Mathematics of Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Approximability of Robust Network Design

Neil Olver, F. Bruce Shepherd

MATHEMATICS OF OPERATIONS RESEARCH
Vol. 39, No. 2, May 2014, pp. 561–572
ISSN 0364-765X (print) | ISSN 1526-5471 (online)
http://dx.doi.org/10.1287/moor.2013.0620

Approximability of Robust Network Design

Neil Olver
Department of Mathematics, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, olver@math.mit.edu

F. Bruce Shepherd
Department of Mathematics and Statistics, McGill University, Montreal, Quebec H3A 0G4, Canada, bruce.shepherd@mcgill.ca

We consider robust (undirected) network design (RND) problems where the set of feasible demands may be given by an arbitrary convex body. This model, introduced by Ben-Ameur and Kerivin [Ben-Ameur W, Kerivin H (2003) New economical virtual private networks. *Comm. ACM* 46(6):69–73], generalizes the well-studied virtual private network (VPN) problem. Most research in this area has focused on 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 Chekuri [Chekuri C (2007) Routing and network design with robustness to changing or uncertain traffic demands. *SIGACT News* 38(3):106–128], 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 polylogarithmic factors. We establish this by showing a general reduction of buy-at-bulk network design to the robust network design problem. Gupta pointed out that metric embeddings imply an $O(\log n)$-approximation for the general RND problem, and hence this is tight up to polylogarithmic factors.

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 of 8 in general, and 2 for the case where the tree has unit capacities. As an application of this result, we consider so-called H-tope demand polytopes. These correspond to demands which are routable in some graph H. We show that the corresponding RND problem has an $O(1)$-approximation if H admits a stochastic constant-distortion embedding into tree metrics.

Keywords: robust network design
MSC2000 subject classification: Primary: 90B18; secondary: 68M10, 90C35, 90C27
OR/MS subject classification: Primary: networks/graphs: theory
History: Received September 7, 2012; revised May 5, 2013. Published online in *Articles in Advance* September 6, 2013.

1. Introduction. Robust network design considers the problem of designing a network, by buying bandwidth on some underlying undirected graph $G = (V, E)$, to support some *set* of *valid* demands. By a "demand," we mean simply a prescription of the traffic requirements of the network at some moment; between any pair of nodes, the needed bandwidth is specified. So we may represent a demand by a matrix D_{ij}, indexed by nodes, and the set or *universe* of valid demands as a set of such matrices; the universe is assumed to be given. Subject to the constraint that all demands in the universe be routable, our goal is to minimize the cost of the network; we may buy arbitrary amounts of bandwidth on any network edge, and pay for each unit of bandwidth. We will use the term *robust network design* (RND) to refer to this optimization problem.

This model is motivated by situations where the traffic across the network is variable, or not known precisely. The primary application is to communication networks; the goal is to create a "virtual private network" (VPN) on top of some underlying network. To guarantee service requirements, we must reserve capacity on this network for our exclusive use. The model, in the generality described above, was introduced by Ben-Ameur and Kerivin [3]; for a certain important special class of valid demands, it was introduced earlier by Fingerhut et al. [11] and Duffield et al. [7].

Since our network must potentially satisfy multiple demand matrices, we must also define how routings are allowed to change as demands vary. 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 generally not practical. 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 *solution template*. Subsequently, if we must support a matrix D, then for each i, j, we send D_{ij} units of flow along P_{ij}. For such a template, the capacity required on edge e is the maximum load put on e by one of the valid matrices in the universe. Hence the optimization problem is to find a template for which the appropriately weighted sum of these maximum loads is minimized. (There is also a middle way; in *multipath* oblivious routing, the template consists of unit flows f_{ij} for each pair $i, j \in V$; this flow specifies in what proportions demand from i to j is fractionally routed. This version of robust network design is polynomially solvable as long as we can optimize over the universe (see Ben-Ameur and Kerivin [3]), and will not concern us in this paper.)

We mention that design criteria besides total cost are also possible. In particular, 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., Valiant [27]) and reaching to more recent breakthroughs of Räcke [24].

Research on robust network design has focused on special classes of demand universes, as will be discussed further below. As pointed out in Chekuri [6], however, the approximability status for robust design with general polytopes had not been resolved. Our first result shows that it is hard to approximate the optimal solution within polylogarithmic factors (under appropriate complexity assumptions). We show this by demonstrating that the buy-at-bulk network design problem can be obtained as a special case of robust network design. Previously, RND was only known to be APX-hard (since the Steiner tree problem is easily seen to be a special case). Our hardness result is tight up to polylog factors, since an $O(\log n)$ factor approximation is known based on metric tree embeddings (Gupta [16]; cf. Chekuri [6], Goyal et al. [13]).

Our negative result gives further motivation for considering special classes of demand universes. So far, relatively few universes have been considered. The most studied are the *hose-demands*, introduced by Fingerhut et al. [11] and Duffield et al. [7]. These come in both a symmetric and asymmetric flavour; we withhold discussion of the asymmetric version until later. In the symmetric hose model, the demand universe is given as

$$\mathcal{H} = \left\{ D_{ij} \,\middle|\, \sum_{j \in V} D_{ij} \le b(i) \;\forall\, i \in V \right\}.$$

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: $D_{ij} = D_{ji}$ represents a single demand between i and j. The *virtual private network* (VPN) problem is the minimum cost robust network design problem for the class $\mathcal{H}$. One primary motivation is the situation where each terminal connects into the main network via a link (the "hose") of limited capacity; the goal is to be able to route any demand that can fit through the hoses.

