Routing and Network Design with Robustness to Changing or Uncertain Traffic Demands*

Chandra Chekuri†

July 27, 2007

Abstract

A new class of network design problems were introduced by Fingerhut et al. [26], and independently by Duffield et al. [20], to address, among other things, the issue of uncertainty in the demand matrix. The so-called hose model (the term was coined in [20]) for demand matrices from [26, 20] was subsequently generalized to the polyhedral model by Ben-Ameur and Kerivin [11, 10]. In a different direction, Räcke [52] showed the existence of good randomized oblivious routings in all undirected graphs. This was followed by a proof of the polynomial time solvability of an optimal oblivious routing scheme [7].

One can view the above developments in a common framework of robust optimization. We give a survey of these developments and related work with the aim of providing a unified picture. We also highlight the remaining open problems.

1 Introduction

Many traditional routing and network design problems in combinatorial optimization model traffic demands as inputs with the assumption that they are known in advance. In many practical settings this assumption is not accurate, nevertheless it is made for simplicity, convenience, and tractability. Moreover, estimating and modeling real world traffic is a difficult task. In recent years some new approaches to modeling and designing for traffic variations have been attempted. The impetus for this line of work came from Fingerhut et al. [26] who were interested in ATM broadband networks and from Duffield et al. [20] who were interested in virtual private networks (VPNs). In both works, the uncertainty in the demand matrix was the main driver in the introduction of the *hose* model [20] for traffic demands. This led to some interesting new network design problems and a simple and as yet unresolved conjecture to be discussed later. Several variants and generalizations have been proposed in the literature which we loosely characterize as *robust* network design. The goal is to design a network that can support not just one given traffic demand matrix but a class of different demand matrices. In a different direction, Räcke, in a breakthrough paper [52], proved the existence of poly-logarithmic competitive randomized *oblivious* routing schemes in undirected graphs. Oblivious routing can be viewed as providing a robust routing solution for a class of demand matrices - we make this precise later.

Robust network design and oblivious routing address different objective functions; in network design the goal is to minimize the cost of the network and in oblivious routing the goal is to minimize congestion. Nevertheless, they share certain features; both have to consider a class of traffic demands and models for them. The three questions that are of interest are:

*The author welcomes comments and corrections. A revised version, if necessary, will be made available through the author's web page.

†Dept. of Computer Science, University of Illinois, Urbana, IL 61801. Email: chekuri@cs.uiuc.edu.

- What classes of demand matrices are interesting and useful and can be specified in a computationally sound way?
- What are the constraints on routing?
- What is the objective function?

In this article we survey the known results on the relevant optimization problems with a focus on their worst case complexity and approximability - we do not attempt to survey the computational and modeling aspects of these problems. We refer the interested reader to [10, 24, 3] for some more pointers. In particular, we emphasize the polyhedral approach to the problems which allows one to make connections between some results for robust network design and oblivious routing. We also highlight a few interesting open problems.

1.1 Notation

We assume that the reader is familiar with basic concepts of flows in networks including multi-commodity flows (see [1, 54] for the relevant background). Let $G = (V, E)$ be a graph that represents a network. Unless mentioned explicitly we assume that G is undirected. For an edge $e \in E$, $c(e)$ and $u(e)$ denote the cost and capacity of edge e and are assumed to be non-negative. Throughout we use n to denote $|V|$. A *demand* matrix D for a graph G is a non-negative $n \times n$ matrix. $D(i, j)$ represents the traffic demand from node i to j. Note that $D(i, j)$ and $D(j, i)$ represent two separate commodities in terms of flow. In some settings it is more natural to have a single commodity for each *unordered* pair of nodes; note that this is relevant only for undirected graphs. In that case we say that G and D are a *symmetric* instance and use the notation $D(ij)$ for the single demand for the unordered pair ij. The general cases is referred to as the *asymmetric* case. We caution the reader that even if $D(i, j) = D(j, i)$ the instance can be asymmetric, thus the term is not referring to the matrix itself. We adopt this convention to be consistent with some previous work [26, 33]. In [18] a different notation (oriented and unoriented) is used to distinguish between the symmetric and asymmetric cases.

Given a capacitated graph G and a demand matrix D we say that D is *fractionally* routable in G if there is a feasible multi-commodity flow in G for D. D is *unsplittably* routed if there is a feasible multi-commodity flow for D in G in which all the flow for each commodity (i, j) is routed on a single path. In the symmetric setting the commodities are the unordered pairs ij, $1 \le i < j \le n$.

A demand matrix D may not be (fractionally) routable in a graph G; we say that D is routable in G with *congestion* $\lambda \ge 1$ if D has a feasible routing in G with each edge capacity increased by a multiplicative factor of λ.

We use $\mathcal{D}$ to denote a set of demands matrices, either discrete or continuous.

2 Demand Matrices, Routing Constraints, and Problems

2.1 Models for Demand Matrices

In this section we discuss some models for specifying $\mathcal{D}$, the set of demand matrices that one wants the network to be able to support. Three models of interest are:

- the discrete model: $\mathcal{D} = \{D_1, D_2, \ldots, D_k\}$ is a finite collection of demand matrices.
- the hose model and variants.
- the polyhedral or convex model.

The discrete model does not require much explanation. It is not clear how useful this model is in terms of modeling robustness. The matrices in $\mathcal{D}$ might not exhibit any smoothness properties which might make certain optimization problems trivially hard. On the other hand certain problems become easier with a discrete set and one may be able to approximate a continuous set with a sufficiently large discrete set.

The hose model(s): This model was introduced in [26, 20]; the name was coined in [20]. This model is motivated and inspired by current data networks in which users and corporations have a fixed capacity connection to the Internet. These capacity upper bounds are sometimes the result of the physical limitations of the devices (say a modem line or an Ethernet connection) or the result of a negotiated purchase from a service provider (as is common for corporations). Thus, in this setting, each node $i \in V$ has a bound $b(i)$ on the total traffic that can terminate at i. Given only this information it is natural to consider $\mathcal{D}$ to be the set of all demand matrices D which respect the bound constraints at each node i.

In the *symmetric* setting in undirected graphs $D \in \mathcal{D}$ iff

$$\sum_{j \neq i} D(ij) \leq b(i) \qquad \forall i \in V.$$

In the asymmetric setting there are separate bounds on the capacities of the incoming and outgoing for each node i. This is also relevant in practice where the upload and download capacities may be different. That is, for each user i there are two bounds $b_{in}(i)$ and $b_{out}(i)$ and $D \in \mathcal{D}$ iff

$$\sum_{j \neq i} D(i,j) \leq b_{out}(i) \quad \text{and} \quad \sum_{j \neq i} D(j,i) \leq b_{in}(i) \qquad \forall i \in V.$$

Polyhedral model: We observe that the hose model imposes some simple linear constraints on the entries of the demand matrices in $\mathcal{D}$. Thus, a natural and powerful model to specify $\mathcal{D}$ is via linear constraints on the n^2 variables $D(i,j)$, $1 \leq i,j \leq n$.[1] Let $P(\mathcal{D})$ be a polyhedron in the non-negative orthant on the variables $D(i,j)$. The polyhedron $P(\mathcal{D})$ could be described via an explicit system of linear inequalities in which case the size of this system is part of the input. Alternatively one could assume that $P(\mathcal{D})$ is available via a separation oracle. Recall that a separation oracle for a polyhedron P in $\mathcal{R}^d$ is an algorithm that when given a point $x \in \mathcal{R}^d$ does the following: if $x \in P$ it says yes and if $x \notin P$ the algorithm returns a hyperplane that separates x from P. By the equivalence of separation and optimization for linear programs [29], one could use the separation oracle to optimize any given linear function over $P(\mathcal{D})$; we see useful applications of this later in the article. We could further generalize the model to include convex bodies equipped with a separation oracle. We say that $\mathcal{D}$ is polyhedral if there is a polyhedron $P(\mathcal{D})$ such that $D \in \mathcal{D}$ iff $D \in P(\mathcal{D})$. The polyhedral model for traffic demands was introduced by Ben-Ameur and Kerivin [11, 10].

