Robust Network Design

Neil Kelvin Olver

Department of Mathematics and Statistics,
McGill University, Montréal
Québec, Canada

October, 2010

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements of the degree of Doctor of Philosophy.

Abstract

Robust network design takes the very successful framework of robust optimization and applies it to the area of network design, motivated by applications in communication networks. The main premise is that demands across the network are not fixed, but are variable or uncertain. However, they are known to fall within a prescribed *uncertainty set*. Our solution must have sufficient capacity to route any demand in this set; moreover, the routing must be *oblivious*, meaning it must be fixed up front, and not depend on the particular choice of demand from within the uncertainty set.

A particular choice of uncertainty set within this framework yields the "hose model", which has received particular attention due to applications to virtual private networks. A 2-approximation was known for the problem, using a solution template in the form of a tree. It was conjectured that this tree solution is actually always optimal; this became known as the *VPN Conjecture*. As one of the central results of this thesis, we prove this conjecture in full generality. In addition, we demonstrate a counterexample to a stronger multipath (fractional routing) version of the conjecture which had also been proposed.

We initiate a study of the robust network design problem more generally, with a focus on approximability. In the general model, where the uncertainty set is given by an arbitrary separable polyhedron, we give a strong inapproximability result. We then consider a new and natural model generalizing the symmetric hose model, based on demands routable on a given tree, and provide a constant factor approximation algorithm.

Lastly, we compare oblivious routing with the much more flexible (but also less practical) *dynamic* routing scheme where the routing may vary depending on the demand pattern. We show that in the worst case, the cost of an optimal oblivious routing solution can be much more expensive than the dynamic optimum, by up to a logarithmic factor.

Résumé

Motivé par les applications concernantes les réseaux de communication, le dessein des réseaux robustes applique les méthodes très réusies provenant de l'optimisation robuste. La prémisse principale est que les demandes sur le réseau ne sont pas fixes, mais variables ou incertaines. Cependant, nous savons qu'elles sont tirées d'un *ensemble d'incertitude* prescrit. Il faut que la solution ait une capacité suffisante pour pouvoir router toute demande appartenant à cet ensemble. En outre, il faut que le routage soit *oublieux*, ce qui signifie qu'il peut être fixé à l'avance, et ne dépends pas du choix particulier de la demande appartenant de l'ensemble d'incertitude.

Dans ce cadre, il existe un choix particulier d'ensemble d'incertitude qui mène au « modèle de tuyau ». Ce modèle a reçu une attention particulière à causede ses applications aux réseaux privés virtuels. On connaissait un 2-rapprochement utilisant une solution en forme d'arbre. La *Conjecture de* VPN énonce que cette solution en forme d'arbre est toujours optimale. L'un des résultats principaux de cette thèse démontre cette conjecture en toute généralité. En outre, nous donnons un contre-exemple à une version plus forte de la conjecture concernant les chemins multiples (le routage étant fractionnel) qui avait également été proposée.

Nous initions l'étude du problème de la conception de réseaux robustes dans une plus grande généralité, en insistant sur l'approximabilité. Dans le modèle général, où l'ensemble d'incertitude est un polyèdre séparable arbitraire, nous donnons un résultat fort d'inapproximabilité. Nous considérons ensuite un nouveau modèle naturel généralisant le modèle de tuyau symétrique, qui est basé sur des demandes qui peuvent être routées sur un arbre donné, et nous fournissons un algorithme ayant un facteur de rapprochement constant.

Finalement, nous comparons le routage oublieux avec le schéma beaucoup plus flexible (mais moins pratique) du routage dynamique, où le routage peut varier en fonction de la structure des demandes. Nous montrons que dans le pire des cas, une solution optimale de routage ou-

blieux peut être beaucoup plus chère que l'optimum dynamique, jusqu'à un facteur logarithmique.

Dedicated to my parents, Anne and Keith.

Acknowledgements

First and foremost, I would like to thank my supervisors, Bruce Shepherd and Adrian Vetta, for their wonderful wisdom, advice, and encouragement.

Working with Bruce has been a real joy for me, and I feel very lucky to have had an advisor I could juggle ideas with so comfortably. After every meeting, there was always at least one promising avenue to explore, and Bruce's ceaseless optimism has helped me through some roadblocks that seemed insurmountable. I still remember the inspiring talk Bruce gave when he first came to McGill, where he mentioned the VPN Conjecture and piqued my curiosity. This thesis has benefited from his guidance at every step, and I am very thankful for it. I've also enjoyed our many good times outside of work, and benefited from his great taste in wine.

Adrian was the one who introduced me to the world of discrete mathematics and algorithms, and I would not be where I am now without his early guidance; it was because of him that I decided to stay at McGill for my doctorate. While research-wise our paths have diverged somewhat in the course of my PhD, I have still benefited greatly from his advice on many things over the years. He is also a voice of reason when I decide to get stressed about unimportant things.

I have greatly enjoyed collaborating with Navin Goyal: Navin's arrival sparked the serious push on the VPN Conjecture. I also really enjoyed my experience working with him on other projects in India towards the end of my studies—it was an amazing opportunity.

Grazie mille to Gianpaolo Oriolo, for some really enjoyable thoughts and discussions on a range of topics, and an unforgettable visit to bella Roma. I have very fond memories of thinking about mathematics while wandering the beautiful streets, and sampling his delicious Italian cooking (also, some terrifying scooter experiences).

Thanks to Chandra Chekuri, for feedback, advice, and a useful discussion as I was finishing this thesis, and Sanjeev Khanna, for two extremely enjoyable visits to U Penn. Vašek Chvátal's weekly meetings were always a refreshing change of perspective on how to approach research.

I am very grateful to the authors of [71], Fabrizio Grandoni, Volker Kaibel, Gianpaolo Oriolo and Martin Skutella, for providing a preprint of their paper before it was published. Their ideas substantially altered our approach to the VPN problem. I am also grateful to András Sebő for communicating his shortened proof of Theorem 3.16, reproduced in §3.3.3. The work in Chapter 6 was sparked by a question of Moses Charikar to Navin.

My studies were supported through a Québec Merit Scholarship for Foreign Students, as well as a Schulich fellowship in my final year. I am very grateful to have received this generous support.

The administrative staff in the department have been unfailingly dependable; I especially thank Angela White, Raffaella Bruno and Spencer Keys-Schatia for all their assistance. Nicolas Sonnerat and Christina Goldschmidt kindly helped with the French translation of the abstract (Nic also baked some stupendous cakes).

I have enjoyed the company of some great colleagues and friends at McGill over the years. My officemates Dave and Joel provided an ear, a different perspective, or just light relief; the office was not the same once they left. Mike was the absolute perfect roommate. Too many friends made in Montreal left before me—now it is my turn, but I hope to see them all someplace. Thanks to my brother and all my friends back home, who have made visits back such a pleasure; there are too many to name, but I must mention Patryk: cheers!

The DMO group at McGill made things lively and stimulating. Thanks to Bruce Reed for some dinners to remember; Jamie, Louigi, Luc, Nic, Nic, Nic, Ross and Simon for fun times in Barbados as well as McGill; Sean for covering my lectures so I could make it there (also a roadtrip); and Conor for somehow managing to get a Saffer to play ice hockey (I can almost skate backwards now).

I am grateful to the external referee, Lisa Fleischer, and the members of my committee, which aside from Bruce and Adrian included Louigi Addario-Berry and Mohit Singh. Their comments and corrections were very helpful (Louigi was especially generous with his time).

Finally, thank you to Alejandra, for keeping me calm and just being wonderful; and to my parents, for their constant love and support from afar, for which I am profoundly grateful.

Contribution of Authors

The content of Chapter 3 and Chapter 7 is based on joint work with Navin Goyal and Bruce Shepherd. The content of Chapter 5 is based on joint work with Bruce Shepherd. The presentation has been substantially reworked and expanded. The work in the remaining chapters is my own.

Copyright notice

Contents

Chapter 1

Introduction

1.1 The topic

Robust network design is essentially the study of network design in the face of variable or uncertain demands. The main motivations (discussed in Chapter 2) come from the design of communication networks, but for now we keep the discussion somewhat broad, and define, in a general form, the central optimization problem that will concern us in this thesis.

The problem was first defined in this generality by Ben-Ameur and Kerivin [21, 22]. Imagine we are tasked with building a private network between some set of "terminals" (which are simply the entities—computers or otherwise—we need to connect), by buying capacity on some underlying network. We wish to do this in order to ensure a certain level of service; by reserving the bandwidth, we ensure that the private network is not adversely affected by traffic from the rest of the network. Reserving this bandwidth is expensive, and we want to spend as little as possible.

We represent the underlying network with an undirected graph G, with node set V and edge set E. A directed graph might also be appropriate, but we consider only undirected graphs, and in fact throughout this thesis. Each edge $e \in E$ has an associated nonnegative cost $c(e)$, representing the cost per unit of bandwidth on that edge, so we are assuming a simple linear cost model here. There are no capacity constraints—as much bandwidth as needed can be bought on each edge. The terminals are specified by some subset $W \subset V$; let $k = |W|$ denote the number of terminals. We will identify W with $\{1, 2, \ldots, k\}$, in order to conveniently index by terminals.

We must specify the requirements of our private network: to set the scene, we begin with a very simple problem. Suppose we know exactly the amount of bandwidth required for each pair of terminals. In other words, for every $i, j \in W$, $i \neq j$, the rate at which data needs to be sent from terminal i to terminal j. We call this the *demand* from i to j, and denote it by D_{ij}. We can package the entire pattern of demands into a single matrix D, indexed by the terminals, known as the *demand matrix* or *traffic matrix*.

We wish to find a capacity reservation of least cost which can *support* the demand matrix D—in other words, there should be enough capacity to simultaneously route the required demand between each pair. This is extremely simple, essentially because there is no sharing of capacity between different routing pairs. Compute a shortest path between every pair of terminals; say P_{ij} is a shortest path between i and j. We then reserve, cumulatively, an amount D_{ij} on path P_{ij}, for every $i \neq j \in W$. It is clear that this is optimal.

Uncertain demands However, the traffic pattern of a real-world network is typically not fixed; rather, it varies over time. Moreover, it is often difficult to measure or estimate traffic patterns reliably in large networks, even if these traffic patterns are roughly static. Robust network design deals with this uncertainty in traffic patterns via the methodology of robust optimization. We assume that the demand, while not fixed, comes from some prescribed *universe* of possible demands. The solution we give must be able to route any demand matrix in this universe.

So let $\mathcal{U}$, the universe, be some given subset of $\mathbb{R}_+^{k \times k}$. In fact, it will turn out that the problem we will consider is unaffected by replacing $\mathcal{U}$ with its convex hull, so we may assume that $\mathcal{U}$ is a convex set.

Routing strategies An important issue that needs to be considered is the *routing strategy*. Conceivably, one might allow the routing to depend on the current demand—the specific (feasible) demand pattern across the network at the current moment. Such a flexible approach turns out to be rather impractical, and so we will be primarily interested in *oblivious* routings. This means that the routing used for a particular pair of terminals must be fixed in advance, and cannot depend on the current traffic pattern.

Thus a solution to a robust network design instance is given by a *routing template*, as well as a *capacity reservation* (but we will see shortly that this is easily computed given the routing template). A routing template must specify, for every pair $i, j \in W$, a routing between this pair; this is the routing that will be used irrespective of the current demand matrix. The two most important oblivious routing variants are:

- **Multipath routing** (MPR): the solution is specified by a unit i-j flow f_{ij}, for each pair $i, j \in W$. The routing template $\mathcal{P}$ is defined by $\mathcal{P} := \{f_{ij} : i, j \in W\}$. Given a demand matrix D in the universe, the flow from i to j will be routed proportionally according to this template. In other words, the amount of flow on an edge e will be $D_{ij} f_{ij}(e)$.

- **Single-path routing** (SPR): the flows in the routing template are restricted to be integral, i.e., each pair i,j routes along a single path P_{ij}.

It is also often useful to consider

- **Tree routing** (TR): the flow template is again integral, but with the additional restriction that the union of all the routing paths must form a tree.

Cost and capacity reservations Once the routing template is specified, a capacity reservation $u(e)$ must be made on every edge $e \in E$. The cost of the solution is then given by $C(\mathcal{P}) := \sum_{e \in E} c(e)u(e)$. The capacity reservation must be valid, given the routings that we've picked; for any feasible demand, the total load on any edge must not exceed its capacity. Thus the exact minimal required capacity on edge e is simply

$$u(e) := \max_{D \in \mathcal{U}} \sum_{i,j \in W} f_{ij}(e) D_{ij}. \tag{1.1}$$

Altogether, this defines the robust network design problem (RND). In summary:

Given: Graph $G = (V, E)$ with costs $c : E \to \mathbb{R}_+$, terminals $W \subset V$, universe $\mathcal{U}$.

