i.e. $C^*(\bar{D}) \leq \alpha \cdot \textsc{opt}$. Since static routings are on shortest paths, we have a simple formula for $C^*(\bar{D})$:

Claim 2. $C^*(\bar{D}) = \mathbb{E}(C^*(D^\ell))$.

Proof. We know that the optimal solution to route the fixed demand matrix D consists of adding together shortest paths between each pair, weighted by the appropriate entry of the demand matrix.

$$C^*(\bar{D}) = \sum_{u,v \in W} \bar{D}_{uv} d(u,v). \tag{3}$$

The same is true for any of the D^ℓ's:

$$C^*(D^\ell) = \sum_{u,v \in W} D^\ell_{uv} d(u,v).$$

Taking expectations of both sides, and then using (3), we have

$$\mathbb{E}(C^*(D^\ell)) = \sum_{u,v \in W} E(D^\ell_{uv}) d(u,v) = \sum_{u,v \in W} \bar{D}_{uv} d(u,v) = C^*(\bar{D}).$$

$\square$

It follows from this claim that there must be some ℓ s.t. $C^*(D^\ell) \leq C^*(D)$. By Lemma 3.2, the cost of a solution to the hierarchical hubbing algorithm is at most the cost of routing any fixed D^ℓ. Since any hierarchical hubbing solution yields an oblivious template whose cost to support demands in $\mathcal{H}_T$ is the same as the hierarchical hubbing cost, we would thus obtain a factor α approximation for the tree demand problem.

3.6 Expected loads for a distribution

We will now define a distribution over connected labellings of T with the desired properties. We must first consider the loads induced by a fixed D^ℓ.

Consider an arbitrary edge $e = uv \in E(T)$. Let L_e and R_e be the leaf sets of the two components of $T \setminus \{e\}$, with u in the same component as L_e and v in the same component as R_e. It is useful for us to give an orientation to the edges. Orient e from u to v, and orient all other edges to be consistent with this. In other words, for each edge f in the component L_e, orient f to point towards e, and for f in R_e, orient away from e. Call the arcs in this orientation $\mathbf{A}_e(T)$.

First, we need to calculate the load for a fixed connected labelling ℓ. Consider the contracted tree T^ℓ defined earlier, which in turn defines D^ℓ. Edges in T^ℓ correspond to nonzero demands between the terminals of the labels of the endpoints. Every edge f in T^ℓ which has one endpoint x labelled with a terminal in L_e and the other endpoint y labelled by a terminal in R_e, contributes to the load of e. These are the only demands in D^ℓ that do. The contribution of f is exactly the capacity of the unique edge between the components $\ell^{-1}(x)$ and $\ell^{-1}(y)$ in T.

So the total contribution is

$$\begin{aligned} &\sum_{f \in E(T)} b_f \cdot \mathbf{1}_{(\text{one endpoint of } f \text{ has label in } L_e \text{, the other in } R_e)} \\ &= \sum_{(x,y) \in \mathbf{A_e}(T)} b_{xy} \cdot \mathbf{1}_{\ell(x) \in L_e \wedge \ell(y) \in R_e}. \end{aligned} \tag{4}$$

Now consider any distribution over the labellings. We're interested in the average, i.e. expected, load on edges of T. By linearity of expectations, this is

$$\begin{aligned} &\sum_{(x,y) \in \mathbf{A_e}(T)} b_{xy} \mathbf{P}(\ell(x) \in L_e \wedge \ell(y) \in R_e) \\ &= \sum_{(x,y) \in \mathbf{A_e}(T)} b_{xy} (\mathbf{P}(\ell(y) \in R_e) - \mathbf{P}(\ell(x) \in R_e)). \end{aligned} \tag{5}$$

This follows since there are only three possible events for the pair x, y: (i) $\ell(x), \ell(y) \in L_e$ (ii) $\ell(x) \in L_e, \ell(y) \in R_e$ or (iii) $\ell(x), \ell(y) \in R_e$ (see Figure 2).