We remark that if $\mathcal{D}_1$ and $\mathcal{D}_2$ are polyhedral then so is $\mathcal{D}_1 \cap \mathcal{D}_2$. We describe a few interesting and useful classes of demand matrices that are polyhedral.

Example 1. The hose model in the symmetric setting allows all demand matrices D such that

$$\sum_{j \neq i} D(ij) \leq b(i) \qquad \forall i \in V.$$

[1] We could omit the uninteresting variables $D(i,i)$, $1 \leq i \leq n$ but we keep them for simplicity of notation. For symmetric demands we could restrict attention to $n(n-1)/2$ variables $D(ij)$, $1 \leq i < j \leq n$ but we avoid these details.

Thus, for a pair ij there exists a $D \in \mathcal{D}$ where $D(ij) = \min\{b(i), b(j)\}$. However in many practical settings $D(ij)$ may be much less than this because $b(i)$ and $b(j)$ are usually larger to accommodate communication with many different entities. Thus a natural set of constraints to add is

$$D(ij) \leq B(ij) \qquad \forall ij$$

where B is a matrix of upper bounds. Similarly one can add lower bounds if there is information available.

Example 2. Let D^* be a specific traffic matrix that is obtained from some estimates with potentially some errors. One could then consider the class $\mathcal{D}$ of matrices that are close to D^* in either a relative or additive sense.

$$\mathcal{D} = \{D \mid (1 - \epsilon)D^*(i,j) \leq D(i,j) \leq (1 + \epsilon)D^*(i,j), 1 \leq i \leq j \leq n\}$$

for some $\epsilon < 1$ or

$$\mathcal{D} = \{D \mid \min\{0, D^*(i,j) - d\} \leq D(i,j) \leq D^*(i,j) + d, 1 \leq i \leq j \leq n\}$$

for some $d > 0$. These and other variants are clearly expressible in the polyhedral model.

Example 3. Let G be a capacitated graph with edge capacities $u : E \rightarrow \mathcal{R}^+$. Let

$$\mathcal{D}_u = \{D \geq 0 \mid D \text{ is fractionally routable in } G\}.$$

The following is a simple exercise.

Proposition 2.1 *$\mathcal{D}_u$ is polyhedral and has an explicit representation of size polynomial in n.*

Single-Source Model: We say that $\mathcal{D}$ is a single-source model if there exists a node $s \in V$ such that for all $D \in \mathcal{D}$, $D(i,j) = 0$ if $i \neq s$.

2.2 Routing Models

In routing and network design problems one may have constraints on how the flow for various commodities is constrained. We already mentioned fractional and unsplittable routings for a given demand matrix D in G. Here we are interested in handling multiple demand matrices $\mathcal{D}$. The question is how the routings for different demand matrices in $\mathcal{D}$ are jointly constrained. Below we discuss several possibilities that are either natural or motivated by applications. The notation below is borrowed from [38].

- (FR) For every $D \in \mathcal{D}$ a different fractional routing is allowed.
- (UFR) For each $D \in \mathcal{D}$ a different unsplittable routing is allowed.
- (SPR) Single path or fixed unsplittable routing. For each pair (i,j) a fixed path $P(i,j)$ (or a path $P(ij)$ in the symmetric case) is used for all $D \in \mathcal{D}$.
- (MPR) Proportional routing along multiple paths. For each pair (i,j) a *template* fractional flow of one unit between (i,j) is fixed. In other words, there are paths $P_1(i,j), P_2(i,j), \ldots, P_h(i,j)$ and associated non-negative numbers $f_1(i,j), f_2(i,j), \ldots, f_h(i,j)$ such that $\sum_{k=1}^{h} f_k(i,j) = 1$. Alternatively, these numbers can be interpreted as probability distribution on paths between i and j. For any $D \in \mathcal{D}$ the routing of D for (i,j) is split according to $f(i,j)$, that is, $D(i,j)f_r(i,j)$ flow is sent along the path in $P_r(i,j)$.

- (TR) Tree routing. This applies only to undirected graphs and is a further specialization of single-path routing. The union of the fixed paths $P(i,j)$ induce a tree. Although this model is more natural for the symmetric case, it does apply to the asymmetric case as well but it then constraints $P(i,j)$ to be the same as $P(j,i)$.

- (TTR) Terminal tree routing. This is also a specialization of SPR routing. For each $i \in V$, the union of the paths $P(i,j)$, $j \in V$ induce a tree rooted at i.

We use the term *dynamic* to refer to routings that can change with D (FR, UFR) while the other ones are *static* (SPR, TR, TTR) or *stable* (MPR).

2.3 Optimization Problems

We now describe the broad classes of optimization problems that arise from the models for demand matrices and routing.

Cost models for network design: In network design the high level goal is to install capacity in a network to enable the routing of demands. The easiest problems arise when the cost of installing a capacity of $x(e)$ units on a link e is a simple linear function $c(e)x(e)$. Another important model that arises often in applications, due to discrete capacity constraints and economies of scale, is the buy-at-bulk or concave cost model. Here, for each each e there is a real valued function $f_e : \mathcal{R}^+ \rightarrow \mathcal{R}^+$ which specifies the cost $f_e(x)$ to install a capacity of x units on link e. To model economies of scale we assume that f_e is concave, or more generally sub-additive ($f(x+y) \leq f(x) + f(y)$). If for all e, $f_e = c(e)f$ for some sub-additive function f we obtain the so-called uniform model, the general case is referred to as the non-uniform case. There are several interesting functions that span the spectrum from the simplest linear costs to the most general buy-at-bulk costs and we will not discuss these in detail here. Assuming we have a cost model one obtains the following general problem.

Robust Network Design Problem: Given a graph G, a set of demand matrices $\mathcal{D}$, a routing model, and a cost model, find a minimum cost installation of edge capacities in G to support a feasible routing of all demand matrices $D \in \mathcal{D}$ subject to the routing model constraints.

The VPN Design problem is a special case of the above question where $\mathcal{D}$ is obtained from one of the hose models and the costs are linear. Further the emphasis is on the SPR and more restricted models such as TTR and TR although other routing models are also considered. The hose model has attracted substantial attention for both its simplicity and intuitive appeal and we treat it separately in Section 4. We treat the general robust network design question in Section 3.

The second optimization problem that we consider has a different objective function, specifically, to minimize congestion.

Oblivious Routing to Minimize Congestion: Given a graph $G = (V,E)$ with capacities $u : E \rightarrow \mathcal{R}^+$, a demand class $\mathcal{D} \subseteq \mathcal{D}_u$ and a static or stable routing model, find the minimum congestion λ^* such that all $D \in \mathcal{D}$ can be routed in G with congestion at most λ^* according to the routing model constraints.

Two comments on the above problem definition are in order. First, we observe that oblivious routing is not meaningful for dynamic routing models, for, in those models, each $D \in \mathcal{D}$ can be treated separately. Thus the only routing models of interest are the static or stable ones. Strong lower bounds exist for the static unsplittable model and thus the main focus will be on the MPR model. Second, we restrict attention to $\mathcal{D} \subseteq \mathcal{D}_u$ in order to use an absolute congestion bound instead of a relative congestion bound. Given an arbitrary demand class $\mathcal{D}$ consider a new demand class $\mathcal{D}'$ obtained as follows. For $D \in \mathcal{D}$ let $\beta_D = 1$ if $D \in \mathcal{D}_u$ and otherwise let β_D be the largest real number in $[0, 1]$ such that $\beta_D D \in \mathcal{D}_u$. Define $\mathcal{D}' = \{\beta_D D \mid D \in \mathcal{D}\}$. An oblivious routing for $\mathcal{D}'$ with absolute congestion bound λ achieves the same bound for $\mathcal{D}$ in the relative sense.