Solution: A routing template $\mathcal{P} = \{f_{ij} : i, j \in W\}$, where each f_{ij} is a unit i-j-flow, either fractional or integral depending on the routing scheme required.

Minimize: The total cost $C(\mathcal{P}) = \sum_{e \in E} c(e) u(e)$, where $u(e)$ is given by (1.1).

Symmetric vs. asymmetric We have considered flow from i to j (corresponding to entry D_{ij} in the demand matrix) to be distinct from flow from j to i (corresponding to entry D_{ji}). In some situations, it makes sense to consider demand to be undirected, making these two demands indistinguishable. In this case, we say that the problem is *symmetric*.

This is most simply embedded within the above framework by considering a universe consisting only of lower-triangular demand matrices; the undirected demand between a pair $\{i, j\}$ is then given by D_{ij} if $i < j$, or D_{ji} if $i > j$. However, it will be more convenient notationally for us to take D to be symmetric, so that $D_{ij} = D_{ji}$ and refers to the same demand. The flow template must also be symmetric, with the only change being a reversal of direction: for MPR, $f_{ij} = -f_{ji}$, and for SPR, $P_{ij} = P_{ji}$, for all $i, j \in W$. The correct form of (1.1) in the symmetric case is then

$$u(e) := \max_{D \in \mathcal{U}} \sum_{i < j \in W} f_{ij}(e) D_{ij}; \tag{1.2}$$

only the lower triangular part of the demand is used. We will thus consider demand universes in the symmetric case to be subsets of $\mathbb{R}_+^{\binom{k}{2}}$.

We will see later that the robust design problem with multipath routing is solvable in polynomial time with linear programming methods. This is not the case for single-path routing, which is in general NP-hard. Different forms of the universe lead to different combinatorial optimization problems. Various well-studied network design problems, for example Steiner tree and single-sink rent-or-buy (even multicommodity buy-at-bulk, as we will see in Chapter 5) can be found as special cases by choosing the universe suitably.

The hose model An important class of universes is defined by the so-called *hose model*. It has received a lot of interest in the networks community as a way of specifying requirements for a virtual private network; this will be discussed in detail in the next chapter. Each terminal $i \in W$ has an associated incoming capacity b_i^- and outgoing capacity b_i^+. For any feasible demand pattern, the total demand from all other terminals to terminal i cannot exceed b_i^-, and similarly the total demand from i to the other terminals cannot exceed b_i^+. This defines a universe, called a *hose polytope*:

$$\mathcal{H}(\boldsymbol{b}^+, \boldsymbol{b}^-) := \{D \in \mathbb{R}_+^{k \times k} : \textstyle\sum_j D_{ij} \le b_i^- \text{ and } \sum_j D_{ji} \le b_i^+ \;\; \forall\, i \in W\}. \tag{1.3}$$

We write simply $\mathcal{H}$ when there is no ambiguity.

There is also a symmetric version. Since demands are undirected, there is only a single capacity b_i for each terminal i:

$$\mathcal{H}(\boldsymbol{b}) := \{D \in \mathbb{R}_+^{\binom{k}{2}} : \textstyle\sum_j D_{ij} \le b_i \;\; \forall\, i \in W\}; \tag{1.4}$$

note that D is necessarily symmetric in the above.

For reasons that will be made clear in the next chapter, the robust network design problem with universe given by a hose polytope is called the *VPN problem*. This can be symmetric or asymmetric, with single-path or multipath routing; when not specified, we will assume single-path routing.

1.2 Contribution of this thesis

Chapter 2 gives the background on the topic of robust network design, and various areas leading up to it. Robust optimization is discussed first; it provides the general framework. Robust network design is discussed next, followed by the distinct but related work on oblivious routing, and finally a discussion of some more theoretical work on network design. The material in this chapter is not necessary for an understanding of the technical results in the remainder of the thesis.

Chapter 3 considers the symmetric hose model with single-path routing. It had been conjectured by a number of authors that this problem has the property that there is always an optimal solution in the form of a tree; this became known as the "VPN Conjecture". One reason for its importance is that the complexity status of the problem was open, and since it was known how to find an optimal tree solution, a positive resolution of this conjecture would also imply that the symmetric VPN problem is polynomially solvable. In Chapter 3, we give a proof of the conjecture, in full generality.

Chapter 4 compares multipath and single-path routing for the symmetric hose model. We demonstrate a counterexample to a stronger multipath version of the VPN Conjecture, by showing that in some cases the optimal multipath solution is cheaper than the best single-path routing. We also begin an investigation into the worst-case gap between the MPR and SPR optima.

A fundamental problem regarding the approximability of the general robust network design is considered in Chapter 5. It is shown that the problem is in general hard to approximate within polylogarithmic factors, even when reasonable constraints are put on the complexity of the demand universe. The proof proceeds by showing that the uniform buy-at-bulk problem, which is known to be hard to approximate, can in fact be encoded as a special case of robust network design.

Given this negative result, it is interesting to ask for broader classes of universes where we *can* approximate to within a constant factor. In Chapter 6, we consider a natural generalization of the symmetric hose model, the *tree demand model*. Here, the universe is specified by a given edge-capacitated tree, which need bear no relation to the network graph except that the leaves of the tree correspond exactly to the terminals in the network. A demand is considered feasible if it can be routed on the tree without violating the edge capacities. We give an algorithm for SPR with this demand universe, and show that it gives a constant factor approximation.

In Chapter 7, different routing schemes are compared. Primarily, we are interested in the question of how much better dynamic routing can be, compared to its oblivious counterpart. We give a construction showing that oblivious routing (even multipath) can in general be a logarithmic factor worse; the universe used for this construction is an asymmetric hose universe (1.3).

In the concluding chapter, we discuss some avenues for further work in this area. Many of these questions are motivated by the work in this thesis.

The results of Chapter 3, and some of Chapter 4, were published in [69]. The work in Chapter 5 and Chapter 6 appeared as [110]. The work in Chapter 7 was published as [70].

1.3 Notation and conventions

Here we give some definitions and notations that will be used throughout; most are standard in combinatorial optimization.

The reals are denoted by $\mathbb{R}$, the nonnegative reals by $\mathbb{R}_+$, the integers by $\mathbb{Z}$ and the nonnegative integers by $\mathbb{N}$.

Vectors will be notated using boldface: $\boldsymbol{v}$ is a vector, and v_i is one of its components. Generally we think of vectors as column vectors, which we may write explicitly as, e.g.,

$$\boldsymbol{v} = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix}.$$

If $\boldsymbol{u}$ and $\boldsymbol{v}$ are two vectors, then $\binom{\boldsymbol{u}}{\boldsymbol{v}}$ refers to the vector obtained from their concatenation. The transpose of a vector $\boldsymbol{v}$ is denoted $\boldsymbol{v}^T$.

The notation $\mathbb{1}$ refers to an "indicator function", but will be used slightly more generally. Given any Boolean condition P, $\mathbb{1}_P$ is defined to be 1 if P is true, and 0 if P is false. So for example, the characteristic function of a set could be written $\chi_A(v) = \mathbb{1}_{v \in A}$.

A vector written $\boldsymbol{e_i}$ is zero everywhere, except at index i where it takes the value 1. In other words, $(\boldsymbol{e}_i)_j = \mathbb{1}_{i=j}$.

Uppercase calligraphic script such as $\mathcal{P}$ and $\mathcal{H}$ will usually denote sets (in particular, sets of routings, and sets of possible demand matrices).

An undirected *graph* $G = (V, E)$ consists of a *node set* V and *edge set* E. Each edge $e \in E$ consists of a pair of elements of V; we will sometimes use the notation uv as shorthand for the edge $\{u, v\}$. In this thesis, if not specified graphs are taken to be undirected and simple, with no self loops. The *bidirection* of G is the directed graph (V, A), where each edge $e = uv \in E$ becomes a pair of arcs $(u, v), (v, u) \in A$. An arc (u, v) has *tail* u and *head* v.

For any subset S of nodes in a graph G, we denote by $\delta_G(S)$ the set of edges with exactly one endpoint in S; if the context is clear we simply write $\delta(S)$. On a directed graph, we denote by $\delta^+(S)$ the set of arcs with tail in S and head in $V \setminus S$; similarly $\delta^-(S)$ denotes the set of arcs with tail in $V \setminus S$ and head in S. If the graph G is undirected, we use, e.g., $\delta^+(S)$ to refer to the outgoing arcs in the bidirection of G.

A single-commodity flow on G is a function $f : A \to \mathbb{R}_+$ on the bidirection of G. The *supply* of a flow f at a node $v \in V$ is $\text{supply}_f(v) := \sum_{a \in \delta^+(v)} f(a) - \sum_{a \in \delta^-(v)} f(a)$. We can also talk about the *demand* at v, which is just the negative of the supply. Given a vector $\boldsymbol{b} \in \mathbb{R}^V$, a $\boldsymbol{b}$-flow is a flow f with $\text{supply}_f(v) = b_v$ for all $v \in V$. An *i-j-flow* is a unit flow from i to j, i.e., a $(\boldsymbol{e_i} - \boldsymbol{e_j})$-flow. Similarly an *$i$-$j$-path* is a simple path from i to j. Sometimes it is convenient to use an alternative formulation of flows via a path decomposition: then $f(P)$ refers to the weight of flow f along path P. Such a decomposition is not unique for a given arcwise defined flow however. The addition and subtraction of flows on a bidirected instance is defined in the natural way, with flow in opposing directions along a digon pair cancelling. Given two flows f and g,

$$(f+g)(a) = \max\{0, f(a) + g(a) - f(a^-) - g(a^-)\},$$

where a is any arc in the bidirected instance and a^- is its reverse arc. Note that with this definition, only one of $(f+g)(a)$ and $(f+g)(a^-)$ can be nonzero. We may also define $f(e)$ on a bidirected instance to be the quantity of flow on edge e, irrespective of its direction; this is always nonnegative.

A multicommodity flow $\boldsymbol{f}$ on G consists of a vector of flows f_r for r in some index set, which could for example be the set of pairs of terminals.

Chapter 2

Background

2.1 Robust optimization

2.1.1 The basic paradigm of robust optimization

Robust optimization is primarily a tool for optimizing under uncertainty. Suppose we are given some real-world optimization problem—for now, say a large LP,

$$\min \quad \boldsymbol{c}^T\boldsymbol{x} \qquad \text{s.t.} \qquad A\boldsymbol{x} \geq \boldsymbol{b}.$$

An LP solver can determine an optimal solution $\boldsymbol{x}^*$ to this program quite efficiently. However, there is a potential problem: some of the coefficients in our LP—elements of A and b in particular—are not exactly known quantities, but rather estimates. So in fact, the "real" optimization problem is

$$\min \tilde{\boldsymbol{c}}^T\boldsymbol{x} \qquad \text{s.t.} \qquad \tilde{A}\boldsymbol{x} \geq \tilde{\boldsymbol{b}},$$

where $\tilde{A}$, $\tilde{\boldsymbol{b}}$ and $\tilde{\boldsymbol{c}}$ are unknown, but "close" to our estimates A, $\boldsymbol{b}$, $\boldsymbol{c}$.

The difficulty that may then arise is that our computed optimum $\boldsymbol{x}^*$ might not even be a feasible solution to the "real" LP! If the constraints are hard, meaning that no violation can be tolerated, this is a serious problem. Moreover, even small perturbations in the uncertain data can cause these violations to be very large. In [27], the authors considered random perturbations on problems in the NETLIB family of test problems, and found that in many of them, even perturbations of the order of 0.01% could lead to large constraint violations.

One approach to this problem is *stochastic optimization* (see, e.g., [119]), and in particular *chance-constrained optimization* [40]. This approach assumes knowledge of the distribution of the uncertainties, and asks for the best solution which is feasible with probability at least $1 - \epsilon$, for some specified tolerance ϵ. This is a very reasonable goal, but a problem with this approach is that we often do not have a good handle on the distribution of the uncertain coefficients. The problems obtained from chance-constrained optimization are also most often intractable ([96], cf. [23]).

Robust optimization is a newer paradigm for dealing with uncertainty, and it has been extremely successful to date. This method proposes to handle uncertainty by computing a solution $\boldsymbol{x}$ that is feasible for a whole *set* of possible values for the uncertain data. On first glance, this is just a very conservative version of stochastic optimization—a special case where the tolerance is set to zero, and we must ensure feasibility for all possibilities in the support of the distribution. However, there is a lot of flexibility hidden in the choice of uncertainty set, and there is a huge gain in the tractability of the optimization problems, as we will see shortly. Indeed, somewhat backwardly, it turns out that robust optimization gives a useful attack on chance-constrained optimization problems!

Let us now define the underlying optimization problem, and the robust approach to it, formally in the general setting.