We now describe a particular distribution of connected labellings. We show that in the case where $b_e = 1$ for all $e \in E(T)$, this produces an expected load of 2, and hence the hierarchical hubbing algorithm is a 2-approximation. For general capacities, this distribution does not yield a constant expected load; however, it is the starting point for constructing a distribution that does.

Define the random labelling ℓ using a coupled random walk scheme as follows. First, pick an arbitrary non-leaf node of T to be the root; call it r. For every non-leaf node s, pick one of its children uniformly at random and draw an arrow to it from s. Now for any node s of T, define $\ell(s)$ to be the terminal reached by following the arrows from s. This clearly gives a (random) connected labelling.

Fix an edge $e \in E(T)$. We must compute the expected load on e, as given in Equation (5). Let us choose to orient e away from the root, so that R_e is the component of $T \setminus \{e\}$ below e, i.e. not containing the root. It is clear that any edges below e do not contribute to the sum, since walks from x and y definitely end up in R_e (the walks can't go up the tree). Likewise, any edge that is not on, or touching, the path from e to the root cannot contribute—x and y would both have to end up in L_e.

Now label the nodes on the path from e to the root by $x_0 = y, x_1 = x, \ldots, x_t = r$. Let B_i be the sum of the capacities of the downward edges from x_i, and write $b_i := b_{x_i x_{i-1}}$ (see again Figure 2).

There are two types of edges to consider:

- An edge of the form $x_i x_{i-1}$ contributes

$$\begin{aligned}&b_i(\mathbf{P}(\ell(x_{i-1}) \in R_e) - \mathbf{P}(\ell(x_i) \in R_e))\\ &= b_i(B_i/b_i \cdot \mathbf{P}(\ell(x_i) \in R_e) - \mathbf{P}(\ell(x_i) \in R_e))\\ &= (B_i - b_i)\mathbf{P}(\ell(x_i) \in R_e)\end{aligned}$$

- An edge of the form $g = zx_i$, where z is a child of x_i, not equal to x_{i-1}. g contributes

$$b_g(\mathbf{P}(\ell(x_i) \in R_e) - \mathbf{P}(\ell(z) \in R_e)) = b_g \mathbf{P}(\ell(x_i) \in R_e)$$

  since $\ell(z) \in L_e$. If we sum the contributions of all the edges (other than $x_i x_{i-1}$) hanging from x_i, we thus obtain

$$(B_i - b_i)\mathbf{P}(\ell(x_i) \in R_e).$$

Summing the contributions of all these edges, we find that the expected load on edge e is exactly

$$\sum_{i=1}^{t} 2(B_i - b_i)\mathbf{P}(\ell(x_i) \in R_e) = 2\sum_{i=1}^{t}(B_i - b_i)\prod_{j=1}^{i}\frac{b_j}{B_j} \tag{6}$$

3.7 Trees with unit capacities

If $b_e = 1$ for all $e \in E(T)$, then we have from Eq. (6) that the expected load on any edge is at most

$$\begin{aligned}&2\sum_{i=1}^{t}(B_i - 1)\prod_{j=1}^{i} 1/B_j\\ &= 2\sum_{i=1}^{t}\prod_{j=1}^{i-1} 1/B_j - 2\sum_{i=1}^{t}\prod_{j=1}^{i} 1/B_j\\ &= 2 - 2\prod_{j=1}^{t} 1/B_j \;\;\le\;\; 2.\end{aligned}$$

So $\bar{D}/2 \in \mathcal{H}_T$, as claimed.

3.8 Trees with arbitrary capacities

The same distribution does not work for arbitrary capacities. A complete binary tree of height h, with all edges at height i having capacity $2^i + 1$, can easily be shown to have an expected load of $O(\log h)$ on a leaf edge.