We note that for the above two problems to be computationally well defined, the demand class $\mathcal{D}$ can be specified in an appropriate way. If $\mathcal{D}$ is a discrete set then it is given explicitly and becomes part of the input size. For polyhedral $\mathcal{D}$, a separation oracle can be given, or $P(\mathcal{D})$ can be explicitly specified as a system of linear inequalities in which case the size of the system becomes part of the input size.

3 Robust Network Design with Polyhedron of Demand Matrices

In this section we survey results for the robust network design problem in its full generality. Most of the work in the robust network design area has focused on linear costs and we discuss it first before mentioning some simple results that can be obtained for buy-at-bulk costs.

Given a set of demand matrices $\mathcal{D}$, let $\mathcal{U}(\mathcal{D})$ denote the set of capacity reservations in G that are feasible for $\mathcal{D}$. In other words, if $u \in \mathcal{U}(\mathcal{D})$ then the graph G with edge capacities given by u admits a feasible multi-commodity flow for each $D \in \mathcal{D}$.

Proposition 3.1 *For any $\mathcal{D}$, $\mathcal{U}(\mathcal{D})$ is convex for both the FR and MPR models of routing.*

In addition to the graph G and the set of demand matrices $\mathcal{D}$, the input to the robust network design problem consists of non-negative edge costs $c : E \rightarrow \mathcal{R}^+$. Note that for any fixed demand matrix D the network design problem is trivial. Simply route each commodity (i, j) along a shortest path from i to j according to edge lengths given by c. Thus the routing is unsplittable. The capacity installed on e is simply the total demand routed on e by the shortest path routing.

Consider the FR and MPR models. The optimization problem we need to solve is easy to state with the notation in place.

$$\begin{aligned} &\min c \cdot u \\ &\text{subject to} \\ &\qquad u \in \mathcal{U}(\mathcal{D}). \end{aligned}$$

This specific problem was first formulated in [10]. Since $\mathcal{U}(\mathcal{D})$ is convex, to solve the above problem it is necessary and sufficient to have a separation oracle for $\mathcal{U}(\mathcal{D})$. As we remarked earlier, this follows from the equivalence of separation and optimization [29].

We observe that if $\mathcal{D}$ is a discrete set then one can write an explicit large formulation for $\mathcal{U}(\mathcal{D})$ (left as an exercise to the reader) and hence the optimization problem can be solved in polynomial time. Now we consider the polyhedral and hose models and ask whether the separation problem can be solved in polynomial time.

Theorem 3.2 ([18]) *The separation problem for $\mathcal{U}(\mathcal{D})$ is co-NP hard for the single-source hose model in the FR model of routing even in undirected graphs and hence the robust network design problem with linear costs in the FR model is co-NP hard.*

One of the difficulties in proving the above theorem is the fact that the routing is allowed to be fractional and varies with the demand matrix. Therefore it helps to restrict attention to the single-source case where one can work with cuts instead of flows; here we rely on the maxflow-mincut theorem for single-commodity flows.

The reduction in [18] shows that if the graph expansion problem is co-NP hard to approximate to within a factor of $\alpha > 1$ then the robust network design problem is hard to approximate to within a factor of $2\alpha/(1+\alpha)$. It was erroneously claimed in [18] that a super-constant hardness was shown for expansion assuming the unique-games conjecture [42]; the hardness was shown only for the non-uniform sparsest cut problem []. Although there is no known hardness of approximation for the problem of computing the expansion of a given graph, the current best approximation ratio is $O(\sqrt{\log n})$ [6].

In contrast to the hardness for the FR model of routing, there is a polynomial time algorithm for the robust network design problem in the MPR model. Erlebach and Rüegg [24], and independently Hurkens et al. [38], obtained a polynomial time algorithm for demand matrices in the hose model. The fact that a separation oracle exists for any $P(\mathcal{D})$ follows indirectly from the work on oblivious routing [7, 5] and we discuss this in more detail in Section 5. The following theorem is explicitly pointed out in [10].

Theorem 3.3 ([10]) *The separation problem for $\mathcal{U}(\mathcal{D})$ is polynomial time solvable in the MPR model even in directed graphs when $P(\mathcal{D})$ has a separation oracle. Thus in the MPR model there is a polynomial time algorithm for robust network design with polyhedral demands.*

We observe that a compact formulation also exists if $P(\mathcal{D})$ is given explicitly by a polynomial number of constraints. This also follows from the work on oblivious routing that will be discussed in Section 5.

In [24] it is observed that additional constraints on the capacities can be also taken into account; in particular, they consider upper bounds on the capacity reservation on the edges. To see why these additional constraints are easy to handle, consider the more general optimization problem given below.

$$\begin{aligned} &\min c \cdot u \\ &\text{subject to} \\ &\quad u \in \mathcal{U}(\mathcal{D}) \\ &\quad u \in P_u, \end{aligned}$$

where P_u is some arbitrary polyhedron that constrains u. To solve the above problem it is necessary and sufficient to have separation oracles for $\mathcal{U}(\mathcal{D})$ and P_u. Clearly, upper bounds on u can be easily handled as simple linear constraints.

Now we consider the unsplittable routing models.

Theorem 3.4 ([33]) *The robust network design problem is NP-hard in the unsplittable routing model (both static and dynamic) with linear costs even for the single source hose model. Moreover the single-source problem is at least as hard to approximate as the Steiner tree problem.*

The above theorem follows from a reduction from the Steiner tree problem [33]. Given a rooted Steiner tree problem instance $G = (V, E)$ with root r and terminals $S \subseteq V$, we construct an instance of a single-source robust network design problem. The graph and the edge costs remain the same. We set up $\mathcal{D}$ with $b(r) = 1$, $b(i) = 1$ for each $i \in S$, and $b(j) = 0$ for $j \in V - (S \cup \{r\})$. Consider a demand matrix D^i with $D^i(ri) = 1$ and $D^i(rj) = 0, j \neq i$. Since $D^i \in \mathcal{D}$ for each i, we see that the path $P(ri)$ needs to have a capacity reservation of at least one unit on all its edges. Thus the path collection $\{P(ri) \mid i \in S\}$ induces a Steiner tree in G. Moreover it is easy to see that a Steiner tree with unit capacity is sufficient to support $\mathcal{D}$. Thus a minimum cost capacity reservation for $\mathcal{D}$ is equivalent to a minimum cost Steiner tree for S.

We remark that the above reduction can be easily generalized to show that the robust network design problem in the SPR model is at least as hard as the Steiner forest problem — note that this reduction requires us to use the polyhedral model. In directed graphs the Steiner forest problem is hard to approximate to within almost polynomial factors [19] and hence the main interest is in the undirected case.

What can we say about approximation algorithms for the above problems? Gupta [31] observed the following. Note that the approximation holds for all routing models.

Theorem 3.5 ([31]) *There is an $O(\log n)$ approximation for the robust network design problem in undirected graphs with linear costs for polyhedral demands in all routing models. Moreover the cost of an optimal solution in the SPR model is $O(\log n)$ times the cost of an optimal solution in the FR model.*

We give a sketch of the proof of the above theorem. First we make a simple observation.

Proposition 3.6 *For the robust network design problem in undirected graphs with linear or uniform buy-at-bulk costs, we can assume without loss of generality that $G = (V, E)$ is a complete graph and that c satisfies the triangle inequality and hence induces a metric on V.*

We reduce the problem to instances in which the input graph is a tree. This is accomplished, in by now, standard way. The finite metric induced on V by c (from Proposition 3.6) can be probabilistically approximated by a dominating tree metric on V [8]. Using [25], the increase in the cost is, in expectation, an $O(\log n)$ multiplicative factor. Thus it suffices to solve the problem on a tree. Observe that routing on a tree is unsplittable and static. If G is a tree then what is the capacity $u(e)$ that needs to be installed on an edge $e \in E$? Let $V_1(e)$ and $V_2(e)$ be the partition of the vertex set obtained by removing e from T. In the symmetric setting, we claim that

$$u(e) = \max_{D \in \mathcal{D}} \sum_{i \in V_1(e), j \in V_2(e)} D(ij)$$

is necessary and sufficient (a similar bound can be derived for the asymmetric setting). Necessity is clear. For sufficiency, consider an arbitrary $D \in \mathcal{D}$; note that the routing is fixed on a tree and the total flow on an edge $e \in E$ is $D(e) = \sum_{i \in V_1(e), j \in V_2(e)} D(ij)$. From the definition of $u(e)$, $D(e) \leq u(e)$ for all $e \in E$. Therefore all $D \in \mathcal{D}$ can be routed with capacities given by u. Thus it remains to show how the $u(e)$ can be computed. We observe that $u(e)$ is the maximum of a linear objective function over the polytope $P(\mathcal{D})$ and hence a separation oracle for $P(\mathcal{D})$ implies a polynomial time algorithm to solve for $u(e)$. The algorithm can be derandomized – see [13, 25].