The VPN problem was shown to have a 2-approximation algorithm independently in Fingerhut et al. [11], Gupta et al. [17]. 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 (Italiano et al. [21]) that the optimal solution was in fact such a tree (making the VPN problem polytime solvable). This was resolved positively in Goyal et al. [12], building on work in Grandoni et al. [14] (the conjecture had earlier been proved for ring networks and some other special cases (Hurkens et al. [20])).

In the second part of this paper, we propose and study a generalization of the symmetric hose model. In addition to a network $G = (V, E)$, we are given a capacitated tree T with leaf set $W \subseteq V$; there is no other restriction on T. The tree T naturally defines a demand universe $\mathcal{U}_T$ (the T-*tope*) consisting of all demands that are routable on T. That is, simultaneously routing D_{ij} units between i and j in T, for each $i < j$, should not overload the capacity, denoted by $b(e)$, of any edge $e \in E(T)$. We call the new class of demand matrices *tree-demands*. Note that if we take T to be a star with center node v, then the set of tree-demands is just the class of hose matrices, with the marginal of terminal i given by $b(iv)$.

We observe another interpretation of T-topes 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; see Figure 1.

The extra generality of this class of universes can be used to more accurately define the requirements of a network, such as a VPN, potentially yielding a much cheaper final network. For instance, if there is a natural grouping of the terminals, such that the communication requirement within each group is larger than the requirement between groups, then a tree demand universe can be chosen that captures this information.

In §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 $\mathcal{U}_T$; this is improved to a factor of 2 for the unit capacity case. The algorithm generalizes the algorithm for the hose model described above. It essentially finds the best possible way to "embed" T into the graph, where each internal node of T is mapped somewhere in the graph, and edges of the tree are mapped to shortest paths between the endpoints.

Our proofs actually imply something stronger than just an approximation guarantee for the oblivious RND problem. They show that the proposed algorithm returns a solution which costs at most a factor of 8 larger than the optimal solution with *dynamic* fractional routing. This is again improved to 2 in the case of unit capacities, which is best possible; even in the hose case, the gap between oblivious and dynamic routing can be arbitrarily

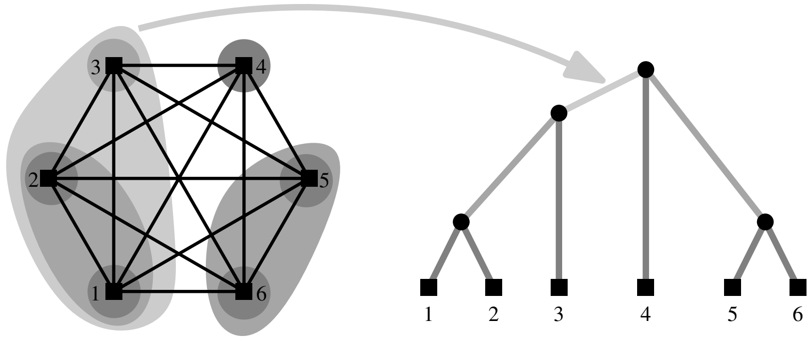

Figure 1. An interpretation of tree-demands in terms of cuts in the demand graph.

close to 2. Note that for arbitrary universes, the gap between oblivious and dynamic routing can be as large as $\Theta(\log n)$ (Goyal et al. [13]).

A natural further extension of the class of tree demand polytopes is to consider demands routable on some given graph. Let H be an arbitrary capacitated graph, except that some subset of its vertices are associated with the terminal set W. We can define a polytope (the *H-tope*) by the set of demands that are routable on H. In §4 we show that for certain classes of graphs H, our 8-approximation can be extended to give constant factor approximations for H-topes.

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 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 (it contains the Steiner tree as a special case). A randomized constant-approximation algorithm for the robust design problem for the class of asymmetric hose matrices is given in Gupta et al. [18], and the approximation factor has been improved in a number of works (Eisenbrand and Grandoni [8], Eisenbrand et al. [9], Grandoni and Rothvoß [15]). 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.

Eisenbrand and Happ [10] consider the following generalization of the hose model. The terminal set W is partitioned into some number of groups $W_1, W_2, \ldots, W_k$. Each terminal has a marginal, as in the usual hose model. The demand universe is simply the set of hose demands for which $D_{ij} = 0$ for all i, j in the same group. In particular, the case of two groups corresponds to the asymmetric hose model. They demonstrate a constant factor approximation algorithm.

In Shepherd and Winzer [26], an application of the hose model 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 Shepherd and Winzer [26] 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 §3.2).

1.1. Model and definitions. A general instance of robust network design consists of an undirected graph $G = (V, E)$, where each edge $e \in E$ has an associated nonnegative cost $c(e)$, and a demand universe $\mathcal{U} \subset \mathbb{R}_+^{V \times V}$. The edge cost $c(e)$ represents the per-unit cost of bandwidth reserved on edge e. 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. We will use the notation $\mathbf{1}_A$ to denote the indicator function of a predicate A.