Definition 2.1 ([25])**.** An *uncertain optimization problem* is a family of deterministic optimization problems $P(\boldsymbol{\zeta})$, where $\boldsymbol{\zeta} \in \mathbb{R}^M$ represents the uncertain data. For any fixed $\boldsymbol{\zeta}$, the optimization problem can be specified as

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & f(\boldsymbol{x},\boldsymbol{\zeta}) \\ \text{s.t.} \quad & F(\boldsymbol{x},\boldsymbol{\zeta}) \in \mathcal{K} \end{aligned} \tag{2.1}$$

for some choices of f, F and $\mathcal{K} \subset \mathbb{R}^m$.

Given some *uncertainty set* $\mathcal{U}$, the *robust counterpart* to the above uncertain program is

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \left\{ \sup_{\boldsymbol{\zeta}\in\mathcal{U}} f(\boldsymbol{x},\boldsymbol{\zeta}) \ : \ F(\boldsymbol{x},\boldsymbol{\zeta}) \in \mathcal{K} \ \ \forall \boldsymbol{\zeta} \in \mathcal{U} \right\}; \tag{2.2}$$

the optimum choice of $\boldsymbol{x}$ to this robust counterpart is called the *robust optimum*.

So the robust optimum is the solution to a min-max type problem, minimizing the worst possible outcome over all data in some set, while ensuring feasibility. In fact, we can simplify things slightly and assume without any loss of generality that the objective does not depend on the uncertain data. This follows by rewriting (2.1) as the following equivalent program:

$$\begin{aligned} & \min \quad t \\ \text{s.t.} \quad & f(\boldsymbol{x},\boldsymbol{\zeta}) \le t \\ & F(\boldsymbol{x},\boldsymbol{\zeta}) \in \mathcal{K} \end{aligned} \tag{2.3}$$

Various special cases of the above are of interest from an optimization standpoint: linear programs (which we will see next), conic programs, quadratic programs, semidefinite programs etc. For each, one can ask questions about the tractability of the robust version of the optimization problem. We will be almost exclusively interested in *robust linear programs*, such as the example at the start of this section. Here is the definition, similar to Definition 2.1 but with a linear program as the uncertain optimization problem; following the above comment, we also assume that the objective is certain.

Definition 2.2. An *uncertain linear program* is a family of linear programs $P(A,\boldsymbol{b})$ given by

$$\begin{aligned} \min_{\boldsymbol{x}\in\mathbb{R}^n} \quad & \boldsymbol{c}^T\boldsymbol{x} \\ \text{s.t.} \quad & A\boldsymbol{x} \ge \boldsymbol{b}. \end{aligned} \tag{2.4}$$

Given an uncertainty set $\mathcal{U}$, the robust counterpart to this uncertain LP is

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \{ \boldsymbol{c}^T\boldsymbol{x} : A\boldsymbol{x} \ge \boldsymbol{b} \ \ \forall (A,\boldsymbol{b}) \in \mathcal{U} \}; \tag{2.5}$$

2.1.2 Tractability of robust optimization

Perhaps the central question in robust optimization is that of *computational tractability*. The approach is only useful if we can in fact solve robust optimization problems of interest. As we are coming from a theoretical perspective, we will mostly equate tractability with polynomial solvability. For cases where the optimization problem is NP-hard, we will also be interested in good approximation algorithms—polynomial time algorithms with a specified performance guarantee. From a more applied perspective, often theoretically difficult problems do need to be attacked, in which case various heuristics need to be evaluated to determine if the problem at hand is amenable to computation. Fortunately, there is a large class of such problems that are both interesting and tractable. In particular, robust linear programming is solvable under very reasonable assumptions. We need the following definition:

Definition 2.3. Let $\mathcal{P}$ be a polytope in $\mathbb{R}^n$, and let φ be the size complexity[1] of $\mathcal{P}$.

Given a point $\boldsymbol{z} \notin \mathcal{P}$, a *separating hyperplane* is a vector $\boldsymbol{y} \in \mathbb{R}^n$ such that $\boldsymbol{y}^T\boldsymbol{z} > 0$ and $\boldsymbol{y}^T\boldsymbol{x} \leq 0$ for all $\boldsymbol{x} \in \mathcal{P}$.

A *separation algorithm* for $\mathcal{P}$ is an algorithm which given any $\boldsymbol{z} \in \mathbb{R}^n$, in time polynomial in n, φ and the size complexity of $\boldsymbol{z}$, determines whether $\boldsymbol{z} \in \mathcal{P}$, and if not, returns a separating hyperplane.

If there exists a separation algorithm for $\mathcal{P}$, we call it *separable*.

The polytopes we consider will all have size complexity polynomial in n, and we may ignore this technicality. The above definition can be extended to non-polyhedral convex sets via the notion of *weak separation*; see [72] for details. The results below can then be extended to this setting; we prefer to constrain ourselves to polytopes in order to simplify the presentation.

[1] This is essentially an upper bound on the number of digits needed to describe any component of any extreme point of $\mathcal{P}$ as a rational number; see [72])

Theorem 2.4 (Robust LP separation [26])**.** *If $\mathcal{U}$ is separable, and the uncertain LP is compact, then the feasible set $\{\boldsymbol{x} : A\boldsymbol{x} \geq \boldsymbol{b}\ \forall\,(A, \boldsymbol{b}) \in \mathcal{U}\}$ for the robust counterpart is separable.*

Proof. It is sufficient to determine, for a given i, whether the i'th row of the system $A\boldsymbol{x} \geq \boldsymbol{b}$ is always feasible. Let $\mathcal{U}_i$ be the projection of $\mathcal{U}$ onto the coordinates corresponding to $\boldsymbol{a_i}$ and b_i. Now we clearly have that $\boldsymbol{a}_i^T\boldsymbol{x} \geq b_i$ for all $\binom{\boldsymbol{a}_i}{b_i} \in \mathcal{U}_i$ if and only if the optimal solution to the system

$$\min \quad (\boldsymbol{x}^T \ \ -1)\begin{pmatrix}\boldsymbol{a}_i\\ b_i\end{pmatrix} \qquad \text{s.t.} \qquad \begin{pmatrix}\boldsymbol{a}_i\\ b_i\end{pmatrix} \in \mathcal{U}_i \tag{2.6}$$

is nonnegative. Write $\boldsymbol{\alpha} := \binom{\boldsymbol{a_i}}{b_i}$, and for ease of notation, suppose that the coordinates corresponding to $\mathcal{U}_i$ come first in the space of the full universe, so that we can write

$$\boldsymbol{\alpha} \in \mathcal{U}_i \quad \text{iff} \quad \begin{pmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\gamma}\end{pmatrix} \in \mathcal{U} \qquad \text{for some } \boldsymbol{\gamma}.$$

Then we may rewrite (2.6) as

$$\min \quad (\boldsymbol{x}^T \ \ -1)\boldsymbol{\alpha} \qquad \text{s.t.} \qquad \begin{pmatrix}\boldsymbol{\alpha}\\ \boldsymbol{\gamma}\end{pmatrix} \in \mathcal{U}$$

Since $\mathcal{U}$ is separable, by the equivalence of separation and optimization [72] we can determine the optimal value of this program. □

Using the ellipsoid method, we are thus able to solve the robust counterpart in polynomial time.

If $\mathcal{U}$ can be defined by a compact system, we can do even better than the above result. Ben-Tal and Nemirovski [26] also showed that if both the universe and the robust LP are specified in a compact form, then the robust counterpart itself can be made into a compact LP.[2] To obtain maximum generality, we need the concept of an extended formulation:

[2] The result is hidden away slightly in the appendix of the paper—see [26, Remark 4.1]. In the main text, they consider more general ellipsoidal uncertainty sets, where the same technique yields a conic program.

Definition 2.5. Let $\mathcal{P}$ be a polytope in $\mathbb{R}^n$. An *extended formulation* for $\mathcal{P}$ is a system $P\boldsymbol{x} \leq \boldsymbol{q}$, where $\boldsymbol{x} \in \mathbb{R}^{n+m}$, $\boldsymbol{q} \in \mathbb{R}^r$ and P is an $r \times (n+m)$ matrix, for some nonnegative integers m and r, such that

$$\mathcal{P} = \{\boldsymbol{z} \in \mathbb{R}^n : \exists\, \boldsymbol{u} \in \mathbb{R}^m \text{ with } P\tbinom{\boldsymbol{z}}{\boldsymbol{u}} \leq \boldsymbol{q}\}.$$

In other words, $\mathcal{P}$ is the projection onto the first n coordinates of the feasible polytope for the system. If the extended formulation is of size polynomial in n, we say it is a *compact extended formulation* for $\mathcal{P}$.

Theorem 2.6 (Robust LP formulation [26])**.** *If $\mathcal{U}$ is polyhedral, and is specified explicitly via a compact extended formulation, and if the robust LP is compact, then the robust counterpart has a description as a compact LP.*

Proof. Let us assume for now that our uncertain LP has only a single constraint, $\boldsymbol{a}^T\boldsymbol{x} \geq b$. We will see at the end how the full result easily follows.

Let P and $\boldsymbol{q}$ define the compact extended formulation for $\mathcal{U}$, so that

$$\mathcal{U} = \{\begin{pmatrix}\boldsymbol{a}\\ b\end{pmatrix} : \exists\, \boldsymbol{u} \text{ where } P\begin{pmatrix}\boldsymbol{a}\\ b\\ \boldsymbol{u}\end{pmatrix} \leq \boldsymbol{q}\}.$$

Given a fixed $\boldsymbol{x}$, the requirement $\boldsymbol{a}^T\boldsymbol{x} \geq \boldsymbol{b}$ for all $\tbinom{a}{b} \in \mathcal{U}$ is equivalent to asking whether the optimum of the following linear program is nonnegative:

$$\begin{aligned} \min \quad & (\boldsymbol{x}^T \;\; -1)\begin{pmatrix}\boldsymbol{a}\\ b\end{pmatrix} \\ \text{s.t.} \quad & P\begin{pmatrix}\boldsymbol{a}\\ b\\ \boldsymbol{u}\end{pmatrix} \leq \boldsymbol{q}. \end{aligned}$$

By weak LP duality, if the dual LP has a nonnegative solution, this provides a certificate that the above optimum is nonnegative. Thus feasibility of $\boldsymbol{x}$ is equivalent to feasibility of the following system:

$$\begin{aligned} \boldsymbol{q}^T z &\geq \boldsymbol{0} \\ P^T \boldsymbol{z} &= \begin{pmatrix} \boldsymbol{x} \\ -1 \\ \boldsymbol{0} \end{pmatrix} \\ \boldsymbol{z} &\geq \boldsymbol{0} \end{aligned}$$

We thus have the following LP formulation of the robust problem:

$$\begin{aligned} \min \quad & \boldsymbol{c}^T \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{q}^T z \geq \boldsymbol{0} \\ & P^T \boldsymbol{z} = \begin{pmatrix} \boldsymbol{x} \\ -1 \\ \boldsymbol{0} \end{pmatrix} \\ & \boldsymbol{z} \geq \boldsymbol{0} \end{aligned}$$

This was for only a single constraint. It is clear however how to generalize this to a system $A\boldsymbol{x} \geq \boldsymbol{b}$ with multiple constraints. Each row $\boldsymbol{a_i}^T \boldsymbol{x} \geq b_i$ in the system has an associated uncertainty set $\mathcal{U}_i$ obtained from $\mathcal{U}$ by projecting onto the relevant coordinates. Each $\mathcal{U}_i$ necessarily has an extended compact formulation, since $\mathcal{U}$ does. Thus for each i, we have a dual system

$$\boldsymbol{q_{(i)}}^T \boldsymbol{z_{(i)}} \geq \boldsymbol{0}, \qquad P_{(i)} \boldsymbol{z_{(i)}} \geq \begin{pmatrix} \boldsymbol{x} \\ -1 \\ \boldsymbol{0} \end{pmatrix}, \qquad \boldsymbol{z_{(i)}} \geq \boldsymbol{0}.$$

Since each row must separately be feasible, we simply combine all these constraints into a single larger system. □

Theorems 2.4 and 2.6 are very useful, and we will see later that important results in robust network design and oblivious routing in congestion minimization can be seen as consequences of these results.

We end this section with a caveat. While the class of robust linear programs is already large and useful, it is somewhat more restrictive than we might think given our experience with (non-robust) LPs. When modelling without robustness, we have the flexibility of adding new variables to our system, and these need not have any "real" meaning. For example, consider the convex program

$$\min\ \boldsymbol{c}^T\boldsymbol{x} \quad \text{s.t.} \quad \|\boldsymbol{x}-\boldsymbol{p}\|_1 \le 1. \tag{2.7}$$

The constraint $\|\boldsymbol{x}-\boldsymbol{p}\|_1 \le 1$ can be represented by the system

$$x_i - p_i \le u_i, \quad p_i - x_i \le u_i, \quad \sum_i u_i \le 1,$$

and so the convex program can be represented easily as a linear program. However, if $\boldsymbol{p}$ is uncertain, we cannot apply the same transformation. The difficulty is that the new variables u_i must be specified as part of our feasible solution, and *may not vary as a function of the uncertain coefficients*. This is completely different from asking for a solution to (2.7) which is feasible for all $\boldsymbol{p} \in \mathcal{U}$. Indeed, it is observed in [23] that Eq. (2.7), with $\boldsymbol{p}$ uncertain in a given polyhedral uncertainty set, is in general NP-hard. Given the earlier positive results for robust LPs, it follows that it is impossible to represent this robust convex program as a robust LP. It also follows that there can be no general tractable extension of the robust LP concept to one that allows "scenario-dependent" variables.