Open Problems: The main open problems are to resolve the approximability of the robust network design with linear costs. The VPN problem is a special case for which constant factor approximations are known and we discuss this in more detail in Section 4.

- Is there an $O(1)$ approximation for the robust network design problem for polyhedral demands in the FR model? Is the problem APX-hard in undirected graphs? Is there an $O(1)$ approximation for the single-source polyhedral demands case? Does the problem admit a non-trivial approximation in directed graphs?

- Is there an $O(1)$ approximation for the robust network design problem in the SPR model.

Buy-at-bulk Costs: For buy-at-bulk costs, the basic network design problem is NP-hard even with a single demand matrix; the current best inapproximability results are due to Andrews [4]. He showed that the uniform cost problem is $\Omega(\log^{1/4-\epsilon} n)$ hard and the non-uniform problem is $\Omega(\log^{1/2-\epsilon} n)$ hard under the assumption that $NP \not\subseteq ZPTIME(n^{\text{polylog}(n)})$. Nevertheless, the simple technique of reducing the problem to a tree gives an $O(\log n)$ approximation even for the robust network design problem with uniform buy-at-bulk costs. The proof is essentially the same as the one for linear costs.

Theorem 3.7 ([31]) *There is an $O(\log n)$ approximation for the robust network design problem in undirected graphs with uniform buy-at-bulk costs for polyhedral demands in all routing models.*

Open Problem: Is there a poly-logarithmic approximation for the non-uniform buy-at-bulk cost model?

The problem is open for all demand models and routing models. Some recent work on the non-uniform buy-at-bulk problem for a single demand matrix [14, 15] may provide some tools to study these problems. We believe that a poly-logarithmic approximation can be obtained for the special case of the hose model. Details will appear elsewhere.

4 VPN Design

In this section we focus on the VPN problem which is a special case of the robust network design problem where $\mathcal{D}$ is one of the hose models and costs are linear. Further we restrict attention to undirected graphs. The VPN problem has attracted substantial attention in the networking, algorithms and mathematical programming communities after the introduction of the hose model in [20]. We note that some of the same problems and algorithms for them were studied earlier in [26]. The work of Gupta et al. [33] brought the mathematical models and algorithmic problems to a wider audience; a simple but interesting result (see Theorem 4.1), first shown in [26], was rediscovered in [33].

We treat the symmetric (each node i has a single bound $b(i)$) and asymmetric (each node i has separate bounds $b_{in}(i)$ and $b_{out}(i)$) cases separately. We remark that in most of the literature it is assumed that $b(i) > 0$ for a subset of nodes $W \subseteq V$ and the nodes in W are referred to as terminals. We simplify the discussion by not distinguishing the terminals from the other nodes.

4.1 Symmetric Case

The input to a symmetric VPN problem consists of an undirected graph $G = (V, E)$ with edge costs $c : E \to \mathcal{R}^+$ and bounds $b(i) \geq 0$, $i \in V$. Each routing model gives rise to a different problem. For a given instance, let $OPT(R)$ denote the optimum solution value with R being the constraint on the routing model. (Notation borrowed from [38]). It is easy to see the following set of inequalities since the routing constraints get stricter from left to right:

$$OPT(FR) \leq OPT(MPR) \leq OPT(SPR) \leq OPT(TTR) \leq OPT(TR).$$

With the above notation it is relatively easy to summarize the known results and state the main open problems. We mention results for computing the various quantities above — however, an algorithm that computes or approximates one of the quantities also yields a witness to the value.

Theorem 4.1 ([26, 33]) *There is a polynomial time algorithm to compute $OPT(TR)$. Moreover $OPT(TTR) = OPT(TR) \leq 2(1 - 1/n)OPT(FR) \leq 2(1 - 1/n)OPT(SPR)$.*

The above theorem therefore yields a 2-approximation to $OPT(SPR)$. The algorithm in [26, 33] for $OPT(TR)$ is the following. First, consider an arbitrary spanning tree T as a solution for the given instance of tree reservation. The capacity reservation on the edges of T, and hence the cost of the reservation $c(T)$, can be computed as follows. For $e \in E(T)$ let V_1 and V_2 be the nodes of the two sub-trees obtained from $T - e$. Then $u(e) = \min\{b(V_1), b(V_2)\}$ is necessary and sufficient. For $i \in V$, let T_i be a shortest path tree from i with edge lengths given by c. Output the tree with the least cost among T_i, $i \in V$. It is shown in [33] that this gives a solution of cost $OPT(TR)$. From Proposition 3.6, we can assume without loss of generality that each T_i is a star rooted at i. Therefore this routing is also referred to as *hub routing* since all paths in T_i go through the hub i. Hub routing is useful in some applications (see [51] and references therein).

We give the simple proof that $OPT(TR) \leq 2OPT(SPR)$. Since $OPT(TR)$ is achieved by a shortest path tree rooted at some node, we restrict our attention to the trees $T_i, i \in V$ and further assume that each of them is a star. Let $c(T_i)$ denote the cost of the reservation for T_i. We can assume without loss of generality that for each $i \in V$, $b(i) \leq \sum_{j \neq i} b(j)$. The cost of the reservation for T_i, noting that it is a star, can be seen to be $\sum_j b(j)c(vj)$.

Now consider the demand matrix $D \in \mathcal{D}$ where $D(ij) = b(i)b(j)/b(V)$. Clearly, the cheapest network that supports this specific demand matrix via fractional routing is a lower bound on the cost of a network that supports all matrices in $\mathcal{D}$. The cheapest network to support D is obtained by routing the demand $D(ij)$ along a shortest path between i and j. Focus on a specific i. Then the demands $D(ij)$, $j \neq i$ are routed, without loss of generality, along T_i. Thus the total cost of routing demands from i is $\frac{b(i)}{b(V)} \sum_{j \neq i} b(j)c(ij)$. Note that this is simply $\frac{b(i)}{b(V)} c(T_i)$. Thus the total cost of routing D is $\frac{1}{2} \sum_i \frac{b(i)}{b(V)} c(T_i)$ where the factor $1/2$ takes into account the fact that the demand ij appears both in the cost for i and for j. This immediately gives us the following inequalities.

$$OPT(SPR) \geq OPT(FR) \geq \frac{1}{2} \sum_i \frac{b(i)}{b(V)} c(T_i) \geq \frac{1}{2} \min_i c(T_i) = \frac{1}{2} OPT(TR).$$

A bit more care in the above analysis shows that $OPT(TR) \leq 2(1 - 1/k)OPT(SPR)$ where k is the number of terminals (nodes with positive $b(i)$ values).

Now we consider the MPR model. The following theorem is a special case of Theorem 3.3.

Theorem 4.2 ([24, 38]) *$OPT(MPR)$ can be computed in polynomial time.*

Proposition 4.3 *There are instances in which $OPT(FR) < OPT(MPR)$.*

A simple example from [38] is the following. G is a triangle (K_3) withe each edge of cost 1 and $b(i) = 1$ for each i. $OPT(FR) = 3/2$ and $u(e) = 1/2$ on each edge certifies this. However $OPT(MPR) = 2$; two of the three edges need a capacity of 1 each.

This leads us to the main open problems.