A solution to an RND instance is specified by a routing template $\mathcal{P} = \{P_{ij} \mid i, j \in V\}$, where P_{ij} is an $i-j$-path. Once this routing template has been specified, we can determine the required capacity on any edge e: it is given by the maximum load that can be induced by a valid demand; i.e.,

$$u(e) = \max_{D \in \mathcal{U}} \sum_{i, j \in W : e \in P_{ij}} D_{ij}. \tag{1}$$

The cost of the solution is then simply

$$C(\mathcal{P}) = \sum_{e\in E} c(e)u(e).$$

It follows immediately from (1) that only the convex hull of $\mathcal{U}$ is relevant, and so we will always take $\mathcal{U}$ to be convex; typically it is a polyhedron. In addition, we require that the $\mathcal{U}$ is *separable*; i.e., there is a polytime algorithm to solve its separation problem. If not, even determining the capacity requirement for a given solution is intractable.

In general, demand from i to j and demand from j to i may be considered as distinct. In some situations, it makes sense to consider demand to be undirected, making these two demands indistinguishable. We say then that the RND problem is *symmetric*. This may be embedded into the general RND framework by taking the universe to consist only of lower triangular matrices. However, for notational convenience, when we are in this setting we will take $D_{ij} = D_{ji}$, both referring to the undirected demand between i and j. Similarly, $P_{ij} = P_{ji}$ refers to the same path. Our lower bound construction in §2, as well as the tree demand universes in §3, are both symmetric.

For convenience, we may also specify a subset W of the nodes of G as *terminals*; demand will always be between terminals, and so we may consider that demand matrices are indexed only by W.

2. An equivalence between robust and buy-at-bulk network design. The *uniform buy-at-bulk problem* is defined as follows. 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 and 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 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. For example, choosing $f(x) = \min\{x, 1\}$ recovers the Steiner tree problem.

In this section, we show that the uniform buy-at-bulk network design problem can be simulated by a robust network design problem over an appropriately chosen universe. This allows us to use a seminal result of Andrews [2]:

The uniform buy-at-bulk network design problem cannot be approximated to within a factor of $\Omega(\log^{1/4-\epsilon} n)$ for any $\epsilon > 0$, assuming that NP $\not\subset$ ZPTIME$(n^{\mathrm{polylog}\, n})$.

(Note that the buy-at-bulk problem is defined on an *undirected* graph; strong hardness results for directed graphs are easy to show.)

We begin with an instance of uniform buy-at-bulk. From this, we will construct an instance of robust network design with a polytope that can be described very simply, and separated in polynomial time. Combining this with the above result of Andrews [2], we immediately obtain that RND is hard to approximate to within $\Omega(\log^{1/4-\epsilon} n)$ for any $\epsilon > 0$, under the appropriate complexity assumptions.

Let Π be the set of permutations of the integers 1 through k. For any $\pi \in \Pi$, define the demand matrix D^π by

$$D^\pi_{uv} = \begin{cases} f(\pi_i) - f(\pi_i - 1) & \text{if } \{u, v\} = \{s_i, t_i\} \text{ for some } 1 \le i \le k, \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

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

$$\mathcal{B} := \mathrm{conv}\{D^\pi \colon \pi \in \Pi\}. \tag{3}$$

THEOREM 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 $\mathcal{B}$ 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 $c(e)f(l_e)$. In the robust instance, the required capacity $u(e)$ is

$$\begin{aligned} u(e) &:= \max_{D\in\mathcal{B}} \sum_{i:\, e\in P_i} D_{s_it_i} \\ &= \max_{\pi\in\Pi} \sum_{i=1}^{k} \mathbf{1}_{e\in P_i}(f(\pi_i) - f(\pi_i - 1)), \end{aligned}$$

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

$$\begin{aligned}\sum_{i=1}^{k} \mathbf{1}_{e\in P_i}(f(\pi_i)-f(\pi_i-1)) &\le \sum_{i=1}^{l_e} f(i)-f(i-1)\\ &= f(l_e).\end{aligned}$$

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

It remains to show that this choice of demand universe can be separated. This follows immediately from the following claim.

Claim 1. *The polytope $\mathscr{B}$ defined in (3) has a compact extended formulation.*

Proof. Let $\mathscr{I}=\{\{s_i,t_i\}: 1\le i\le k\}$. Any $D\in\mathscr{B}$ must satisfy $D_{uv}=0$ for any pair $\{u,v\}\notin\mathscr{I}$. This gives us the first set of linear constraints; from now on, we consider only demand matrices satisfying these constraints. We index the remaining entries of D with a vector $\mathbf{d}$, defined as $d_i=D_{s_it_i}$ for all i. Also define $\boldsymbol{\delta}^\pi$ by $\delta_i^\pi=D^\pi_{s_it_i}$; note that these are fixed vectors.