Instead we proceed as follows. Consider any edge $e = xy$ in T with x higher in T (with respect to the root) than y. If

$$b_e \geq \sum_{e' \in \delta_T(y) \setminus \{e\}} b_{e'}, \tag{7}$$

then $\mathcal{H}_T$ is not changed even if we work with the tree T' obtained by contracting e. Thus we may assume that no such edges exist at the outset. We look at an approximate form of this inequality to eliminate problematic edges in T. Call an edge $e \in T$ as above such that $b_e \geq \frac{1}{2} \sum_{e' \in \delta_T(y) \setminus \{e\}} b_{e'}$ *wide*. Find a lowest level wide edge and contract it. Note that since (7) does not occur for any such edge, we have that this contraction will not create any new wide edges. Repeat this process until we have a new tree $\hat{T}$, with associated demand polytope $\mathcal{H}_{\hat{T}}$. Since we only contracted wide edges of T, one easily checks that for any $D \in \mathcal{H}_{\hat{T}}$, $D/2 \in \mathcal{H}_T$. Thus the optimal solution to route all $\hat{T}$-demands costs at most twice the optimal solution routing all T-demands.

We now return to the analysis for the expected load with the additional assumption that there are no wide edges. In this case, we have $b_i \leq B_{i-1}/2$ for all $i \geq 2$ and so

$$\prod_{j=1}^{i} \frac{b_j}{B_j} \leq \frac{b_1}{B_1} \frac{B_1/2}{B_2} \frac{B_2/2}{B_3} \cdots \frac{B_{i-1}/2}{B_i} = \frac{b_1}{2^{i-1} B_i}$$

Thus the total expected load on edge e is

$$2 \sum_{i=1}^{t} (B_i - b_i) 2^{-(i-1)} \frac{b_1}{B_i} \leq 4b_1 = 4b_e$$

and so the congestion of e is at most a factor of 4. Thus we achieve a factor of 4 with respect to the optimal routing for $\mathcal{H}_{\hat{T}}$, giving a factor 8 approximation to the T-demand problem.

It may be possible to improve the approximation factor by a better choice of distribution. However, it will not be possible to improve the factor to the 2 obtained in the uncapacitated case with this method, since one can construct examples where for every connected labelling ℓ, the cost of routing D^ℓ is essentially 4 times the cost of the optimum.

4 Conclusion

We have described an algorithm which guarantees a robust network design for the class of tree demands which is within a factor 8 of the optimal (in fact within a factor 8 of the optimal dynamic solution). But it may even be the case that the algorithm always gives an *optimal* solution—we are not aware of any counterexamples. By the VPN result [8], this is true in the case that T is a star.

There are many directions to explore in terms of even more general demand constraints. For example, one might consider arbitrary (not nested) cut constraints on the demand graph. Another possibility is to consider the polytope of demands that are (fractionally) routable on some given capacitated graph H; we are then asking to design a network on G's topology that can support any traffic pattern which is routable in H.

Acknowledgements: We are grateful to Navin Goyal for many helpful discussions, and to Moses Charikar who suggested looking at generalizations of the VPN problem involving cut families. We also thank Chandra Chekuri for very helpful suggestions.

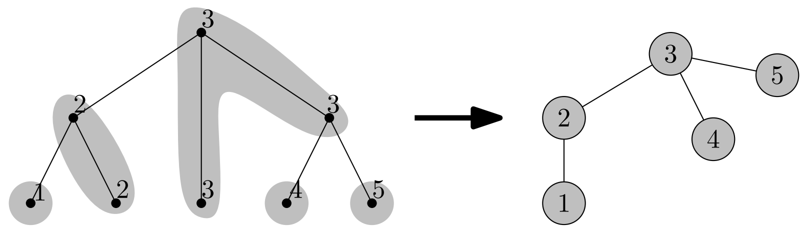

Figure 1: A connected labelling, and the associated T^ℓ obtained by contracting.