Open Problems:

- Can $OPT(SPR)$ be computed in polynomial time? In [39] it is conjectured that $OPT(SPR) = OPT(TR)$ for all instances. If this were true then Theorem 4.1 would yield a polynomial time algorithm for $OPT(SPR)$.
- Is $OPT(MPR) = OPT(TR)$ for all instances? This is conjectured to be true in [38]. This would imply that $OPT(MPR) = OPT(SPR) = OPT(TTR) = OPT(TR)$ and also would yield a polynomial time algorithm for $OPT(SPR)$.

- Can $OPT(FR)$ be computed in polynomial time?

In some important progress, Hurkens et al. [38] showed the following.

Theorem 4.4 ([38]) *$OPT(MPR) = OPT(TR)$ when G is a ring network. Moreover this is true for the following special cases.*

- *G is a complete graph and $c(e) = 1$ for all $e \in E$.*
- *There is a node $v \in V$ such that $b(v) \geq b(V \setminus \{v\})$.*
- *$|V| \leq 4$.*

In fact, Hurkens et al. [38] are able to prove the conjecture for some larger classes of instances obtained from the above special cases using some restricted 1-sum operations on the graphs. See [38] for more details. Grandoni et al. [23] obtained a simpler combinatorial proof for the ring.

4.2 Asymmetric Case

Now we consider the asymmetric case. We use the same notation as in the symmetric case in referring to the optimum solution values for the different routing models. In contrast to the symmetric case, the asymmetric problems are NP-hard even in undirected graphs. Asymmetric instances can be used to model the single-source symmetric hose model problem which leads to the known hardness results. The reduction is as follows. First, a symmetric single-source hose problem has a root vertex r and the valid demand matrices are given by the following linear program.

$$\begin{aligned} \sum_j D(rj) &\leq b(r) \\ D(rj) &\leq b(j) \qquad j \in V \setminus \{r\} \\ D(ij) &= 0 \qquad i \neq r. \end{aligned}$$

One can easily model the above as a special case of the the asymmetric case as follows: set $b_{in}(r) = b(r), b_{out}(r) = 0$ and for $j \in V \setminus \{r\}$ set $b_{in}(j) = 0, b_{out}(j) = b(j)$.

The known results on the hardness of the asymmetric problems are given below.

- $OPT(SPR)$ is NP-hard and also max-SNP hard. In fact, an α approximation for $OPT(SPR)$ implies an α approximation to Steiner tree problem. These results were shown in [33] and the reduction from the Steiner tree problem is discussed in Section 3 after Theorem 3.4. The same reduction applies in an identical fashion to $OPT(TR)$.
- $OPT(FR)$ is co-NP hard. This follows from the reduction used to prove Theorem 3.2 in [18]. For directed graphs this was shown earlier in [33].
- $OPT(MPR)$ can be computed in polynomial time (see Theorem 3.3).

Several approximation algorithms have been devised for the asymmetric problem. In [33], a new network design problem called connected facility location was introduced (see also [30, 41, 53]) to approximate $OPT(TR)$ and the following theorem was shown.

Theorem 4.5 ([33]) *There is a 10-approximation for $OPT(TR)$.*

The first constant factor approximation for $OPT(SPR)$ was obtained by Gupta, Kumar and Roughgarden [32].

	Routing Model			
	FR	SPR	TR	MPR
Polyhedral	coNP-hard $O(\log n)$	APX-hard $O(\log n)$	APX-hard $O(\log n)$	Poly-time 1
Sym VPN	? 2	? 2	Poly-time 1	Poly-time 1
Asym VPN	APX-hard 3.55	APX-hard 3.55	APX-hard 4.74	Poly-time 1

Table 1: Summary of known results on the complexity and approximability of robust network design in undirected graphs with linear costs.

Theorem 4.6 ([32]) *There is a randomized polynomial time algorithm that computes a tree solution T such that $c(T) \leq 5.33OPT(FR)$. Therefore $OPT(FR) \leq OPT(SPR) \leq OPT(TR) \leq 5.33OPT(FR)$.*

Building upon some of the ideas in [32], Eisenbrand and Grandoni [21] obtained a randomized 4.74-approximation, again via a tree solution. The current best ratio is due to Eisenbrand et al. [22] who introduced a new lower bound and analysis technique.

Theorem 4.7 ([22]) *There is a 3.55-approximation algorithm for $OPT(SPR)$. The solution produced by this algorithm is not necessarily a tree solution.*

Open Problems:

- What are the approximability thresholds for $OPT(SPR)$, $OPT(TR)$ and $OPT(FR)$?
- What is the worst case ratio between $OPT(FR)$ and $OPT(MPR)$?

We observe that the currently known algorithms for $OPT(SPR)$ that achieve small constant factor approximations are inherently randomized [32, 21, 22]. Is there a deterministic approximation algorithm that achieves a comparable ratio? Note that a deterministic $O(1)$ approximation can be obtained by the fact that that $OPT(SPR) \leq 5.33OPT(TR)$ (see Theorem 4.6 above) and the deterministic 10-approximation for $OPT(TR)$ [33].

Balanced Thresholds: A special case of the asymmetric case is obtained by adding the constraint that

$$\sum_i b_{in}(i) = \sum_i b_{out}(i)$$

which is natural in some settings. For this case Fingerhut et al. [26] gave a 3-approximation for $OPT(SPR)$. Italiano et al. [39] showed that $OPT(TR)$ can be computed in polynomial time and also obtained a 3-approximation for $OPT(SPR)$. In [22] the approximation is improved to 2. These results can be thought of as generalizing those in [33] for the symmetric case. The complexity of $OPT(SPR)$ remains open as in the symmetric case.

5 Oblivious Routing

Oblivious routing ideas have roots in the area of distributed routing algorithms. In networks it is advantageous to have easy to specify routes between pairs of nodes. A first question is whether one can specify a single path $P(i, j)$ for each pair of nodes (i, j) that is good for *all* demand matrices (one can think of $\mathcal{D}$ as the set $\mathcal{D}_u$). The measure of goodness here is edge-congestion as defined earlier. For example, such a path collection exists when the graph is a tree since there is only one path between each pair of nodes. However, it is not hard to show that even on simple non-trivial graphs, such as a grid, no such deterministic scheme can obtain a congestion bound better than $\Omega(\sqrt{n})$. Thus a natural relaxation is to consider a *randomized* oblivious routing scheme. More formally, in such a scheme, for each pair of nodes (i, j) there is a probability distribution on $\mu(i, j)$ on the set of all paths between i and j and for each demand matrix the routing is chosen by picking for each (i, j) a path according to $\mu(i, j)$. Note that $\mu(i, j)$ does not vary with the demand matrix. Typically, the performance of an oblivious routing (also referred to as the competitive ratio), is defined as a relative measure; a worst case ratio, over all demand matrices $D \in \mathcal{D}$, between the congestion achieved by the oblivious routing for D and the best congestion achievable for D. As we mentioned earlier, we use an absolute measure of performance by restricting attention to $\mathcal{D} \subseteq \mathcal{D}_u$. Hence we can define the performance of an oblivious routing as an absolute congestion bound; in other words what is worst case congestion over all $D \in \mathcal{D}$ for a given oblivious routing scheme. Valiant and Brebner [56] showed a randomized oblivious routing scheme for the hyper-cube with congestion $O(\log n)$. It is also known that even in planar graphs (in fact on grids) there is a lower bound of $\Omega(\log n)$ on the congestion achievable by a randomized oblivious scheme [47, 9]. In a breakthrough paper [52] Räcke showed the following.

Theorem 5.1 ([52]) *There is a randomized oblivious routing scheme in undirected graphs with congestion $O(\log^3 n)$.*

Räcke proved Theorem 5.1 by constructing a hierarchical decomposition of a given undirected graph; this decomposition results in a tree representation of the graph and routing on the tree serves as an approximate proxy for routing in the graph. This representation has already found several important applications in offline and online algorithms [52, 16, 48, 2, 28]. The bound on the congestion has been improved subsequently by Harrelson, Hildrum and Rao [37] and a polynomial time algorithm to construct the hierarchical decomposition was also obtained [37, 12].