2.1.3 Choices of uncertainty set, and relations to stochastic optimization

In this thesis, we will be concerned only with various polyhedral uncertainty sets, as per the discussion above. It is interesting to briefly discuss some other important cases.

One of the original motivations for robust optimization was as a more tractable alternative to stochastic optimization. Since robust optimization uses a very conservative notion of feasibility—the solution must be feasible for every possible choice of data within the uncertainty set—it is not at first clear that there is a close connection.

In a stochastic optimization problem, our uncertain data is assumed to be governed by some given probability measure μ, and we are required to find a solution so that the probability of failure is at most some tolerance ϵ. To obtain a robust optimization problem, the idea is to pick as the uncertainty set some subset U of the support of the measure, such that $\mu(U) \geq 1 - \epsilon$. Clearly, a solution to the robust problem with this uncertainty set will be feasible for the stochastic optimization problem, though it need not be optimal (note that there are many possible choices for U). Such a robust problem is called a *safe tractable approximation* [25, 23] to the chance-constrained problem.

Ben-Tal and Nemirovski [25, 27] showed that for a wide range of natural probability distributions, a safe tractable approximation can be found with an *ellipsoidal* uncertainty set (we define this next). This class of distributions includes, for example, the case where each piece of data is independently random and Gaussian, or independent and bounded.

Define an *ellipsoid* in $\mathbb{R}^n$ to be a set of the form

$$\{P\boldsymbol{u} + \boldsymbol{r} : \boldsymbol{u} \in \mathbb{R}^m, \|Q\boldsymbol{u}\|_2 \leq 1\},$$

where Q is an $m \times m$ matrix, P is a $n \times m$ matrix, and $\boldsymbol{r} \in \mathbb{R}^n$. This is slightly different to the usual definition of an ellipsoid, due to the introduction of Q, which may be singular; it includes degenerate cases such as half-planes and cylinders, which are very useful in this setting. An *ellipsoidal uncertainty set* is then defined as one which can be expressed as the intersection of ellipsoids.

This class is particularly convenient, because it is shown in [26] that the robust counterpart of an uncertain linear program with an ellipsoidal uncertainty set can be cast as a *conic program*. Conic programs can be solved efficiently via interior point methods; see, e.g., [34] for details.

Although conic programs are tractable (especially by the theoretically motivated use of the term in this thesis), they are still more computationally demanding to solve than similarly sized linear programs. Perhaps more importantly, integer or mixed-integer versions of linear programs, while NP-hard, are much easier to deal with in practice than conic programs with integral constraints, as there are many tools and heuristics designed to handle them. This is one motivation for an alternative safe tractable approximation to certain chance-constrained problems introduced by Bertsimas and Sim [29]. They consider robust LPs of the form (2.5). Each entry of A and $\boldsymbol{b}$ has a nominal value, as well as upper and lower bounds. An extra parameter Γ is introduced, and it governs the maximum number of rows in the system where coefficients may differ from their nominal values. The heuristic motivation is that in some settings, it is "unlikely" that many coefficients vary from their nominal values, and Γ can be chosen larger or smaller depending on how conservative a solution is required.

In addition to this qualitative motivation, they show in [29] that this choice also yields a safe tractable approximation to certain kinds of chance-constrained problems (essentially the same as discussed in the previous paragraph). The tolerance ϵ depends on the choice Γ; in fact, $\epsilon \propto \exp(-\Gamma^2/C)$ for some problem-dependant constant C, so a small increase in Γ dramatically reduces the tolerance. Their uncertainty set (after taking the convex hull) can be shown to have a compact linear description, and hence falls within the framework of the previous section; thus an efficient linear program solver can be used. The linearity also aids in the tractability of discrete robust optimization variants; in [28] it is shown that certain network flow and design problems are tractable under this uncertainty model.

2.1.4 Other work

There is a large body of work on robust optimization, and we have touched only on the part most relevant to this thesis. See [23, 30] for more. We briefly mention some of the other broad directions.

In *robust conic optimization*, the linear constraints $A\boldsymbol{x} \geq \boldsymbol{b}$ are replaced with $A\boldsymbol{x} - \boldsymbol{b} \in \mathcal{K}$, where $\mathcal{K}$ is some convex cone. Cases of particular interest are conic quadratic optimization, where $\mathcal{K}$ is a Lorentz cone, and semidefinite programs, where $\mathcal{K}$ is the cone of positive semidefinite matrices of the appropriate dimension. These problems are no longer tractable for general separable uncertainty sets, but some special cases of interest are solvable [26, 52].

In *adjustable* robust optimization, some variables in the solution may have some flexibility to be changed depending on the actual values of the uncertain parameters. This is a robust analogue of 2-stage or multistage stochastic optimization. While very useful for modelling, this version is highly intractable except for certain special cases [24].

2.1.5 Historical notes

The roots of robust optimization stretch back to Soyster [127] in 1973. Significant advances were than made independently by three different groups: Ben-Tal and Nemirovsky, El-Ghouai and Lebret, and Kouvelis and Yu.

The discussion above follows most closely the work of Ben-Tal and Nemirovski, in particular their first two papers on the subject [26, 25]. El Ghaoui et al. arrived at essentially the same formulation; their original motivation was slightly different. In [51], they considered an overdetermined linear system $A\boldsymbol{x} = \boldsymbol{b}$. The standard solution of minimizing $\|A\boldsymbol{x} - \boldsymbol{b}\|_2$ is very sensitive; they consider instead a solution minimizing this under bounded uncertainty over A and $\boldsymbol{b}$. In [52], they considered robust semidefinite programs under norm-bounded uncertainty sets.

Both of these groups primarily take the perspective of continuous optimization, and the problems they consider are very much influenced by this. Kouvelis and Yu [100] consider *discrete* robust optimization problems (as we will, primarily, in this thesis). As operations researchers, they also spend considerably more effort on modelling issues, especially in considering how uncertainty sets might be determined. Most of the problems they consider are NP-hard however: as usual, discrete optimization is more difficult than its continuous counterpart.

In this thesis, we will encounter only robust linear programs, and typically the universes we consider will be separable polytopes. The main difficulty is that we will be be most interested in integral solutions; since these are in general intractable, we will need to understand the combinatorial structure of the specific problems we are dealing with to make progress.

2.2 Robust optimization for network problems

Having discussed robust optimization in its general form, it is now time to turn to more specific problems in network design.

Interestingly, from the perspective of this thesis, Kouvelis and Yu [100] appear to be the first to coin the term "robust network design". They consider a number of standard network design problems from a robustness perspective. The uncertainty in the problems they consider are in the costs (typically edge weights) of the instance. For example, consider perhaps the simplest of all network design problems, minimum spanning tree. Given uncertainties in the edge weight, and indeed an uncertainty set describing all possible combinations of edge weights, one may ask for a robust solution to the problem—i.e., a spanning tree that minimizes the worst possible cost. This problem is already NP-hard [136]. They also consider other important network design problems, e.g., the 1-median problem.

In this thesis, however, we will be concerned with a completely different kind of robust network design problem. Namely, we will be interested in *demand uncertainty*. The pattern of demands across the network will be uncertain and time-varying, and we will need a solution that is robust to these changes.

The early work on the hose model (which we will see next) coincided with the initial work on robust optimization. Work on robust network design (although not under that name) continued for quite some time without any connection to robust optimization; Altın et al. [2] appear to be the first to mention the connection.

2.2.1 ATMs and VPNs: the hose model

The hose model, already introduced in Chapter 1, has received a lot of attention in the networking community. Here we describe the motivations for this model, and survey the relevant literature.

This model was introduced independently in two different papers, first in 1997 by Fingerhut, Suri and Turner [58] in relation to broadband networks, and then in 1999 by Duffield et al. [46][3] in relation to the design of virtual private networks. Since the motivation of the second paper is somewhat closer to the discussion in this thesis, we reverse chronology and discuss it first.

Virtual private networks Duffield et al. consider the problem of specifying a *virtual private network* (VPN). The customer, perhaps a large corporation, wants to set up a private network connecting (say) various branches across the world. Building a completely separate physical network for this task would be extremely expensive and inefficient; rather, the goal is to build on top of existing communication networks (probably the internet). However, the customer has certain specific requirements from the VPN; the network must always be available and have sufficient bandwidth to meet the customer's needs. These service requirements imply that the network operator(s) will have to essentially reserve some capacity on the underlying network for the exclusive use of this customer. The amount and distribution of this reserved bandwidth depends on the exact customer requirements, and these need to be specified up front in some fashion. The simplest (at least from the network operator's perspective) way for this specification to be provided is via "point-to-point" demands. The customer specifies the bandwidth requirement for each pair of terminals in the VPN (different pairs may have different requirements), and

[3] The term "hose model" originates from this paper.

the network must be able to route all these demands simultaneously. This is known as the *pipe model*—the VPN can be thought of as a collection of pipes between the different terminals, and the specification is the capacity of each of these pipes.

While the pipe model is convenient for the network operator (it is very easy to determine what bandwidth needs to be allocated to the customer to guarantee their service requirements; see §1.1), it is not so convenient from the perspective of the customer. It may be difficult for the customer to obtain good estimates on each pairwise demand. Moreover, it is rather unlikely that the bandwidth demand between some given pair of terminals is really constant; more likely, demands will fluctuate over time. In order for the customer to give a specification in the pipe model which covers all likely circumstances, the worst-case demand between every pair would have to be used. But this could be very expensive. Imagine a situation where a VPN is used for a company's voice (telephone) network, or perhaps for more bandwidth-intensive video calls. Potentially any pair of terminals in the network might be involved in a telephone call; thus, bandwidth is required between every pair. However, an individual terminal will be involved in at most one call at any given time; thus only a fraction of the total bandwidth guaranteed by the pipe specification would be used at any particular moment.

In the hose model, a different and more flexible description of the customer's requirements is given. Rather than specifying the exact demand between each pair of terminals, only a maximal *aggregate* demand at each terminal needs to be specified. More precisely, the customer must specify, for each terminal in the VPN, upper bounds on the both the total incoming and total outgoing demand at that terminal. The service guarantee provided by the network operator is that any pattern of demands across the network must be routable, as long as they respect these upper bounds.

ATM broadband networks Now we come to the second motivation, considered by Fingerhut et al. [58]. The Asynchronous Transfer Mode (ATM) protocol has become an important standard in communication networks. Unlike packet switching networks like TCP/IP, the ATM protocol is specifically designed to handle real-time applications such as voice and video. For typical data applications, occasional variability in the available bandwidth and transmission latency is acceptable. But for voice and video, even very short delays cause problems. The ATM protocol has a number of features designed to ensure "quality of service" (QoS) guarantees.

While even the basics of the protocol are too complicated to describe here, there is a particular feature of ATM networks which is relevant to our discussion, namely the concept of *virtual circuits*. A virtual circuit (between some source and destination) provides guarantees on bandwidth for communications between this pair; in addition, all traffic on this circuit is routed along exactly the same path in the network, ensuring in-order transmission. Thus (simplistically speaking) a virtual circuit, once set up, gives the illusion of a private direct connection between its endpoints. This virtual circuit will remain until it is no longer needed (e.g., the video chat is completed), after which its bandwidth resources will be released.

For the network operator, providing these service guarantees causes extra complications compared to "best-effort" networks. When a request for a virtual circuit between two particular terminals is made, there is the possibility that insufficient bandwidth is available, given the current state of the network where other virtual circuits are already in place. If this happens, the virtual circuit is said to have been *blocked*. Of course no matter how much bandwidth the network provides, these resources will be limited, and it will not be possible to route an arbitrary collection of demands without blocking. Some constraints on what traffic patterns the network needs to be able to support must be provided. Fingerhut et al. proposed using hose constraints for this purpose, again as an improvement over the pipe model.

As well as introducing the model, the Fingerhut et al. paper proves a number of important theoretical results, discussed later in this chapter. Unfortunately, the paper remained unnoticed for some time, and some of the results it contained were subsequently rediscovered.

Symmetric and asymmetric As mentioned in Chapter 1, there is both a symmetric and asymmetric variant of the VPN problem. In the symmetric version, demands are considered to be undirected, and so there is only a single hose capacity b_i for each terminal i; in the asymmetric version, incoming and outgoing demands have separate bounds b_i^-, b_i^+.

Capacity reservations and costs The cost of a solution depends on how much capacity is required on each link. In the case of broadband ATM networks, the costs might relate to the costs of building the physical network, in the form of switches and high-bandwidth connections. In the case of VPN design, the network operator will charge the customer based on how much capacity is reserved for the VPN.