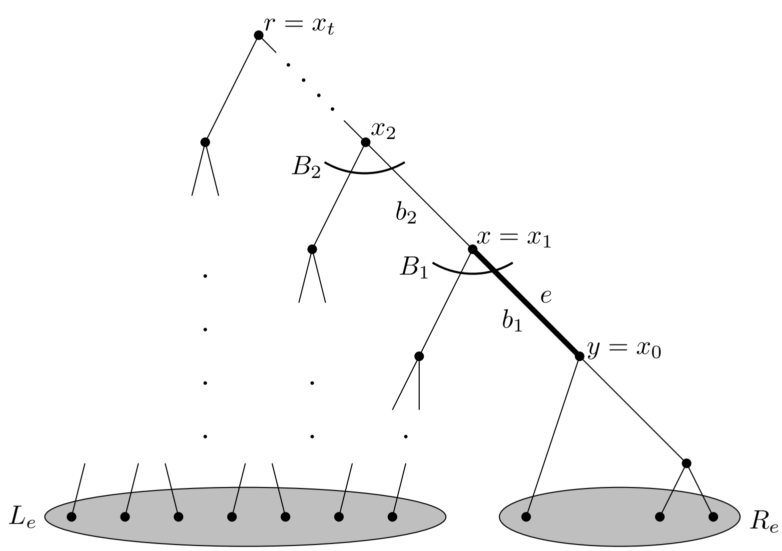

Figure 2: Calculating the expected load on edge e.

References

[1] M. Andrews. Hardness of buy-at-bulk network design. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 115–124, Washington, DC, USA, 2004. IEEE Computer Society.

[2] W. Ben-Ameur and H. Kerivin. New economical virtual private networks. *Communications of the ACM*, 46(6):69–73, 2003.

[3] G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. In *SODA '01: Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms*, pages 8–16, Philadelphia, PA, USA, 2001. Society for Industrial and Applied Mathematics.

[4] C. Chekuri. Routing and network design with robustness to changing or uncertain traffic demands. *SIGACT News*, 38(3):106–128, 2007.

[5] F. Eisenbrand, F. Grandoni, G. Oriolo, and M. Skutella. New approaches for virtual private network design. *SIAM Journal on Computing*, 37(3):706–721, 2007.

[6] T. Erlebach and M. Ruegg. Optimal bandwidth reservation in hose-model vpns with multi-path routing. In *Proc. of INFOCOM*, pages 2275–2282, 2004.

[7] A. J. Fingerhut, S. Suri, and J. S. Turner. Designing least-cost nonblocking broadband networks. *Journal of Algorithms*, 24(2):287–309, 1997.

[8] N. Goyal, N. Olver, and B. Shepherd. The VPN conjecture is true. In *STOC*, 2008.

[9] N. Goyal, N. Olver, and B. Shepherd. Dynamic vs oblivious routing in network design. *Manuscript*, 2009.

[10] F. Grandoni, V. Kaibel, G. Oriolo, and M. Skutella. A short proof of the VPN tree routing conjecture on ring networks. Accepted to *Operations Research Letters*, 2007.

[11] A. Gupta. Private communication.

[12] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flow. In *Proc. of ACM STOC*, pages 389–398, 2001.

[13] A. Gupta, A. Kumar, M. Pál, and T. Roughgarden. Approximation via cost sharing: Simpler and better approximation algorithms for network design. *J. ACM*, 54(3):11, 2007.

[14] C. A. J. Hurkens, J. C. M. Keijsper, and L. Stougie. Virtual private network design: A proof of the tree routing conjecture on ring networks. In *Proc. of IPCO*, pages 407–421, 2005.

[15] G. Italiano, S. Leonardi, and G. Oriolo. Design of trees in the hose model: the balanced case. *Operations Research Letters*, 34(6):601–606, 2006. A preliminary version appeared in *Proc. of ARACNE*, 65-76, 2002.

[16] A. Karzanov. Metrics with finite sets of primitive extensions. *Annals of Combinatorics*, 2(3):211–241, 1998.

[17] H. Räcke. Minimizing congestion in general networks. *Proceedings of the 43rd Symposium on Foundations of Computer Science*, pages 43–52, 2002.

[18] F. B. Shepherd and P. J. Winzer. Selective randomized load balancing and mesh networks with changing demands. *Journal of Optical Networking*, 5(5):320–339, 2006.

[19] L. Valiant. A scheme for fast parallel communication. *SIAM J. Comput.*, 11(2):350–361, 1982.