Theorem 5.2 ([37]) *There is a randomized oblivious routing scheme in undirected graphs with congestion $O(\log^2 n \log\log n)$. In planar graphs the bound improves to $O(\log n \log\log n)$.*

We now connect the above results and some others to the framework of robust network design and routing. A randomized oblivious routing is essentially the same as an MPR routing. Given a demand class $\mathcal{D}$ and a capacitated graph G let $\lambda_o^*(\mathcal{D}, G)$ denote the minimum congestion achievable for $\mathcal{D}$ in G under the MPR model of routing. The results of [52, 37] show that $\lambda_o^*(\mathcal{D}_u, G) = O(\log^2 n \log\log n)$ for undirected graphs. Note that $\lambda_o^*(\mathcal{D}, G) \leq \lambda_o^*(\mathcal{D}_u, G)$ for any $\mathcal{D} \subseteq \mathcal{D}_u$. One can ask whether $\lambda_o^*(\mathcal{D}, G)$, and a corresponding routing that achieves it, can be computed in polynomial time. The answer turns out to be yes even for *directed* graphs. Although [7] state the theorem only for $\mathcal{D}_u$, the proof applies to any $P(\mathcal{D})$ that has a polynomial time separation oracle.

Theorem 5.3 ([7]) *Given a directed or undirected graph G and a polyhedral set of demand matrices $P(\mathcal{D})$ with a polynomial time separation oracle, there is a polynomial time algorithm to compute $\lambda_o^*(P(\mathcal{D}), G)$ and a corresponding routing that achieves this bound.*

First we note that Theorem 3.3 can be obtained from the above theorem; the separation oracle is to check whether a given capacity vector $u \in \mathcal{U}(\mathcal{D})$ under the MPR model. This is the same as checking whether $\lambda_o^*(P(\mathcal{D}), G) \leq 1$ with G having edge capacities u.

We sketch the proof of Theorem 5.3 which casts the problem as a large linear program. This can be solved by the ellipsoid method by giving a polynomial time separation oracle that is based on the separation oracle for $P(\mathcal{D})$. Note that a MPR routing specifies for each pair (i, j) a probability distribution over paths from i to j. This can alternatively be thought of as a unit-flow from i to j since such a flow can be decomposed into flow paths. The advantage of thinking in terms of flow instead of paths is that one can use a polynomial (in n) number of variables. It is convenient here to assume that G is directed. Thus, an MPR scheme can be thought of as specifying variables $f_{(i,j)}(e)$ for each pair $(i, j) \in V \times V$ and arc $e \in E$. For a node i let $\delta^+(i)$ denote the set of arcs leaving i and let $\delta^-(i)$ denote the arcs entering i. We cast $\lambda^*_o(\mathcal{D}, G)$ as the optimum of the following linear program:

$$
\begin{aligned}
\text{LP-OR:} \quad & \min \lambda \\
& \text{subject to} \\
\sum_{e\in\delta^+(i)} f_{(i,j)}(e) - \sum_{e\in\delta^-(i)} f_{(i,j)}(e) &= 1 \qquad \forall i \in V \\
\sum_{e\in\delta^+(v)} f_{(i,j)}(e) - \sum_{e\in\delta^-(v)} f_{(i,j)}(e) &= 0 \qquad v \neq i, v \neq j \\
\sum_{1\leq i\leq j\leq n} D(i,j) f_{(i,j)}(e) &\leq \lambda \cdot u(e) \qquad e \in E, D \in \mathcal{D} \\
f_{(i,j)}(e) &\geq 0 \qquad 1 \leq i \leq j \leq n, e \in E.
\end{aligned}
$$

The first set of constraints state that the total (i, j) flow is 1. The second set of constraints are the standard flow conservation constraints. The third set of constraints state that for any D, the congestion of the routing f is at most λ. Note that there is a separate constraint for each $D \in \mathcal{D}$ and hence if $\mathcal{D}$ is not a discrete set, there are potentially an infinite number of constraints. First, let us consider the case that $\mathcal{D}$ is a discrete set given explicitly, in which case the above is a linear program whose size is polynomial in the input size. It is easy to see that the optimum solution value for LP-OR is the same as $\lambda^*_o(\mathcal{D}, G)$.

Next we consider the case that $\mathcal{D}$ is polyhedral given by $P(\mathcal{D})$. Observe that $P(\mathcal{D})$ is bounded since we assume that $\mathcal{D} \subseteq \mathcal{D}_u$. Let $\mathcal{D}'$ be the finite set of demand matrices that correspond to the extreme points of $P(\mathcal{D})$. Then, instead of writing the constraints as one for each $D \in \mathcal{D}$ we can write the constraints for only $D \in \mathcal{D}'$. Thus we do obtain a linear program for polyhedral $\mathcal{D}$. It only remains to show how the above linear program can be solved since $|\mathcal{D}'|$ can be exponential in n. We observe that the separation oracle can be constructed for LP-OR in a relatively easy fashion from one for $P(\mathcal{D})$; given values for λ and $f_{(i,j)}(e)$ we can easily check whether any constraint in the first two sets of flow constraints is violated. We consider the remaining constraints. Consider one particular edge e. To check whether all $D \in \mathcal{D}$ satisfy the congestion bound on e for the given λ and f we solve the following linear program.

$$
\begin{aligned}
& \max \sum_{(i,j)} f_{(i,j)}(e) \cdot D(i,j) \\
& \text{subject to} \\
& \qquad D \in P(\mathcal{D}).
\end{aligned}
$$

Let $\gamma(e)$ be the optimum solution for the above linear program. This linear program can be solved using the separation oracle for $P(\mathcal{D})$, again using the equivalence of separation and optimization. If $\gamma(e) > \lambda u(e)$, then a demand matrix D_e that achieves $\gamma(e)$ identifies a violated constraint for LP-OR. We solve a separate linear program for each $e \in E$ and if $\gamma(e) \leq \lambda(e)$ for all e then $\mathcal{D}$ satisfies the desired constraints for the given f and λ. Thus we have a polynomial time separation oracle for LP-OR assuming one for $P(\mathcal{D})$.

The above proof requires one to solve LP-OR via the ellipsoid method. In [5] it is shown that there exists a compact formulation to solve $\lambda_o^*(\mathcal{D}, G)$ for certain classes of $\mathcal{D}$, in particular for $\mathcal{D}_u$. In [44, 3, 38] a compact formulation is given for the hose model. In fact, as observed by several people, the proof applies for all $P(\mathcal{D})$ described compactly. A published proof appears in [3]. The high level idea is not difficult and we briefly describe it. Assume without loss of generality that $P(\mathcal{D})$ is described by the system of inequalities $A \cdot D \leq b, D \geq 0$ for some matrix A. In the separation oracle we described above for LP-OR the main task is to check that $\gamma(e) \leq \lambda u(e)$ for all $e \in E$. Since $\gamma(e)$ is the value of an optimum solution of a linear program, by duality it is also equal to the optimum solution to the dual linear program which is given below. Y_e is the set of dual variables and $f(e)$ is the $n^2 \times 1$ column vector of values $f_{(i,j)}(e)$, $(i, j) \in V \times V$.

$$\begin{aligned} &\min b^T \cdot Y_e \\ &\text{subject to} \\ &\quad Y_e^T \cdot A \;\geq\; f(e) \\ &\quad\quad Y_e \;\geq\; 0. \end{aligned}$$

Thus the constraints $\gamma(e) \leq \lambda u(e)$ for all e can be equivalently written as the system below.

$$\begin{array}{rcll} b^T \cdot Y_e & \leq & \lambda \cdot u(e) & e \in E \\ Y_e^T \cdot A & \geq & f(e) & e \in E \\ Y_e & \geq & 0 & e \in E. \end{array}$$

Note that the above system is linear in the variables $f_{(i,j)}(e)$! Hence we can equivalently write LP-OR as the following when $P(\mathcal{D}) = \{D \mid A \cdot D \leq b, D \geq 0\}$ is given as an explicit system.