A matrix D is in $\mathscr{B}$ if and only if $D=\sum_{\pi\in\Pi} w_\pi D^\pi$, for some nonnegative weights w_π that sum to 1, or equivalently,

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

But $\boldsymbol{\delta}^\pi=P^\pi\boldsymbol{\delta}^1$, where P^π is the permutation matrix associated with π, and $\boldsymbol{\delta}^1$ is the demand vector associated with the identity permutation. Hence, $D\in\mathscr{B}$ if and only if $\mathbf{d}$ is a convex combination of vectors in $\{P^\pi\boldsymbol{\delta}^1: \pi\in\Pi\}$; equivalently, there is some doubly stochastic matrix[1] M so that

$$\mathbf{d}=M\boldsymbol{\delta}^1.$$

This is clearly a linear system in the unknowns, M and $\mathbf{d}$.

3. Tree-demands.

3.1. The demand model. As the general robust network design problem is hard to approximate to within a polylog factor, it is interesting to examine the universes of demand matrices for 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 (constant factor approximable) described in the introduction (Gupta et al. [18]). In this section, we describe an 8-approximation for the class of tree-demand matrices. This class includes the hose matrices but is incomparable to the class of asymmetric hose matrices. Throughout, we use opt to 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 will sometimes abuse notation and 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 $\mathscr{U}_T$.

The *tree demand problem* (for a given T) is defined as the robust network design problem induced by G and the universe $\mathscr{U}_T$. 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.

The approximation algorithm we describe 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 (Goyal et al. [12]).

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.

[1] A doubly stochastic matrix is a square matrix of nonnegative entries, such that every row and column sum is 1.

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* (Karzanov [22], Calinescu et al. [4]) on a tree. Given our capacitated tree T, find a mapping h: $V(T) \to V(G)$ such that $h(w) = w$ for each leaf w, and subject to this, we wish to minimize $\sum_{e=wz\in E(T)} b(e) d_G(h(w), h(z))$. A solution to this problem also defines an oblivious *hierarchical routing template*, where P_{ij} (possibly nonsimple) is defined as the union of the shortest paths between nodes $h(w)$, $h(z)$ over all edges $e = wz$ on the unique i-j- 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, of the product of the edge weight and the distance between the terminals to which its endpoints are assigned. If $W = V(G)$, then the hubbing problem is just the 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 1. *An optimal hierarchical hubbing solution can be found in polynomial time.*

Proof. It is clear that the solution should map an edge $uv \in E(T)$ 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 let S be 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 $1 \le i \le k$ 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.

Remark 1. Karzanov [22] shows (with a different algorithm) polynomial solvability of a larger class of 0-extension problems that includes trees as a special case.

We mention two extensions of the problem, *hub-constrained* and *leaf-constrained* hierarchical hubbing, which relate to other natural metrics for network design (cf. the optical design problem in Shepherd and Winzer [26]). While we considered the issue of total bandwidth, it may also be that long paths between terminals are undesirable; e.g., this could lead to transmission delays. 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 $d(h(u), h(v)) \le U(u, v)$. The algorithm described above can be easily modified to find the optimal solution subject to the version where such constraints are imposed.

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 $\mathcal{P}^l$. Moreover, if G has enough capacity to route a given D^l, then it also has enough capacity to support all demands in $\mathcal{U}_T$ via the template $\mathcal{P}^l$. Unfortunately, it is not the case that every $D^l \in \mathcal{U}_T$. Instead, we define a distribution over this class of matrices such that $\bar{D} := \mathbb{E}(D^l)$ lies in $\alpha\mathcal{U}_T$ 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 $\alpha = 8$.

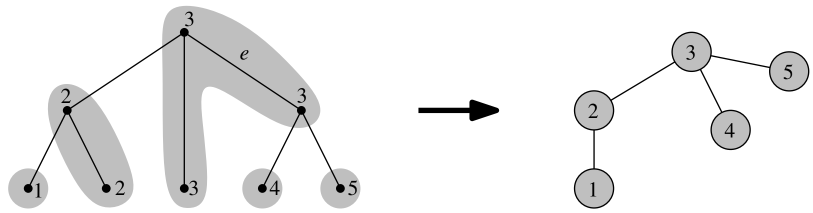

FIGURE 2. A connected labelling and the associated T^l obtained by contracting.

3.4. Connected labellings and hub routings.

DEFINITION 2. A *connected labelling* of a tree T is a function $l: V(T) \to W$ satisfying the properties that $l(w) = w$ for all $w \in W$, and $l^{-1}(w)$ is 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 $l^{-1}(w)$ to obtain a new tree T^l, with $V(T^l) = W$ (see Figure 2). The edges of T^l determine the nonzero demands—if $ij \notin E(T^l)$, then $D^l_{ij} = 0$. Now consider $ij \in E(T^l)$. There is a unique edge $e \in E(T)$ that connects the components $l^{-1}(i)$ and $l^{-1}(j)$. Define $D^l_{ij} = b(e)$.

The optimal solution to route just the single demand matrix D^l simply consists of routing on shortest paths. This has a cost of $C^*(D^l) = \sum_{i,\, j\in W} D^l_{ij} d(i, j)$. 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 $\sum_{wz\in E(T)} b(wz)d(h(w), h(z))$. This means that the optimal solution for the single matrix D^l is exactly a hierarchical hubbing solution where we enforce $h(w) = l(w)$ for each node $w \in V(T)$. It follows that:

LEMMA 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 $u^{\mathcal{Q}}$ 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 $v_0 v_1 \ldots v_m$, 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 $\mathcal{P}^l$ to be the routing template given by the P_{uv}.