A reasonable cost model for both cases is to give each link e an associated cost function $f_e(u)$, which defines the cost to reserve u units of capacity on this link. This does not allow for specifying costs for switches—although it may be possible to roughly handle this by spreading the cost of a switch among its incident links, assuming that the cost of a switch increases with the amount of traffic it needs to route [58].

The simplest choice is to choose this cost function to be linear for all edges, and this is the case considered in the majority of the literature. The case where the cost function is arbitrary but concave—corresponding to "economies of scale" has also been recently considered [60, 120]. Altın et al. [5, 4] investigate a mixed-integer formulation of a robust version of the network loading problem, where capacity is packaged in price/capacity bundles, and the total capacity on an edge must be provided by purchasing some integral combination of these bundles.[4]

Another issue that can be considered is upper bounds on capacity reserva-

[4]This is exactly the cost model of the cable-type formulation of the buy-at-bulk problem, discussed in §2.4.

tions. In much of the literature (and in this thesis) it is assumed that there are no such bounds. The robust network design problem, even with the hose model, is NP-hard if there are capacity constraints [77]. Capacity constraints will play a much larger role in the theoretical results on oblivious routing that will be discussed in §2.3.

2.2.2 Routing strategies

An important issue that needs to be considered is the *routing strategy*. Conceivably, one might allow the routing to vary, depending on the current demand pattern. This is refered to as *dynamic* routing. This method, while the most flexible, and potentially the least costly (more on this in Chapter 7), suffers from being simply impractical to implement.

One reason for this is that information required to make a routing decision is distributed throughout the network; this information would have to be communicated via the network, in addition to the actual data. Very often, this overhead would be prohibitive in a fully adaptive scheme. Moreover, computational resources are limited in network switching equipment, particularly as routing decisions must be made very quickly in high throughput networks. Solving a multicommodity flow problem, while polynomially tractable, is not a simple computation. For ATM networks, it is not possible to change the routing of a virtual circuit during its lifetime. While a different virtual circuit could be used each time one is requested between a certain pair of terminals, this would mean that the state of the network (in terms of what capacity is available after taking into account the reservations of currently active virtual circuits) would depend on the history of requests in a complicated way. This would make it much more difficult to guarantee that the network is nonblocking; using a fixed path for virtual circuits between any specific pair, as is done by Fingerhut et al. [58], simplifies the situation dramatically.

It is also in fact hard to determine the optimal capacity allocation for dynamic routing, as shown by Chekuri et al. [42] (and earlier by Gupta et al. [77] for directed graphs); the approximability of this problem is still open.

Much more practical is a routing strategy which is *oblivious* (sometimes referred to as “static” or “stable”), meaning that it is fixed up front, and does not depend on the specific demand being routed. This is, essentially, how current network protocols work (changes to the routing templates can be made, but only very slowly compared to the functioning of the network). In addition to being much simpler to implement, this also helps ensure the stability of the network. The routing between two particular terminals will not be affected by changes in other parts of the network; since different paths may have different latency characteristics, this kind of stability can be important for real-time applications such as video streaming.

Oblivious routing strategies can be further subdivided into fractional and integral variants. In the integral variant, which we refer to as *single-path routing*, all flow for a given pair must follow a single path (and since the routing is oblivious, this must be fixed in advance). The fractional variant, *multipath routing*, allows the demand for a given pair to be split amongst a number of paths. Since the routing must still be oblivious, a template must be given that describes what fraction of the flow takes each path. The particular application and network protocol determines whether multipath routing is an option or not; certainly single-path routing is simpler to implement. In packet switching networks, the fractional nature of a multipath template might be implemented via randomization (although there are obstacles in practice). At each node, the next link for a packet is picked randomly, with probabilities proportional to the weights of the outgoing arcs in the template. The fraction of packets taking a particular edge will then be as specified by the flow template.

Some more restrictive models are also considered. In the more restrictive *tree routing* scheme, it is required that the path used for every pair is induced by the unique such path on a fixed tree (the same tree for all pairs) [46, 101]. Tree routing has some advantages over arbitrary single-path routings when the MPLS protocol is used [101]. The TCP/IP protocol uses a shortest-path routing strategy; each link in the network is given an associated weight, and the route taken between two terminals is simply the shortest path according to these weights. This more restrictive scheme in the context of robust network design is discussed in Altın et al. [3] and Chu et al. [44].

2.2.3 The polyhedral model

Given the discussion of robust optimization in §2.1, the following generalization of the hose model will come as no surprise. The connection with robust optimization had not yet been made however, and it was Ben-Ameur and Kerivin in 2003 [21, 22] that first introduced the concept of a polyhedral uncertainty set in this context. In their model, the customer specifies their service requirements by an arbitrary polytope. This polytope could be obtained in a number of ways; for example, traffic patterns could be monitored for some time; the convex hull (or a relaxation thereof) then defines a potential polytope. Of course, the hose model is a special case of this more general model; see (1.3).

2.2.4 Work on robust network design

There has been a large body of work on robust network design since these models were defined, particularly on the hose model. The literature spans a spectrum from the applied to the theoretical, and the discussion here emphasizes the theoretical work.

Tractability of multipath routing In the paper where they introduce the polyhedral model, Ben-Ameur and Kerivin [22] show that the multipath version of this problem is polynomially solvable. [5] Their approach is quite complicated however, and involves iteratively solving many linear programs in turn.

Not long after, and independently, Erlebach and Rüegg [54] consider multipath routing for the VPN problem, also showing that it is polynomially solvable. They show this by demonstrating a separation oracle for the problem; by the

[5] The paper was only published in 2005, but was apparently widely distributed as a technical report as early as 2002.

ellipsoid method, this then implies that the optimization problem is polynomial. Their proof method is easily seen to generalize to the full polyhedral model. Subsequently Altın et al. [2] and Hurkens et al. [86] independently demonstrated a compact LP formulation for the hose model. These two results also show that for multipath routing, capacity constraints can be added without any difficulty.

Knowing the connection with robust optimization, we can now see that the results follow as special cases of Theorem 2.4 and Theorem 2.6, respectively. The connection with robust optimization had not yet been made however.

Tree routing With single-path routing, we have a combinatorial optimization problem. Robust network design with the asymmetric hose model is APX-hard, as is seen by a reduction from the Steiner tree problem [58]. Kumar et al. [101] show that even finding the cheapest tree solution is hard. They also give a 10-approximation using LP rounding (see both [101] and [77]), and give computational evidence that their approach gives good solutions.

Single-path routing In [2], Altın et al. consider computational (branch-and-bound) approaches to the compact mixed-integer program formulation they define there. From the theoretical side, the first constant factor approximation algorithm for the asymmetric hose model was given by Gupta et al. [81, 79]. The constant was improved by Eisenbrand and Grandoni [47], and then again by Eisenbrand et al. [48] to its current best value. Because of the connections to other important network design problems, more details are deferred to §2.4.

The balanced case For a VPN instance, let $B^+ = \sum_{i \in W} b_i^+$ and $B^- = \sum_{i \in W} b_i^-$. If $B^+ = B^-$, the instance is called *balanced*. Fingerhut et al. [58] gave a 3-approximation for single-path routing in this case. Italiano et al. [87], unaware of their work, also gave a 3-approximation. The factor was improved to 2 by Eisenbrand et al. [48]. The complexity of the VPN problem restricted to balanced instances was open for some time; Rothvoß and Sanità [120] finally showed that it is NP-hard.

The VPN Conjecture Fingerhut et al. [58] showed that for the symmetric hose model, a 2-approximation is obtained by picking the cheapest *shortest path tree* solution; this refers to a solution where there is a specific hub node, and the routing between any pair of terminals is via this hub. In [101, 77], it is shown that in fact the *cheapest* VPN solution using tree routing is of this form. Unaware of the result by Fingerhut et al., Gupta et al. [77] replicated the factor 2 result.

Several researchers [2, 54, 78, 87] noticed at around the same time that the cheapest shortest path tree solution seemed to always be optimal, and not just within a factor of 2. This prompted them to independently formulate the so-called *VPN Conjecture*, which states simply that for every symmetric VPN problem instance, there is an optimal solution in the form of a tree.

Fingerhut et al. [58] observed that this is true if the network is a complete graph with uniform edge weights. Important progress was made by Hurkens, Keijsper and Stougie [86], who proved that the result is true if G is a cycle (and some other cases too). Grandoni et al. [71] later gave a much simplified proof on ring networks. These results are discussed in more details in Chapter 3 (particularly this last paper, which is crucial to our result), where we resolve the conjecture.

Other work Eisenbrand and Happ [50] consider a variation on the symmetric hose model where in addition to the hose constraints, the terminals are partitioned into groups, and communication occurs only between terminals in different groups. Despite the use of symmetric demands, this actually generalizes the *asymmetric* hose model, which corresponds to bipartitioning the terminals into a set of senders and a set of receivers. The authors give a constant factor approximation algorithm for the problem.

Altın et al. [4] consider a "hybrid" model which combines the hose model and pipe model. In addition to hose constraints, each pairwise demand is constrained to lie in a given interval. They investigate a mixed-integer formulation for the problem.

Italiano et al. [88] consider the problem of restoring the private network (given by a VPN tree) in the event of a link failure. Their goal is to provision backup capacity that can be used if any single link of the VPN tree fails. Jüttner et al. [91] compare the hose model with the pipe model (i.e., where a single conservative demand matrix is specified) from a more practical side. Poo and Wang [113] compare tree routing and multipath routing, again from a more practical perspective than is the treatment in this thesis. Meddeb [108] considers the problem of allocating *multiple* VPNs.

Oriolo [111] gives a pleasing result connecting routability of different traffic matrices. Fix a graph G, which we take to be complete, and say that a traffic matrix $D^{(1)}$ *dominates* traffic matrix $D^{(2)}$ if for every capacity reservation for which $D^{(1)}$ is routable, $D^{(2)}$ is routable too. Oriolo shows that $D^{(1)}$ dominates $D^{(2)}$ iff $D^{(1)}$ (thought of as defining a capacity reservation on G) supports $D^{(2)}$.

2.3 Routing and congestion

Contrary to robust network design as we have discussed it, where we may buy as much capacity as we wish on the edges of the network, this section will be about routing on capacitated graphs. Typically the aim will be to find routings which avoid overloading any edge too much—the cost measure will be the largest congestion of any link, rather than the total cost of buying capacity.

Metric embedding techniques play a central role in the proofs of most of these results, and that is where we begin the discussion.

2.3.1 Metric embedding

Metric embedding is concerned with "how well" a certain metric space can be embedded in some other metric space. Of particular interest will be embedding into *normed spaces*:

Definition 2.7. For $p \in [1, \infty]$, The normed space ℓ_p is defined on the space of all infinite sequences of reals $\boldsymbol{x} = (x_1, x_2, \ldots)$, with norm $\|\cdot\|_p$, where

$$\|\boldsymbol{x}\|_p := \left(\sum_{i=1}^{\infty} |x_i|^p\right)^{1/p}$$

for $p \in [1, \infty)$, and

$$\|\boldsymbol{x}\|_\infty := \max_i |x_i|.$$

The normed spaces ℓ_p^d for $d \in \mathbb{N}$ are defined in the same way, except the underlying space is $\mathbb{R}^d$.

The metrics we will be interested in embedding will be finite, since they will correspond to shortest path metrics on graphs. A finite metric space on a set of size n is called an *n-point metric space.*

A metric space (X, d) is said to be *isometrically embeddable* in a metric space (Y, σ) if there exists a map $f : X \to Y$ which preserves distances, i.e., for all $x, y \in X$, $d(x, y) = \sigma(f(x), f(y))$. It is straightforward to show that any n-point metric space can be isometrically embedded into ℓ_∞^n. However, this is false if ℓ_∞^n is replaced by ℓ_p for $p \in [1, \infty)$—even without any bound on the dimension. This motivates a study of *approximate* embedding. The crucial concept is as follows:

Definition 2.8. A metric space (X, d) is said to be α-embeddable into a metric space (Y, σ) if there exists $c > 0$ and a map $f : X \to Y$ such that

$$c \cdot d(x, y) \leq \sigma(f(x), f(y)) \leq \alpha \cdot c \cdot d(x, y).$$

The map f is then called an α-embedding.

The infimum over all α such that f is an α-embedding is called the *distortion* of f.

Two fundamental results in the area are the *Johnson-Lindenstrauss Lemma* [90] (which says that any n-point ℓ_2 metric can be well embedded into $\ell_2^{O(\log n)}$) and *Bourgain's Theorem* [33] (which says that any n-point metric can be embedded into ℓ_2 with $O(\log n)$ distortion).