$$\begin{array}{rcll} \text{LP-OR-compact:} \quad \min \lambda & & & \\ \text{subject to} & & & \\ \sum_{e \in \delta^+(i)} f_{(i,j)}(e) - \sum_{e \in \delta^-(i)} f_{(i,j)}(e) & = & 1 & \forall i \in V \\ \sum_{e \in \delta^+(v)} f_{(i,j)}(e) - \sum_{e \in \delta^-(v)} f_{(i,j)}(e) & = & 0 & v \neq i, v \neq j \\ b^T \cdot Y_e & \leq & \lambda \cdot u(e) & e \in E \\ Y_e^T \cdot A & \geq & f(e) & e \in E \\ f_{(i,j)}(e) & \geq & 0 & 1 \leq i \leq j \leq n, e \in E \\ Y_e & \geq & 0 & e \in E. \end{array}$$

Note that the above program has variables $f_{(i,j)}(e)$ as well as those from $Y_e, e \in E$. If the matrix A is of size polynomial in n then so too is LP-OR-compact. so Although $\lambda_o^*(\mathcal{D}, G)$ can be computed in polynomial time for directed graphs, there exist simple directed graphs for which $\lambda_o^*(\mathcal{D}, G) = \Omega(\sqrt{n})$ [7]. Further, it is also known that $\lambda_o^*(\mathcal{D}, G) = \Omega(\sqrt{n})$ for undirected graphs in the node-capacitated setting [35]. To overcome these negative results, Hajiaghayi et al. [36] propose a different but related model for which they are able to obtain an $O(\log^2 n)$ upper bound even for directed graphs.

Open Problem: The major open problem is to prove tight bounds on $\lambda_o^*(\mathcal{D}_u, G)$ for undirected graphs. The lower bound is $\Omega(\log n)$ even in planar graphs and the upper bound is $O(\log^2 n \log \log n)$ and $O(\log n \log \log n)$ in general and planar graphs respectively.

6 Discussion and Conclusions

The hose model originated from the networking world with practical motivations [26, 20]. Nevertheless, the associated optimization problems have attracted substantial attention in the algorithms and mathematical programming communities. Although the robust network design problem (in particular, the VPN design problem) and randomized oblivious routing to minimize congestion were considered separately, close connections between them have been noticed and exploited. Further, a polyhedral view makes some of the connections more transparent. Gupta et al. [34] consider oblivious network design; their results show the existence of competitive routing strategies to minimize various cost functions; the routing is oblivious to the cost function as long as it belongs to a broad class that satisfies some reasonable properties.

We point out another useful perspective on the hose model. Consider the symmetric model with $b(i)$ as the bound for node $i \in V$, and let $\mathcal{D}$ be the class of matrices induced by these bounds. A network that supports all the matrices in $\mathcal{D}$ behaves like a *switch* with respect to the terminals (those with $b(i) > 0$). The simplest kind of switch is a single node connected to the terminals. In essence, with linear costs, Theorem 4.1 shows that a near-optimal solution is to choose a single node (the hub) as the network switch and route the terminals to this node in the cheapest possible way. Thus the design problem allows us to choose a very simple switch. In other routing settings, a fundamental question is when a capacitated graph can act as a switch for a given set of terminals. As we saw in the proof of Theorem 4.1, the demand matrix D, with $D(ij) = b(i)b(j)/b(V)$ for each pair ij, plays an important role — we refer to this as the universal matrix since it captures the essence of the class $\mathcal{D}$. This is also well known as a product multi-commodity flow instance with weights b and plays an important role in flows and cuts [45]. In particular, the flow-cut gap result shown in [45], and the above universal matrix, are crucial ingredients in the oblivious routing schem of Räcke [52, 37]. Finally, the following question has important implications to disjoint paths problems as shown in [17]: does a capacitated graph that acts as a switch for a set of terminals in terms of fractional routing, also act as an approximate switch for the same set of terminals in terms of integral routing? See [17] for a formal version of this question.

We close by discussing a line of work that explores new network architectures that are motivated by, and connected to, the problems and ideas surveyed in this article. Traditional circuit switched networks provide dedicated high capacity connections between given end points. These connections are typically provisioned upfront and are based on estimated traffic demands. They have the advantage of providing quality of service guarantees and fault tolerance. On the other hand, packet networks must be more flexible to changes in traffic since they dynamically adapt their routes and transmission rates; the flip side is the cost and complexity of switching at routers (as capacities increase dramatically), and the difficulty of providing quality of service guarantees and fault tolerance. A compromise of sorts exists, where packet networks are more prevalent at the edge of a large network such as the internet, while high capacity back-bone networks are, to a large extent, circuit based. Statistical multiplexing allows reliable estimates for end-to-end demand in back-bone networks. In the past decade, new technologies and applications have been driving an ever-increasing request for bandwidth at the edge of the network, with an accompanying increase in the need for quality of service and reliability. Some of the motivation behind the introduction of the hose model in [26, 20], and subsequent related work, is the desire to design networks that can combine the best of both worlds; provide robustness to changing traffic while having the benefits of fixed routes and large capacities. Of course, designing networks to handle all demands in a large class, such as those considered in the hose model(s), is likely to be more expensive. Further, the hub-routing algorithm suggested by Theorem 4.1 has a single point of failure at the hub. Some work [43, 57, 49, 50] suggests the use of randomized load balancing (RLB), a scheme inspired by the ideas of Valiant [55]; one can view this as a fractional version of hub-routing. In [51], based on the fact that RLB can be viewed as a convex combination of hub routings, they propose selective RLB that uses $k \in [1..n]$ hubs. Choosing k appropriately obtains the cost advantages of hub routing and the resilience of RLB. Empirical work [51] also measures the *robustness premium*, the extra cost in network resources needed to support all hose matrices, as opposed to a single target demand matrix. These efforts are an ongoing debate in the evolution of network architectures. We also mention the empirical work in [5] on the effectiveness of oblivious routing mechanisms in providing robustness to changing traffic demands. The theoretical models, algorithms and insights are useful in providing ways to evaluate and study alternative architectures, mechanisms, and policies in a systematic way.

Acknowledgments: The author is grateful to his collaborators on [18] for teaching him much of what he knows about this topic. Several conversations with Anupam Gupta, Gianpaolo Oriolo and Bruce Shepherd have also been illuminating. We thank the aforementioned, Navin Goyal, Samir Khuller and Neil Olver for several comments and corrections on a draft of this article. Thanks to Samir for making me write this.

References

[1] R. Ahuja, T. Magnanti and J. Orlin. Network Flows: Theory, Algorithms and Applications. Prentice Hall, 1993.

[2] N. Alon, B. Awerbuch, Y. Azar, N. Buchbinder, and J. Naor. A general approach to online network optimization problems. *ACM Transactions on Algorithms*, 2(4):640–660, 2006.

[3] A. Altin, E. Amaldi, P. Belotti and M.C. Pinar. Provisioning Virtual Private Networks under Traffic Uncertainty. *Networks*, 49(1):100–115, 2007. Preliminary version in *Proc. of INOC*, 250–257, 2005.

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

[5] D. Applegate and E. Cohen. Making Intra-Domain Routing Robust to Changing and Uncertain Traffic Demands: Understanding Fundamental Tradeoffs. *Proc. of SIGCOMM*, 2003.

[6] S. Arora, S. Rao and U. Vazirani. Expander Flows, Geometric Embeddings and Graph Partitioning. *Proc. of ACM STOC*, 2004.

[7] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Räcke. Optimal oblivious routing in polynomial time. *J. of Computer System Sciences (JCSS)*, 69(3):383–394, 2004. Preliminary version in *Proc. of ACM STOC*, 2003.

[8] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. *Proc. of IEEE FOCS*, 184–193, 1996.

[9] Y. Bartal and S. Leonardi. On-line routing in all optical networks. *Theor. Comp. Sci.*, Vol 221, 19–39, 1999.