LEMMA 3. *The capacities $u^{\mathcal{Q}}$ are enough to support the routing of any $D \in \mathcal{U}_T$ via $\mathcal{P}^l$.*

PROOF. Let D be any T-demand, and let f be any edge of G. Let E' be the set of edges $e = ij$ 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 u and v in T. Let $S(e)$ denote the set of such terminal pairs. Then the total load induced on edge f by demand D via $\mathcal{P}^l$ is at most $\sum_{e\in E'} \sum_{uv\in S(e)} D_{uv} \le \sum_{e\in E'} b(e)$. 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 $D^l_{ij} = b(ij)$ for each edge $ij \in E(T^l)$, the total load does not exceed $\sum_{ij\in E'} D^l_{ij} \le u^{\mathcal{Q}}(f)$ 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 $e = uv \in E(T)$, this is $\sum_{ij\in S(e)} D^l_{ij}$, where, recall that $S(e)$ is the set of terminal pairs separated by e in T. If $l(u) \ne l(v)$, then the only pair in $S(e)$ with nonzero demand in D^l is between $l(u)$ and $l(v)$, 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 (e.g., see Figure 2). But suppose we manage to find a distribution so that the *expected load* 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 $\bar{D}_{ij} = \mathbb{E}(D^l_{ij})$. 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 OPT; i.e., $C^*(\bar{D}) \le \alpha \cdot \text{OPT}$. Since static routings are on shortest paths, we have a simple formula for $C^*(\bar{D})$:

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

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. Thus

$$C^*(\bar{D}) = \sum_{i,\,j\in W} \bar{D}_{ij} d(i,j). \tag{4}$$

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

$$C^*(D^l) = \sum_{i,\,j\in W} D^l_{ij} d(i,j).$$

Taking expectations of both sides and then using (4), we obtain

$$\begin{aligned}\mathbb{E}(C^*(D^l)) &= \sum_{i,\,j\in W} \mathbb{E}(D^l_{ij}) d(i,j)\\ &= \sum_{i,\,j\in W} \bar{D}_{ij} d(i,j)\\ &= C^*(\bar{D}).\end{aligned}$$

It follows from this claim that there must be some l such that $C^*(D^l) \le C^*(D)$. By Lemma 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 $\mathcal{U}_T$ 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 $T\backslash\{e\}$, 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 $A_e(T)$.

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 labelled with a terminal $i \in L_e$ and the other endpoint labelled by a terminal $j \in R_e$, 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 f' between the components $l^{-1}(i)$ and $l^{-1}(j)$ in T.

So the total contribution is

$$\sum_{f'\in E(T)} b(f')\cdot \mathbf{1}_{(\text{one endpoint of } f' \text{ has label in } L_e,\text{ the other in } R_e)} = \sum_{(x,\,y)\in A_e(T)} b(xy)\cdot \mathbf{1}_{l(x)\in L_e \wedge l(y)\in R_e}.$$

Now consider any distribution over the labellings. We are interested in the average, i.e., expected, load on edges of T. By linearity of expectations, this is

$$\sum_{(x,\,y)\in A_e(T)} b(xy)\,\mathbb{P}\big(l(x)\in L_e \wedge l(y)\in R_e\big) = \sum_{(x,\,y)\in A_e(T)} b(xy)\big(\mathbb{P}(l(y)\in R_e) - \mathbb{P}(l(x)\in R_e)\big). \tag{5}$$

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

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 nonleaf node of T to be the root; call it r. For every nonleaf node s, pick one of its children at random, weighting the choices according to the edge capacities, and draw an arrow to it from s. Now, for any node s of T, define $l(s)$ 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 $T\backslash\{e\}$ below e, i.e., not containing the root. It is

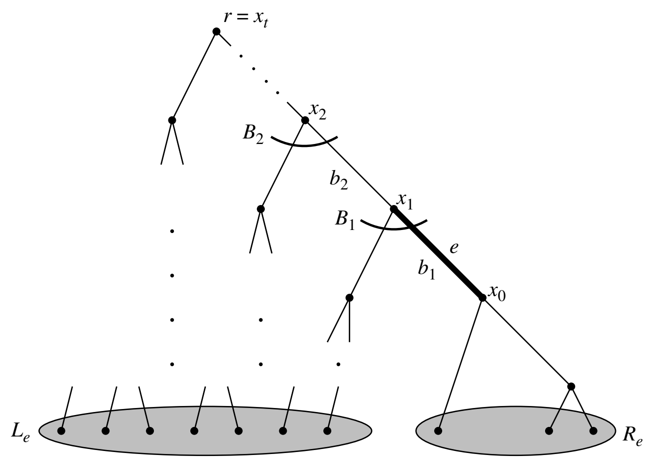

FIGURE 3. Calculating the expected load on edge e.

clear that any edges below e do not contribute to the sum, since walks from both endpoints of such an edge would definitely end up in R_e (the walks cannot go up the tree). Likewise, any edge that is not on, or touching, the path from e to the root cannot contribute—walks from both endpoints would end up in L_e.