Embedding into ℓ_1 The space ℓ_1 will have a starring role in what follows: the reason for this is its close connection with cuts in graphs. Consider a groundset V of size n, and some nontrivial cut $S \subset V$. The *cut semimetric* χ_S defined by S is defined simply by $\chi_S(u,v) = 1$ if S separates u and v, and $\chi_S(u,v) = 0$ otherwise. Now consider some nonnegative combination of cut semimetrics:

$$d(u,v) = \sum_{S \subset V} \alpha_S \chi_S(u,v),$$

where $\alpha_S \geq 0$ for all $S \subset V$. This defines a metric, and in particular, an ℓ_1 metric: simply index the dimensions by 2^V, and let $f(u)_S := \alpha_S \mathbb{1}_{u \in S}$. It is easily seen that $\|f(u) - f(v)\|_1 = d(u,v)$. In fact the converse is also true, and any finite ℓ_1-metric is induced by a nonnegative combination of cut semimetrics [13] (cf. [45, Chapter 4]).

Note that any tree metric, i.e., a metric induced by the shortest-path metric on a tree, is an ℓ_1-metric. Each edge in the tree defines a cut, and the entire metric can be written as a sum over edges of the cut semimetric induced by that edge.

2.3.2 Multicommodity flow and sparsest cut

Consider the multicommodity flow problem on an undirected graph $G = (V, E)$, with edge capacities $u(e)$. We are given k commodities, each with a source s_i, a sink t_i, and a demand requirement D_i. The problem is to fractionally route all the demands, while respecting the capacity constraints.

The *maximum concurrent flow* problem variant asks for the largest possible $\lambda \in \mathbb{R}$ such that it is possible to simultaneously route λD_i units of flow for each $1 \leq i \leq k$. This problem can be described as a separable LP, and so is polynomially solvable (even in strongly polynomial time [131]).

For any cut $S \subset V$, let $\mathrm{cap}(S) = \sum_{e \in \delta(S)} u(e)$ be the capacity across the cut, and $\mathrm{dem}(S)$ be the total demand crossing between S and $V \setminus S$. Then for any $S \subset V$ with $\mathrm{dem}(S) > 0$ we obviously get the upper bound

$$\lambda \leq \frac{\mathrm{cap}(S)}{\mathrm{dem}(S)}. \tag{2.8}$$

In the single commodity case ($k = 1$), the celebrated Ford-Fulkerson max-flow min-cut theorem [62] says that the maximum flow is *equal* to the minimum capacity of a cut separating the source and the sink. In other words, if $k = 1$ then there is some $S \subset V$ where (2.8) holds with equality. Unfortunately, this result is no longer true in the multicommodity setting.

A breakthrough was made by Leighton and Rao [103]. They considered the special case of *uniform* multicommodity flow, where there is a unit of demand between every pair of terminals. In this case, the upper bound one obtains on λ from some cut $S \subset V$ is exactly the *sparsity* of the cut:

$$\lambda \leq \frac{\mathrm{cap}(S)}{|S| \cdot |V \setminus S|}.$$

The cut of minimum sparsity, which gives the best upper bound of this form, is known as the *sparsest cut*. They showed that the *flow-cut gap*—the ratio between the size of the sparsest cut and the max flow—is $O(\log k)$.

This result had a number of immediate important applications, to sparsest cut and other partioning problems, VLSI layout, scheduling, and many others; see [103] for details.

Aumann and Rabani [14] and Linial, London and Rabinovich [105] independently extended the Leighton-Rao result to arbitrary multicommodity flows. Both use an important metric embedding result of Linial et al:

Theorem 2.9 ([106])**.** *Any n-point metric can be $O(\log n)$-embedded into ℓ_1 (in fact, into $\ell_1^{O(\log n)}$).*

The full multicommodity max-flow min-cut result is obtained by applying this embedding theorem to the dual of the multicommodity flow problem. This dual essentially tells us that λ can also be written as

$$\lambda = \min_d \frac{\sum_{e \in E} u(e) d(e)}{\sum_{i=1}^{k} D_i d(s_i, t_i)}, \tag{2.9}$$

where d ranges over all metrics that can be obtained as shortest-path metrics on G with some choice of edge lengths. Let d^* be a length function that achieves the minimum in (2.9). Suppose it were an ℓ_1 metric. Then it is possible to write it as a nonnegative linear combination of cut semimetrics: $d^*(u,v) = \sum_{S\subset V} \alpha_S \chi_S(u,v)$. But then

$$\begin{aligned}\lambda &= \frac{\sum_{S\subset V}\sum_{e\in E} u(e)\alpha_S\chi_S(e)}{\sum_{S\subset V}\sum_{i=1}^{k} D_i\alpha_S\chi_S(s_i,t_i)} \\ &\geq \min_{S\subset V:\alpha_S>0} \frac{\sum_{e\in E} u(e)\chi_S(e)}{\sum_{i=1}^{k} D_i\chi_S(s_i,t_i)} \\ &= \min_{S\subset V:\alpha_S>0} \frac{\mathrm{cap}(S)}{\mathrm{dem}(S)}.\end{aligned}$$

Of course, d^* will not in general be an ℓ_1-metric, but by Theorem 2.9, we can embed it into ℓ_1 with $O(\log n)$ distortion, and the above goes through with this extra factor. (A slight variation on Theorem 2.9 is needed to prove $O(\log k)$.)

The connection of the Leighton-Rao result with the topic of thesis is perhaps made clearer with the following reinterpretation. Recall that a graph $G = (V, E)$ is a *c-expander* for some constant $c > 0$ if for every $S \subset V$ with $|S| \leq |V|/2$, we have $|\delta(S)| \geq c|S|$. The existence of expanders is trivial (consider just a complete graph), but much more surprising is the existence of *constant degree* expanders. In fact, a uniformly random d-regular graph is an expander with high probability, and explicit constructions also exist; see, e.g., [85] for a survey.

We immediately see that a c-expander, with all edges having unit capacity, does not have any sparse cuts. For any $S \subset V$ with $|S| \leq n/2$,

$$\frac{|\delta(S)|}{|S| \cdot |V \setminus S|} \geq \frac{2}{n}\frac{|\delta(S)|}{|S|} \geq \frac{2c}{n}.$$

Thus by Leighton-Rao, it is possible to route demand $1/n$ between every pair simultaneously, with congestion $O(\log n)$. It follows from this also that the expander acts as a *fractional crossbar*, i.e., any matching betwen the terminals can be routed with low congestion. To route a given matching $\{(s_i, t_i) : 1 \leq i \leq k\}$, route $1/n$ from s_i to each terminal, and from there to t_i.

This is just a combination of two uniform multicomodity flow routings, and hence can be routed with logarithmic congestion. This is also the the key idea used in Valiant's load balancing scheme, discussed in §2.3.4.

We will revisit expanders in Chapter 7, where we will make use of the fact that expanders route effectively.

2.3.3 Tree embeddings

Many problems are much simpler on trees than on general graphs, and in particular the technique of dynamic programming is often available when solving problems on trees. For this reason, it would be very useful to approximate a given finite metric by a tree metric. Unfortunately, this cannot be done with sublinear distortion—consider just a single cycle.

Bartal [17] demonstrated how to overcome this problem by considering a random embedding into tree metrics.

Definition 2.10 ([17])**.** An *α-probabilistic embedding* of an n-point metric (X, d) is a distribution of metrics, each on the same groundset X, so that

(i) no metric assigned positive probability by the distribution shrinks any distance compared to d, and

(ii) if ρ is a random metric chosen according to the distribution, $\mathbb{E}(\rho(x, y)) \leq \alpha d(x, y)$ for any $x, y \in X$.

Bartal showed that there exists an $O(\log^2 n)$-probabilistic embedding over tree metrics. Moreover, Bartal showed how to sample from such a distribution in polynomial time. For many problems, this result almost automatically yields randomized polynomial algorithms with polylogarithmic approximation factor, by sampling a tree metric from this distribution and solving the problem optimally on the tree.

Subsequently, Bartal improved his result to achieve an $O(\log n \log\log n)$-probabilistic embedding [18]. Charikar et al. [39] showed that the distribution can be chosen to have only $O(n \log n)$ tree metrics, which allows for approximation algorithms based on this method to be derandomized.

Finally, Fakcharoenpohl, Rao and Talwar [57], using ideas from [37] and [55], showed how to obtain an $O(\log n)$-probabilistically approximate embedding:

Theorem 2.11 ([57])**.** *For any n-point metric, an $O(\log n)$-probabilistic embedding into tree metrics, with only $O(n \log n)$ metrics in the distribution, exists and can be found in polynomial time.*

This is tight up to constant factors [17]. See also Fakcharoenpohl et al. [56] for a high level sketch of this result, and some more details on previous work.

Notice that the result by Fakcharoenpohl et al. is in fact a strengthening of the ℓ_1 embedding result of Linial et al., since tree metrics (and hence convex combinations of tree metrics) are ℓ_1.

2.3.4 Oblivious routing

Oblivious routing, and some motivations for considering it, have already been discussed in §2.2.2. We now consider oblivious routing in a slightly different setting. We are given a undirected graph $G = (V, E)$, but in addition, capacities $u(e)$ for all edges. Unlike for the robust network design problems considered earlier, we have no control over the capacities in the network. Rather, the goal is to choose a routing that minimizes *congestion*, the maximum over all edges of the ratio between load and capacity.

As before, a routing is specified by a routing template, which gives a unit flow f_{ij} between each pair of terminals i, j. For a specific demand matrix D, The congestion of a routing is defined as the maximum over all $e \in E$ of $\ell(e)/u(e)$, where $\ell(e)$ is the total load put on an edge by the routing, and is given by

$$\ell(e) = \sum_{i,j} f_{ij}(e) D_{ij}.$$

The reason for considering the maximum load, rather than some form of average, is that it gives a good proxy for the throughput of the network; if a single edge is highly congested, all packets sent through that link will experience long delays. Other performance measures are discussed below.

For a fixed demand matrix, a fractional routing of minimal congestion can be found in strongly polynomial time [131]. However, as already discussed, often adaptive routing schemes are difficult or impossible to implement in practice. It would be very useful to be able to set up the routing in advance, independent of the specific demand that must be routed. Of course, we cannot expect to do as well as an adaptive scheme, but how well can we do?

One of the earliest results is due to Valiant [132]. Motivated by applications to parallel computing, Valiant considers a network with the topology of a hypercube, with unit capacities, and considers the problem of routing an arbitrary permutation π; each node i must send a packet to destination $\pi(i)$. Notice the similarity to the hose model—here we must be able to route any integral demand matrix where the total incoming demand and the total outgoing demand at any node is at most 1. He suggests the following two-stage scheme for routing a packet from a source u to destination v. First, pick a random intermediate node w uniformly at random, and route the packet from u to w along a shortest path, also chosen at random from the set of all shortest u-w paths. Then route from w to v along a random shortest w-v path.

When considering the aggregate of many packets sent from s_i to t_i via such a randomized scheme, we can define a unit flow f_i by taking $f_i(e)$ to be the fraction of packets that uses edge e (more formally, the expected fraction of packets). Valiant's scheme can thus be interpreted as describing a particular multipath flow template.

Valiant showed that his scheme has congestion $O(\log n)$ for routing any permutation. The result was quickly improved and simplified by Valiant and Brebner [133]. Valiant's two-stage method has prompted a variety of work in the networking community on similar approaches, e.g., [109, 38, 99, 98, 137].

Valiant's scheme is randomized; Borodin and Hopcroft [32] showed that this was a necessary feature. They proved that any deterministic (i.e., single-path) oblivious routing scheme, on any graph, will have worst-case congestion

$\Omega(\sqrt{n}/\Delta^{3/2})$ for routing all permutations, where Δ is the maximum degree. The bound was improved to $\Omega(\sqrt{n}/\Delta)$ by Kaklamanis et al. [92].

The above dealt with routing all possible permutations. A more general framework is to consider routing *any* demand—but using the minimum congestion routing possible for that demand as the benchmark. More formally: think of G as a capacitated graph with capacities given by u, and define $\lambda_{\boldsymbol{f}}(D)$ to be the congestion incurred when routing demand D via routing $\boldsymbol{f}$. Then define the optimal congestion $\lambda^*(D)$ for a demand D as the minimum congestion attainable by any routing. The *competitive ratio* of routing $\boldsymbol{f}$ is the value

$$\Lambda(\boldsymbol{f}) := \max_D \frac{\lambda_{\boldsymbol{f}}(D)}{\lambda^*(D)}, \tag{2.10}$$

where D ranges over all demand matrices where $\lambda^*(D)$ is nonzero. The *optimal oblivious routing* is the routing with minimum competitive ratio.