[10] W. Ben-Ameur and H. Kerivin. Routing of Uncertain Demands. *Optimization and Engineering*, 3:283–313, 2005.

[11] W. Ben-Ameur and H. Kerivin. New economical virtual private networks. *Communications of the ACM*, Vol 46, 69–73, June 2003.

[12] M. Bienkowski, M. Korzeniowski, and H. Räcke. A Practical Algorithm for Constructing Oblivious Routing Schemes. *Proc. of SPAA*, 2003.

[13] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin. Approximating a Finite Metric by a Small Number of Tree Metrics. *Proc. of IEEE FOCS*, 1998.

[14] C. Chekuri, M. Hajiaghayi, G. Kortsarz, and M. Salavatipour. Approximation Algorithms for Non-uniform Buy-at-Bulk Network Design Problems. *Proc. of IEEE FOCS*, 2006.

[15] C. Chekuri, M. Hajiaghayi, G. Kortsarz, and M. Salavatipour. Approximation Algorithms for Node-Weighted Buy-at-Bulk Network Design. *Proc. of ACM-SIAM SODA*, 2007.

[16] C. Chekuri, S. Khanna, and F. B. Shepherd. The All-or-Nothing Multicommodity Flow Problem. *Proc. of ACM STOC*, 2004.

[17] C. Chekuri, S. Khanna, and F. B. Shepherd. Multicommodity Flows, Well-Linked Terminals and Routing Problems. *Proc. of ACM STOC*, 2005.

[18] C. Chekuri, G. Oriolo, M. G. Scutella and F. B. Shepherd. Hardness of Robust Network Design. *Networks*, 50(1):50–54, 2007. Preliminary version appeared in *Proc. of INOC*, 2005.

[19] Y. Dodis and S. Khanna. Design Networks with Bounded Pairwise Distance. *Proc. of ACM STOC*, 750–759, 1999.

[20] N.G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K.K. Ramakrishnan, and J. E. van der Merwe. A Flexible Model for Resource Management in Virtual Private Networks. *Proc. of SIGCOMM*, 1999.

[21] F. Eisenbrand and F. Grandoni. An improved approximation algorithm for virtual private network design. In *Proc. of ACM-SIAM SODA*, 928–932, 2005.

[22] F. Eisenbrand, F. Grandoni, G. Oriolo and M. Skutella. New Approaches for Virtual Private Network Design. *SIAM J. on Computing*, 2007. Preliminary version in *Proc. of ICALP*, LNCS, 1151–1162, 2005.

[23] F. Grandoni, V. Kaibel, G. Oriolo and M. Skutella. Unpublished manuscript, 2005.

[24] T. Erlebach and M. Ruegg. Optimal Bandwidth Reservation in Hose-Model VPNs with Multi-Path Routing. *Proc. of INFOCOM*, March 2004.

[25] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *JCSS*, 69:485–497, 2004. Preliminary version in *Proc. of ACM STOC*, 2003.

[26] J. Andrew Fingerhut, S. Suri and J. Turner. Designing Least-Cost Nonblocking Broadband Networks. *Journal of Algorithms*, 24(2):287–309, 1997.

[27] M. R. Garey, D. S. Johnson, and L. Stockmeyer Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[28] D. Golovin, V. Nagarajan and M. Singh. Approximating the k-Multicut Problem. *Proc. of ACM-SIAM SODA*, 621–630, 2006.

[29] M. Grotschel, L. Lovasz, and A. Schrijver. Geometric Algorithms and Combinatorial Optimization. Springer, 1988.

[30] S. Guha, A. Meyerson and K. Munagala. Hierarchical Placement and Network Design Problems. *Proc. of IEEE FOCS*, 2000.

[31] A. Gupta. Personal communication, 2004.

[32] A. Gupta, A. Kumar, T. Roughgarden. Simpler and better approximation algorithms for network design. *Proc. of ACM STOC*, 365-372, 2003.

[33] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a Virtual Private Network: A Network Design problem for Multicommodity flows. *Proc. of ACM STOC*, 389–398, 2001.

[34] A. Gupta, M. Hajiaghayi and H. Räcke. Oblivious Network Design. *Proc. of ACM-SIAM SODA*, 970–979, 2006.

[35] M. Hajiaghayi, R. Kleinberg, T. Leighton and H. Raecke. Oblivious Routing on Node-Capacitated and Directed Graphs. *Proc. ACM-SIAM SODA*, 782–790, 2005.

[36] M. Hajiahgayi, J. Kim, T. Leighton and H. Raecke. Oblivious routing in directed graphs with random demands. *Proc. of ACM STOC*, 193–201, 2005.

[37] C. Harrelson, K. Hildrum, and S. Rao. A polynomial-time tree decomposition to minimize congestion. *Proc. of SPAA*, 2003.

[38] C. Hurkens, J. Keijsper, and L. Stougie. Virtual private network design: A proof of the tree routing conjecture on ring networks. *SIAM J. on Disc. Math.*, 2007. Preliminary version in *Proc. of IPCO XI*, 2005.

[39] G. Italiano, S. Leonardi and G. Oriolo. Design of Trees in the Hose Model: the Balanced Case. *OR Letters*, 34(6):601–606, 2006. A preliminary version appeared in *Proc. of ARACNE*, 65–76, 2002.

[40] O. E. Karasan, M. Pinar, H. Yaman. Robust DWDM Routing and Provisioning under Polyhedral Demand Uncertainty. *Proc. of INOC*, 242–249, 2005.

[41] D. Karger and M. Minkoff. Building Steiner Trees with Incomplete Global Knowledge. *Proc. of IEEE FOCS*, 2000.

[42] S. Khot. On the power of unique 2-prover 1-round games. *Proc. of ACM STOC*, 767–775, 2002.

[43] M. Kodialam, T. Lakshman and S. Sengupta. Efficient and Robust Routing of Highly Variable Traffic. *HotNets III*, 2004.

[44] M. Kodialam, T. Lakshman and S. Sengupta. Maximum Throughput Routing of Traffic in the Hose Model *Proc. of IEEE INFOCOM*, 2006.

[45] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *JACM*, 46(6):787–832, 1999. Preliminary version appeared in *Proc. of IEEE FOCS*, 1988.

[46] C. Lemaréchal, A. Ouorou and G. Petrou. Robust network design in telecommunications. *Proc. of INOC*, 258–264, 2005.

[47] B. Maggs, F. Meyer auf der Heide, B. Vöcking, M. Westerman. Exploiting locality for networks of limited bandwidth. *Proc. of IEEE FOCS*, 284–293, 1997.

[48] B. Maggs, G. Miller, O. Parekh, R. Ravi and S. Woo. Finding effective support-tree preconditioners. *Proc. of ACM SPAA*, 176–185, 2005.

[49] H. Nagesh, V. Poosala, S. Sengupta, M. Alicherry and V. Kumar. NetSwitch: Load-balanced Data-over-Optical Architecture for Mesh Networks. Lucent Technical Memorandum ITD-04-45867F, 2004.

[50] H. Nagesh, V. Poosala, V. Kumar, P. Winzer and M. Zirngibl. Load balanced architecture for dynamic traffic. *Optical Fiber Communication Conf. (OFC)*, 2005.

[51] F.B. Shepherd and P. Winzer. Selective Randomized Load Balancing and Mesh Networks with Changing Demands. *Journal of Optical Networking*, 5(5):320–339, 2006.

[52] H. Räcke. Minimizing Congestion in General Networks. *Proc. of IEEE FOCS*, 43–52, 2002.

[53] R. Ravi and F. Salman. Approximation Algorithms for the Traveling Purchaser Problem and its Variants in Network Design. *Proc. of ESA*, 29–40, 1999.

[54] A. Schrijver. Combinatorial optimization: Polyhedra and efficiency. Springer, 2003.

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

[56] L. Valiant and G. Brebner. Universal schemes for parallel communication. *Proc. of ACM STOC*, 263–277, 1981.

[57] R. Zhang-Shen and N. McKeown. Designing a Predictable Internet Backbone Network. HotNets III, San Diego, 2004.