Now label the nodes on the path from e to the root by $x_0, x_1, \ldots, x_t = r$, with $e = x_1 x_0$. 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 Figure 3).

There are two types of edges to consider:

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

$$\begin{aligned} &b_i\big(\mathbb{P}(l(x_{i-1}) \in R_e) - \mathbb{P}(l(x_i) \in R_e)\big) \\ &\quad = b_i\big(B_i/b_i \cdot \mathbb{P}(l(x_i) \in R_e) - \mathbb{P}(l(x_i) \in R_e)\big) \\ &\quad = (B_i - b_i)\,\mathbb{P}(l(x_i) \in R_e). \end{aligned}$$

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

$$b(g)\big(\mathbb{P}(l(x_i) \in R_e) - \mathbb{P}(l(z) \in R_e)\big) = b(g)\,\mathbb{P}(l(x_i) \in R_e),$$

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

$$(B_i - b_i)\,\mathbb{P}(l(x_i) \in R_e).$$

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

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

3.7. Trees with unit capacities. If $b(e) = 1$ for all $e \in E(T)$, then we have from Equation (6) that the expected load on any edge is at most

$$2\sum_{i=1}^{t}(B_i - 1)\prod_{j=1}^{i}1\Big/B_j = 2\sum_{i=1}^{t}\prod_{j=1}^{i-1}1\Big/B_j - 2\sum_{i=1}^{t}\prod_{j=1}^{i}1\Big/B_j = 2 - 2\prod_{j=1}^{t}1\Big/B_j \le 2.$$

So $\bar{D}/2 \in \mathcal{U}_T$, 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

$$b(e) \ge \sum_{e' \in \delta_T(y) \setminus \{e\}} b(e'), \tag{7}$$

then $\mathscr{U}_T$ 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 $e \in T$ *wide* if it satisfies $b(e) \geq \frac{1}{2}\sum_{e' \in \delta_T(y)\setminus\{e\}} b(e')$. 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. (However, an edge that was wide may no longer be so after this contraction.) Repeat this process, each time picking a lowest-level wide edge with respect to the current tree, until we obtain a tree $\hat{T}$ with no wide edges. Let $\mathscr{U}_{\hat{T}}$ be the associated demand polytope. Since we only contracted some subset of the wide edges of T, one easily checks that for any $D \in \mathscr{U}_{\hat{T}}$, $D/2 \in \mathscr{U}_T$. 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 $\mathscr{U}_{\hat{T}}$, giving a factor 8 approximation to the T-demand problem.

4. RND for H-topes and tree embeddings. We first discuss an interesting and broader class of robust network design problems. We then show how our algorithm for RND with T-topes can sometimes be used to yield constant factor approximations in this larger class of problems.

4.1. RND for H-topes. Suppose we are given an instance of RND with an undirected topology G and whose demand universe consists of the set of demand matrices that are (fractionally) routable on some given capacitated graph H. In other words, we are asking to design a network on G's topology that can support any traffic pattern which is routable in H. We call the polytope of such demands routable in H the *H-tope* and denote this by $\mathscr{U}_H$. Obviously, H-topes strictly generalize the hose matrices and T-topes.

The other notable instance where a constant factor approximation is known for RND is in the asymmetric hose setting (Gupta et al. [18]). In this case, each node i has two bounds $b^-(i)$, $b^+(i)$, on its ingress/egress capacities. The underlying topology G is still undirected; however, each D_{ij}, D_{ji} represent distinct demands. One easily reduces this problem to the case where for each node one of $b^-(i)$, $b^+(i)$ is 0, and is hence either a sender or a receiver. Thus the class of asymmetric hose matrices can be represented as an H-tope induced by a directed H. There is a central hub h, each sender has a directed arc to h, and each receiver has a directed arc from h. It is natural to ask whether our constant factor results hold also for directed trees H; this remains open.

4.2. A reduction via metric embeddings. In some cases, one can (approximately) represent $\mathscr{U}_H$ via T-topes, and hence our 8-approximation algorithm can be used to give a constant factor approximation for such H-topes. More specifically, suppose that there is a *constant-congestion stochastic embedding* of H into trees as defined below. We define a probability distribution λ_T over trees whose leaf nodes are precisely $V(H)$. In addition, associated to each T there is a mapping of the internal nodes of T to $V(H)$, and a mapping of each $e \in E(T)$ to a path in H between the images of the endpoints of e. Denote this path by $T(e)$; it need not be simple. This induces a mapping from $E(H)$ to paths in H: namely, if $e \in H$, and P is the unique path joining the endpoints of e in T, then the associated path is $T(e) = \bigcup_{f \in P} T(f)$. This can be viewed as a vector, where $T(e)_f$ gives the number of occurrences of f in the path $T(e)$. The *load* of this map on some edge $e \in H$ is $\mathrm{load}_T(e) = \sum_{f \in H} u(f)T(f)_e$, where $u(f)$ is the capacity of f in H. In other words, it is the total capacity that gets mapped to e. The *congestion* of e under the map T is then $\mathrm{con}_T(e) = \mathrm{load}_T(e)/u(e)$. A *low-congestion embedding* is one where $\sum_T \lambda_T \,\mathrm{con}_T(e) = O(1)$ for each edge.