A close connection to robust network design can be made via the following equivalent formulation [41]. Let $\mathcal{G}$ be the universe of demands that are routable in G with edge capacities given by u (assume $u(e) > 0$ for all $e \in E$). Notice that $\Lambda(\boldsymbol{f}) = \max_{D \in \mathcal{G}} \lambda_{\boldsymbol{f}}(D)$. Writing out the definition of $\lambda_{\boldsymbol{f}}(D)$, finding an optimal oblivious routing $\boldsymbol{f}$ is equivalent to solving

$$\min_{\boldsymbol{f}} \max_{e \in E} u(e)^{-1} \max_{D \in \mathcal{G}} \sum_{i,j \in W} D_{ij} f_{ij}(e). \tag{2.11}$$

Compare this to a robust network design problem with edge costs $c(e) = u(e)^{-1}$, and universe $\mathcal{G}$:

$$\min_{\boldsymbol{f}} \sum_{e \in E} u(e)^{-1} \max_{D \in \mathcal{G}} \sum_{i,j \in W} D_{ij} f_{ij}(e). \tag{2.12}$$

The only difference is that a sum over edges has been replaced by a maximum. Note, however, that this does *not* mean that robust network design is equivalent to oblivious routing with cost measure given by the sum of edge congestions, since this also affects the denominator in (2.10); see the section below on other cost measures.

Polylog-competitive oblivious routing schemes were demonstrated for various different network topologies (see [116] for details). A breakthrough came in

2002 when Räcke demonstrated that an oblivious routing with polylogarithmic congestion is possible for arbitrary undirected graphs [114]. The basic scheme used by Räcke, first introduced by Maggs et al. [107], is based on a *decomposition tree*. A decomposition tree for a graph $G = (V, E)$ corresponds to a laminar family on V, with leaves of the tree corresponding to individual nodes and the root corresponding to all of V. Edge capacities for the tree are chosen so that any demand supported in G is supported by T_G also. By choosing a very particular decomposition tree T_G, Räcke showed that then any demand routable in this special T_G could be routed in G, with congestion only a $O(\log^3 n)$-factor larger. Each edge in T_G is essentially mapped to a carefully chosen multicommodity flow; the concatenation of these flows on all the edges of T_G between a particular pair of terminals yields the flow template for that pair in G. The full multicommodity max-flow min-cut result is an important ingredient in the proof.

Räcke's original result was existential, and did not provide an efficient way of constructing the oblivious routing. This was rectified by Azar et al. [16], who showed how to find an *optimal* oblivious routing in polynomial time. Applegate and Cohen [12] gave a compact LP formulation for the problem. We note that the problem of finding a minimum congestion oblivious routing can be phrased as a robust LP over a polyhedral universe $\mathcal{G}$, by virtue of (2.11). So these results, analagous to the results on robust network design with multipath routing, can also be seen as consequences of Theorem 2.4 and Theorem 2.6.

Harrelson et al. [83] improved the approximation ratio to $O(\log^2 n \log\log n)$. Finally, Räcke [115] demonstrated an oblivious routing with $O(\log n)$ congestion, which is asymptotically tight. The technique in this paper is different from the earlier ones in that a *distribution* of decomposition trees is used; the routing used for each tree is much simpler though. Each tree T in the distribution is endowed with a mapping $\phi_T : V(T) \to V$ of its nodes into the graph, with each leaf being mapped to its associated terminal. An edge in the tree is mapped to a shortest path between the image of its endpoints. (See §6.2, where we also use this kind of routing description.)

The embedding result of Fakcharoenpohl, Rao and Talwar plays a crucial role in Räcke's proof, and there is a close analogy between this result and the proof of Leighton-Rao by London et al. via ℓ_1 metric embedding. Note that just as the tree embedding result is a strengthening of the ℓ_1 embedding result, Räcke's result is a strengthening of the multicommodity max-flow min-cut theorem. Andersen and Feige [6] give a beautiful condensation and generalization of Räcke's result. They set up a quite general framework for which both distance-based mappings (i.e., metric embeddings) and capacity-based mappings can be defined. They then demonstrate a short duality argument that shows that results on distributions of distance-based mappings and results on distributions of capacity-based mappings are equivalent.

2.3.5 Other cost measures

Other cost measures besides congestion are possible. A particularly important one for packet networks is the sum of the congestion and the *dilation*, which is the average path length in the routing. This is important because in many packet routing protocols, this sum is a good estimate of the average time taken for a packet to traverse from origin to destination. In fact, a result of Leighton, Maggs and Rao [102] says that for any specified routing, there always exists a scheduling of packets that takes time $O(\text{congestion} + \text{dilation})$. This result was nonconstructive, but almost as good a bound can be obtained with an online algorithm [112]. Bounds obtained for this model are of course stronger than for congestion only: Valiant's result, and some others, apply in this stronger model.

Assume unit edge weights for the remainder of this section. Gupta et al. [76] consider a quite general class of cost measures. A given function $\ell : \mathbb{R}_+^k \to \mathbb{R}_+$ gives the "load" for an edge as a function of the flow: $L_e := \ell(f_1(e), f_2(e), \ldots, f_k(e))$. The cost measure is then given either as the maximum load $\max_e L_e$, generalizing the congestion measure, or as the total (equivalently average) load $\sum_e L_e$. They give polylog-competitive oblivious algorithms for the maximum load case when ℓ is a norm, and for the total load case when ℓ is monotone and subadditive.

Englert and Räcke [53] generalize this result to a larger class of aggregation functions. In the above, the maximum load measure is exactly $\|\boldsymbol{L}\|_\infty$, and the total load measure just $\|\boldsymbol{L}\|_1$. Englert and Räcke allow cost measures of the form $\|\boldsymbol{L}\|_p$ for any $p \in [1, \infty]$ (and the load function ℓ any monotonic norm). They show the existence of an oblivious routing with logarithmic competitiveness with respect to this measure; their routing scheme is again describable via a convex combination of trees. Their result can be seen as an interpolation between tree embeddings and oblivious routing; the case $p = \infty$ is clearly exactly oblivious routing with the maximum congestion measure, and the case $p = 1$ can be shown to be equivalent to the tree embedding theorem of Fakcharoenpohl et al. The Englert-Räcke result is nonconstructive however (except for the Euclidean norm $p = 2$, and the previously studied cases $p \in \{1, \infty\}$). This was recently rectified by Bhaskara and Vijayaraghavan [31].

Harsha et al. [84] consider the *average latency* measure. This corresponds to the sum over edges of the *square* of the load. This does not fall in the framework of the two papers discussed above, since it is a convex function of the load; instead, connections with electrical networks are used to deduce an $O(\log n)$ competitive oblivious algorithm for the single-sink case. The question for the multicommodity case remains open.

A more detailed survey of the results in oblivious routing is given in Räcke [116].

2.4 Network design without side constraints

Robust network design problems (especially the asymmetric VPN problem) have connections with other important network design problems. The problems we will discuss here all have a similar flavour; the goal is to satisfy some form of connectivity requirement, and there are no hard constraints except for these connectivity requirements. We will not consider problems where potential solutions are impacted by upper bounds on edge capacities. Nor will we consider "survivable" network design problems, where solutions must be resilient to various kinds of failures in the network (e.g., a link failure). Such problems are of course important and relevant to communication networks; for a survey, see [73], and the more recent [95] which also covers some more practical aspects. Also see [123] for a discussion of various types of side constraints encountered in the setting of telecommunication networks.

2.4.1 Some problem definitions

First we define the problems under consideration, and note some connections between them (see Figure 2.1).

The *Steiner tree problem* is one of the most fundamental network design problems. Given a (possibly weighted) graph $G = (V, E)$ and some set of terminals $W \subset V$, it asks for a set of edges that connects all the terminals, at minimal cost.

In the *multicommodity rent-or-buy problem* (MCROB), we are given a graph $G = (V, E)$ with edge costs $c : E \to \mathbb{R}_+$, a set of k source-sink pairs (s_i, t_i), $1 \leq i \leq k$, and a parameter $M \geq 1$. A solution must have enough capacity to concurrently route (integrally) one unit of demand between each s_i-t_i pair (more generally, one might specify a demand for each pair; this does not make the problem any more difficult however). Capacity may either be *rented* or *bought* on each edge. If capacity on edge e is rented, an amount $c(e)$ must be paid for each unit of capacity needed on the edge. If it is bought, an amount $M \cdot c(e)$ must be paid, but as much capacity as required may then be used. As usual, the goal is to find a solution of minimum cost.

A special case of this problem is the *single-sink rent-or-buy problem* (SSROB). Here, $t_i = t$ for all $1 \leq i \leq k$, where $t \in V$ is referred to as the *sink*. This special case already includes the Steiner tree problem, by considering the case $M = 1$; then every edge should be bought, and a solution must simply connect

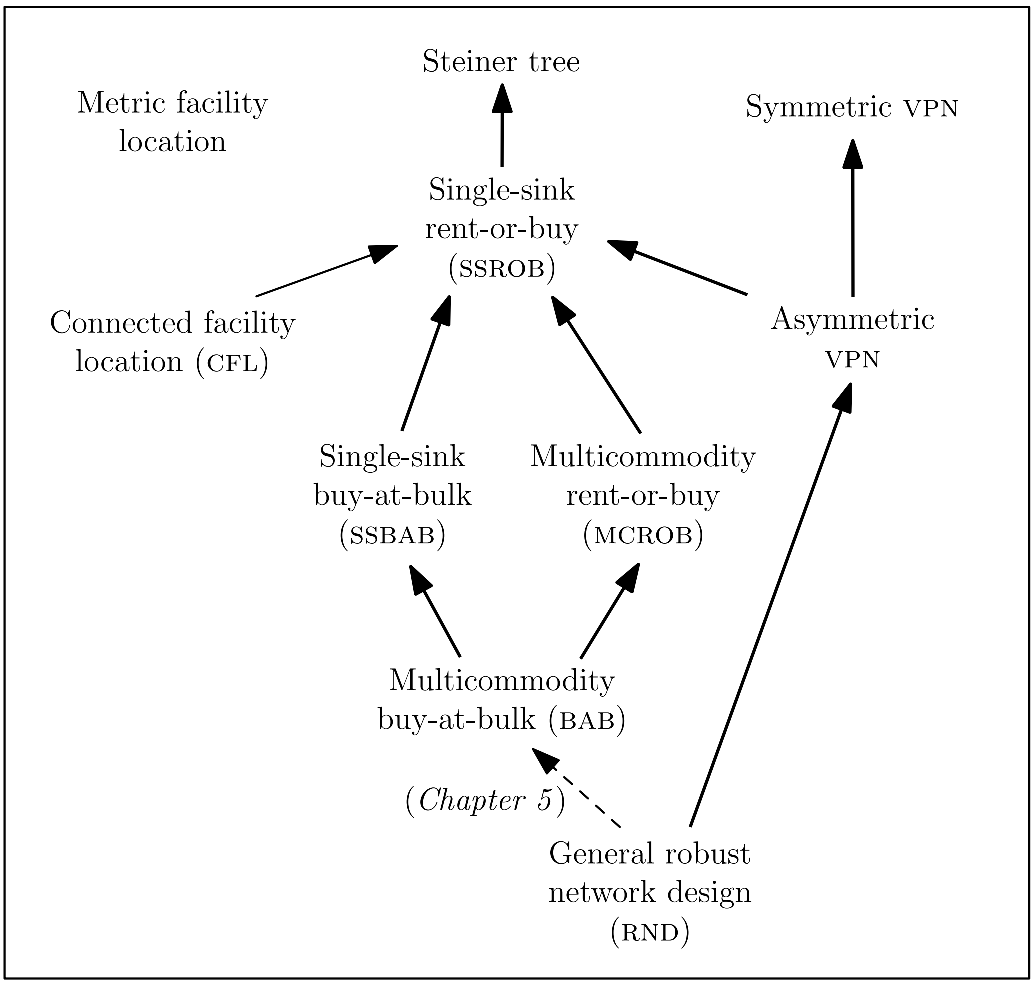

Figure 2.1: The main network design problems considered, with an arrow indicating that a problem is a special case of another.

all the sources s_i to the sink. SSROB is also a special case of the asymmetric VPN problem. To simulate an instance of SSROB with the asymmetric VPN problem, make each source s_i a terminal with hose marginals $b^+_{s_i} = 1$, $b^-_{s_i} = 0$, and make the sink node a terminal with $b^+_t = 0$, $b^-_t = M$. Then a solution to this VPN instance must be able to support routing any subset of M source terminals concurrently to the sink. In particular, the capacity required on any edge never exceeds M, and so the capacity requirement for the VPN instance exhibits the cutoff property of rent-or-buy. See Lemma 7.5 for more formality.

Single-sink rent-or-buy is also related to the well-studied class of facility location problems. There are far too many variants to even mention here (see [125] for a compact survey), and we describe just two. In the *metric uncapacitated facility location problem*, we are given a graph $G = (V, E)$, edge costs $c : E \to \mathbb{R}_+$, a set of *clients* $W \subset V$ and a set of *facilities* $F \subset V$. Each facility $j \in F$ has an *opening cost* $\varphi_j \in \mathbb{R}_+$. A solution to the problem involves opening a subset of the facilities (incurring opening costs), and connecting each client i to some open facility j (assumedly the closest one), incurring a connection cost $d(i, j)$ (with the metric determined by the edge costs c). The goal is to find a solution of minimum cost.