In a breakthrough paper by Räcke [25] (also cf. Andersen and Feige [1]) an equivalence is shown between low-congestion embeddings and low-distortion embeddings. The latter concept is more widely studied and better understood; it asks for maps that bound the stretch on distances between pairs of nodes (distortion). In particular, the existence of constant distortion tree-embeddings for outerplanar graphs (Gupta et al. [19]) immediately implies constant congestion tree-embeddings for this class. (Unfortunately this does not apply much further, since series parallel graphs do not enjoy such embeddings.) Similarly, there exist constant congestion tree embeddings for graphs of bounded pathwidth, via the constant distortion results of Lee and Sidiropoulos [23].

We now show that such embeddings can be used to decompose a given H-tope. For each tree T in the distribution, and each edge $e \in T$, we look at its fundamental cut as induced in H, in other words, the partition S_e, $V - S_e$ of H induced by $T - e$. For the edge e, we use $u^T(e) = u(\delta_H(S_e))$ to define a capacity on e. These capacities induce the T-tope, $\mathcal{U}_T$. It follows that any demand routable in H (indeed, any demand satisfying the cut condition) is also routable in $\mathcal{U}_T$.

Next, define $\mathcal{U}_\lambda = \sum_T \lambda_T \mathcal{U}_T$, where $\sum_T \lambda_T = 1$, $\lambda \geq 0$. For any demand matrix D routable in H, the demand $\lambda_T D$ lies in $\lambda_T \mathcal{U}_T$. Hence, D lies in $\mathcal{U}_\lambda$. Conversely, suppose that some demand matrix D lies in $\mathcal{U}_\lambda$. Then D can be carved up as $D = \sum_T \lambda_T D^T$, where each $D^T \in \mathcal{U}_T$. Consider routing D^T in H using the T-map from H into H. This induces a congestion of at most $\text{con}_T(e)$ on each edge e of H. Hence, taking the λ combination of these flows results in a routing of D in H with congestion $O(1)$. Thus,

$$\mathcal{U}_H \subseteq \mathcal{U}_\lambda \subseteq O(1)\mathcal{U}_H.$$

It is thus sufficient to find a *constant factor approximation to RND for* $\mathcal{U}_\lambda$. We show that this may be done using our 8-approximation for RND on T-topes. The first step is to find the appropriate tree decomposition. This can also be assumed to be of polynomial size (Charikar et al. [5]).

For each T in the decomposition, define $D^{T,l}$ according to the distribution defined in §3. We work with a new distribution where $D^{T,l}$ is chosen with the same probability, scaled back by λ_T. It follows that

$$\bar{D} = \mathbb{E}_{T,l}[D^{T,l}] = \sum_T \lambda_T \mathbb{E}_l[D^{T,l}] \in 8 \sum \lambda_T \mathcal{U}_T = 8\mathcal{U}_\lambda.$$

Hence an optimal cost (fractional, shortest path) routing for $\bar{D}$ is at most $8 \cdot \text{OPT}$. Moreover, an optimal cost routing is equal to the convex combination (using λ_T and individual T-distributions) of optimal cost routings for the $D^{T,l}$. Hence there is some $D^{T,l}$ whose cost to route in G is at most $8 \cdot \text{OPT}$. As seen in Lemma 2, the optimal hierarchical hubbing solution for T costs no more than the optimal cost routing of any individual $D^{T,l}$. Moreover, such a routing for a T, l pair, for any T, yields a feasible capacity reservation vector for $\mathcal{U}_H$. This follows simply because $\mathcal{U}_H \subseteq \mathcal{U}_T$ for any T. This completes the argument.

It is worth emphasizing that if the topology graph G itself admits an $O(1)$-congestion tree embedding, then an $O(1)$-approximation for *any* (polytime separable) demand universe is possible. This follows similarly to Gupta's $O(\log n)$-approximation for general RND, and is spelled out in the final section of Goyal et al. [13].

5. Conclusion. We have described an algorithm that guarantees a robust network design for the class of tree-demands that is within a factor of 8 of the optimal (in fact, within a factor of 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 a counterexample. The VPN theorem (Goyal et al. [12]) implies that this is true in the case that T is a star; it could be that a *generalized VPN conjecture* holds for T-topes. Another interesting direction discussed in the last section is whether RND for directed H-topes admits an $O(1)$-approximation.

Acknowledgments. The authors 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. They also thank Chandra Chekuri for very helpful suggestions, and thank the anonymous referees for their helpful comments. The first author was supported by a Quebec Merit Scholarship for Foreign Students when this work was carried out. The second author is grateful for support from the Canadian Natural Sciences and Engineering Research Council [Discovery Grant G212567].

References

[1] Andersen R, Feige U (2009) Interchanging distance and capacity in probabilistic mappings. Preprint, arXiv http://arxiv.org/abs/0907.3631.

[2] Andrews M (2004) Hardness of buy-at-bulk network design. *Proc. 45th IEEE Sympos. Foundations Comput. Sci.* (*FOCS*) (IEEE Computer Society, Washington, DC), 115–124.

[3] Ben-Ameur W, Kerivin H (2003) New economical virtual private networks. *Comm. ACM* 46(6):69–73.

[4] Calinescu G, Karloff H, Rabani Y (2001) Approximation algorithms for the 0-extension problem. *Proc.* 12*th ACM-SIAM Sympos. Discrete Algorithms* (*SODA*) (SIAM, Philadelphia), 8–16.

[5] Charikar M, Chekuri C, Goel A, Guha S, Plotkin S (1998) Approximating a finite metric by a small number of tree metrics. *Proc.* 39*th Annual IEEE Sympos. Foundations Comput. Sci.* (*FOCS*) (IEEE Computer Society, Washington, DC), 379–388.

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

[7] Duffield NG, Goyal P, Greenberg A, Mishra P, Ramakrishnan KK, van der Merwe JE (1999) A flexible model for resource management in virtual private networks. *Proc. ACM SIGCOMM* (ACM, New York), 95–108.

[8] Eisenbrand F, Grandoni F (2005) An improved approximation algorithm for virtual private network design. *Proc.* 16*th Annual ACM-SIAM Sympos. Discrete Algorithms* (*SODA*) (SIAM, Philadelphia), 928–932.
[9] Eisenbrand F, Grandoni F, Rothvoß T, Schäfer G (2008) Approximating connected facility location problems via random facility sampling and core detouring. *Proc.* 19*th Annual ACM-SIAM Sympos. Discrete Algorithms* (*SODA*) (SIAM, Philadelphia), 1174–1183.
[10] Eisenbrand F, Happ E (2006) Provisioning a virtual private network under the presence of noncommunicating groups. *Proc.* 6*th Internat. Conf. Algorithms and Complexity*, Lecture Notes in Computer Science, Vol. 3998 (Springer, Berlin, Heidelberg), 105–114.
[11] Fingerhut JA, Suri S, Turner JS (1997) Designing least-cost nonblocking broadband networks. *J. Algorithms* 24(2):287–309.
[12] Goyal N, Olver N, Shepherd FB (2008) The VPN conjecture is true. *Proc.* 40*th ACM Sympos. Theory Comput.* (*STOC*) (ACM, New York), 443–450.
[13] Goyal N, Olver N, Shepherd FB (2011) Dynamic vs oblivious routing in network design. *Algorithmica* 61(1):161–173.
[14] Grandoni F, Kaibel V, Oriolo G, Skutella M (2008) A short proof of the VPN tree routing conjecture on ring networks. *Oper. Res. Lett.* 36:361–365.
[15] Grandoni F, Rothvoß T (2010) Network design via core detouring for problems without a core. *Proc.* 37*th Internat. Colloquium on Automota, Languages and Programming* (*ICALP*) (Springer, Berlin, Heidelberg), 490–502.
[16] Gupta A (2004) Private communication.
[17] Gupta A, Kleinberg J, Kumar A, Rastogi R, Yener B (2001) Provisioning a virtual private network: A network design problem for multicommodity flow. *Proc.* 33*rd ACM Sympos. Theory Comput.* (*STOC*) (ACM, New York), 389–398.
[18] Gupta A, Kumar A, Pál M, Roughgarden T (2007) Approximation via cost sharing: Simpler and better approximation algorithms for network design. *J. ACM* 54(3):11.
[19] Gupta A, Newman I, Rabinovich Y, Sinclair A (1999) Cuts, trees and l_1-embeddings of graphs. *Proc.* 40*th Annual IEEE Sympos. Foundations Comput. Sci.* (*FOCS*) (IEEE Computer Society, Washington, DC), 399–408.
[20] Hurkens CAJ, Keijsper JCM, Stougie L (2007) Virtual private network design: A proof of the tree routing conjecture on ring networks. *SIAM J. Discret. Math.* 21(2):482–503.
[21] Italiano G, Leonardi S, Oriolo G (2006) Design of trees in the hose model: The balanced case. *Oper. Res. Lett.* 34(6):601–606.
[22] Karzanov A (1998) Minimum 0-extensions of graph metrics. *Eur. J. Combinatorics* 19(1):71–101.
[23] Lee JR, Sidiropoulos A (2009) On the geometry of graphs with a forbidden minor. *Proc.* 41*st Annual ACM Sympos. Theory Comput.* (*STOC*) (ACM, New York), 245–254.
[24] Räcke H (2002) Minimizing congestion in general networks. *Proc.* 43*rd Sympos. Foundations Comput. Sci.* (*FOCS*) (IEEE Computer Society, Washington, DC), 43–52.
[25] Räcke H (2008) Optimal hierarchical decompositions for congestion minimization in networks. *Proc.* 40*th Annual ACM Sympos. Theory Comput.* (*STOC*) (ACM, New York), 255–264.
[26] Shepherd FB, Winzer PJ (2006) Selective randomized load balancing and mesh networks with changing demands. *J. Optical Networking* 5(5):320–339.
[27] Valiant LG (1982) A scheme for fast parallel communication. *SIAM J. Comput.* 11(2):350–361.