In the *connected facility location problem* (CFL), an extra requirement is that the opened facilities themselves must be connected. Moreover, connecting facilities is more expensive; a parameter $M > 1$ is given, and using an edge e for this purpose incurs a cost $M \cdot c(e)$. Clearly, in the optimal solution the open facilities would be connected via a minimum cost Steiner tree. This problem includes single-sink rent-or-buy as a special case. For some given SSROB instance, we construct a corresponding CFL instance as follows. Each terminals becomes a client, and every node in the network becomes a potential facility; all opening costs are zero. We also tweak the instance to ensure that the facility at the root will be opened (this could be done for instance by introducing a large number of new terminals adjacent to the root, on zero cost edges). A solution to this connected facility location instance yields a solution to the original SSROB instance of equal cost: buy all edges used to connect opened facilities, and rent all edges used to route from a terminal to an open facility. Moreover, it can be shown (Proposition 7.6) that there is always an optimal solution to SSROB where the bought edges form a tree rooted at r, completing the equivalence.

The *uniform multicommodity buy-at-bulk problem* (BAB) generalizes rent-or-buy. Here, in addition to the weighted graph G and k source-sink pairs (s_i, t_i), we are given a cost function $f : \mathbb{N} \to \mathbb{N}$, which must be increasing, concave, and satisfy $f(0) = 0$. The value of $f(x)$ represents the cost per unit distance of reserving x units of capacity on an edge (where the edge weights $c(\cdot)$ represent length). This network design problem represents a situation with economies of scale; since f is concave, reserving 2 units of capacity on a single edge may be less expensive then reserving a single unit on two separate edges. And again we may define a single-sink version of buy-at-bulk, where for some node $t \in V$, $t_i = t$ for all $i \leq k$.

There is a second alternative definition of the buy-at-bulk problem which is also used in the literature. Instead of a concave cost function f, a sequence of K "cable types" is given, each having an associated cost per unit length σ_i, and capacity u_i. As many cables as required, of any type, may be bought on an edge; however, they must be purchased integrally. It is easily seen that the definition of buy-at-bulk that we use is a special case of the cable-type definition; simply provide, for each $i \in \mathbb{N}$, a cable of capacity i and cost $f(i)$. The reverse is not true, since the cable types can introduce additional knapsack-like complications. However the problems are easily seen to be related within a factor of 2 [81], and in [60] it is shown that in the single-sink case, there is a randomized reduction from the cable type version to the concave function version which does not lose any factor.

The *non-uniform* buy-at-bulk problem is similar, except that the cost function may vary depending on the edge. We will not discuss this further here, and "buy-at-bulk" will always refer to the uniform version.

We will see in Chapter 5 that buy-at-bulk is a special case of the general robust network design problem, making it a common generalization of all the problems shown in Figure 2.1, except for the two facility location problems.

2.4.2 A survey

Steiner tree The Steiner tree problem is APX-hard, and cannot be approximated within a factor 96/95 unless P = NP [43]. It is a fundamental problem, and the approximation factors for many other network design problems depend on the approximability of this problem. Let ρ_{st} denote the best possible approximation ratio for the problem. Until very recently, the best bound was $\rho_{\mathrm{st}} \leq 1 + (\ln 3)/2 < 1.55$, due to Robins and Zelikovsky [118]. With some interesting new techniques, this has now been improved to $\rho_{\mathrm{st}} \leq \ln 4 + \epsilon < 1.39$ by Byrka et al. [36]. Their approach involves using a much stronger LP formulation, and a novel combination of iterated rounding and randomized rounding methods. The result of Byrka et al. immediately implies an improvement in the approximation guarantees claimed in many of the references discussed here; in most cases, we will state the approximation guarantees in terms of ρ_{st}.

Three closely related problems The "filtering method" of Lin and Vitter [104], which they first applied to obtain an approximation algorithm for the s-median problem (another facility location variant), has proven to be very useful. The filtering refers to a massaging of the fractional LP optimum to a more suitable form, at only a small increase in cost, before rounding. Shmoys, Tardos and Aardel [126] used this filtering technique to obtain the first constant-factor approximation algorithm for the metric uncapacitated facility location problem. The constant has been improved in a sequence of papers to the current best of slightly under 1.5 by Byrka [35].

Ravi and Salman [117] considered a slight variation on connected facility location, where the open facilities must be connected with a tour. Their result implies a constant-factor approximation algorithm for the problem. Karger and Minkoff [93] were the first to propose the exact connected facility location problem, and gave a purely combinatorial approximation algorithm. In the

2001 paper by Gupta et al. considering the VPN problem [77], they encounter the connected facility location by reducing from the asymmetric VPN problem with *tree routing*. Using the filtering approach, they give a much improved factor 12 approximation algorithm for CFL, which is improved to 10 in the case where opening facilities is free (hence in particular, SSROB).

Swamy and Kumar [129] give a primal-dual algorithm for connected facility location, and achieve an approximation factor of $7 + \rho_{\mathrm{st}}$. For the special case of single-sink rent-or-buy, their analysis yields a factor $3 + \rho_{\mathrm{st}}$.

Gupta et al. [81, 79] used a randomized sampling approach to obtain improved (and also simpler) algorithms for SSROB and asymmetric VPN, highlighting again the close connections between these problems. Their algorithms are quite simple, and their "sample-augment" framework has formed the basis for most of the subsequent work on these problems. We describe their algorithm for the asymmetric VPN problem. First, assume that the terminals have been divided into a set of *senders* S with $b_i^+ = 1$, $b_i^- = 0$, and a set of *receivers* R with $b_i^- = 1$, $b_i^+ = 0$ (this can be done without any loss of generality), and assume $|R| \geq |S|$.

Sample-augment algorithm for asymmetric VPN [81]

- Pick a sender $s \in S$ uniformly at random.
- **Sample:** mark each receiver independently at random with probability $1/|S|$; let R' be the set of marked receivers.
- **Steiner:** Construct a ρ_{st}-approximate Steiner tree T on $R' \cup \{s\}$, and install $|S|$ units of capacity on the edges of this tree.
- **Augment:** Connect all terminals to the closest node of T via shortest paths, installing one unit of capacity cumulatively for each path.

The expected costs of the "Steiner" and "augment" parts of the solution can be calculated separately, and Gupta et al. show that the total expected cost is within a factor $4 + \rho_{\mathrm{st}}$ of optimal. A similar scheme for SSROB yields a randomized $(2 + \rho_{\mathrm{st}})$-algorithm; in fact, their algorithm works for any CFL instance where opening costs are zero.

Eisenbrand and Grandoni [47] improved the algorithm and analysis to obtain a factor 4.74 randomized algorithm (better if the current best value of ρ_{st} is used in their analysis). Their sampling scheme is slightly different: unlike the algorithm above, the returned solution is not necessarily a tree routing. By exploiting Hu's 2-commodity flow theorem, Eisenbrand et al. [48] obtain a much stronger lower bound, and their tighter analysis yields a factor $2 + \rho_{\mathrm{st}}$ randomized algorithm. Another interesting feature of their approach is that an exact, but exponential time, algorithm for Steiner tree is used to improve the constant, by trading off between various different algorithms, and using an exact Steiner tree algorithm when the number of terminals is sufficiently small.

Connected facility location and single-sink rent-or-buy get some more attention from Eisenbrand et al. [49]. Their technique of "core detouring" uses the unknown optimal solution as a tool in their analysis. This allows them to consider a modification of the basic sampling algorithm where the probability of marking a node is larger than in the original Gupta et al. scheme. They obtain a 4.00 approximation factor for CFL and a 2.92 factor for SSROB; these bounds can be improved using the recent $\rho_{\mathrm{st}} < 1.39$ of Byrka et al. [36] as well as the recent uncapacitated facility location result of Byrka [35].

Williamson and Van Zuylen [135] and Van Zuylen [134] provide a method of derandomizing (via the method of conditional expectation) the sampling-based algorithms for single-sink problems (including asymmetric VPN), with some loss in the constant.

Rothvoß and Sanità [120] gave, for any $\epsilon > 0$, a $2 + \epsilon\frac{|R|}{|S|}$ approximation algorithm for the asymmetric VPN problem, where $|R| \geq |S|$ refer to the number of receivers and senders, respectively. In addition, they gave a constant factor approximation for the problem with a concave cost function.

Single-sink buy-at-bulk Andrews and Zhang [9] obtained an $O(K)$ (where K is the number of cable types) approximation algorithm for a special case of the problem they called *access network design*. Garg et al. [64] obtained the same factor for the full SSBAB problem using LP based methods. The first constant approximation factor was obtained by Guha, Meyerson and Munagala [74]. Talwar [130] showed that the natural LP formulation for the problem has a constant integrality gap, building very closely on the work by Garg et al. [64]; the analysis yields another algorithm, with a much better constant of 216.

The sample-augment framework of Gupta et al. [81, 79], applied in a heirarchical fashion, works here too and gives a signficant improvement, both in simplicity and in the constant; they obtain $16(3 + \rho_{\mathrm{st}})$. A slight variation of the algorithm and a tighter analysis yield a factor 24.92 (better using the improved bound on ρ_{st}).

Multicommodity rent-or-buy and buy-at-bulk The first approximation algorithms for multicommodity buy-at-bulk (and hence also rent-or-buy) were given by Awerbuch and Azar [15]. Using Bartal's result on tree embeddings, they gave an $O(\log^2 n)$ factor approximation algorithm (and using the Fakcharoenpohl et al. result, this immediately improves to $O(\log n)$).

Gupta et al. [80] showed that in fact an $O(1)$ approximation factor can be obtained for MCROB, again via their sample-augment framework. A simplification and improvement is given in [79]. Becchetti et al. [20] improved the algorithm and analysis to obtain a $4+2\sqrt{2}$ factor. Fleischer et al. [61] obtained the current best ratio of 5.

In contrast to the situation for rent-or-buy, Andrews [7] gave a strong negative result for multicommodity buy-at-bulk. He showed that under believed complexity assumptions, the uniform buy-at-bulk problem cannot be approximated to within a polylogarithmic factor. The paper is important for introducing techniques from probabilistically checkable proofs (PCPs) for proving hardness in *undirected* network design problems. See for example

[10, 11] for more hardness results based on these methods. This result will be important for the contents of Chapter 5, and will be discussed further there.

Simultaneously good approximations Khuller et al. [97] showed that for any graph, and set of terminals with a specified root, there always exists a tree which *simultaneously* well-approximates both the optimal Steiner tree on these terminals, and a shortest path tree from the terminals to the selected root. More exactly, they show that for any instance, there exists a tree with cost within a factor $1 + \sqrt{2}$ of the optimal Steiner tree, and such that the path between any terminal and the root in this tree is at most $1 + \sqrt{2}$ times the shortest-path distance. This simultaneous approximation result has proven to be very useful as a subroutine in other algorithms (e.g., some of the algorithms for SSBAB [130]).

Goel and Estrin [65] consider the following generalization. We are given an instance of single-sink buy-at-bulk, but we are not told the concave cost function f. The goal is to find a single solution which is good, irrespective of the actual function f; in other words, for any choice of f, the ratio of the cost of our solution (which depends on f, even though our solution does not) to the minimal cost solution tailored to this choice of f, should be as small as possible. Goel and Estrin showed that in polynomial time, a tree can be found for which this ratio is at most $1 + \log k$, where k is the number of source terminals. The result of Khuller et al. interpreted in this setting shows that if f is unknown, but restricted to be either a constant function or a linear function, then a tree exists (and can be found) for which this ratio is constant.

Goel and Post [67] very recently showed that, rather surprisingly, there always exists a tree which exhibits a constant ratio over *all* choices of valid cost functions (and such a tree can be found in polynomial time). This builds on their earlier result [66] showing how to construct a distribution of trees so that the expected cost is always within a constant factor of optimal.

The previously mentioned work by Gupta et al. [76], when considering the total cost measure, is also relevant to this line of work. In this context, Gupta et al. give a routing which is polylog competitive when both the concave cost function *and* the demand requirements for each source are unknown (see also Srinivasagopalan et al. [128]).[6] Their result also applies to the multicommodity case. See also Jia et al. [89] for a general framework on "universal" approximation schemes.

[6]Our discussion has assumed unit demands throughout, but the generalization to arbitrary but fixed demands is obvious.

Chapter 3

The VPN Conjecture

3.1 Introduction

We begin by refreshing the definition of the VPN problem and VPN Conjecture, and discuss previous and related work. In §3.2 we discuss the "pyramidal routing" of Grandoni et al. [71], a crucial component of our proof. Our resolution of the Pyramidal Routing Conjecture (and hence the VPN Conjecture) is given in §3.3. Finally, in §3.4 we briefly discuss a slightly different definition of the symmetric hose model which is sometimes used, and observe that the conjecture holds there also.

3.1.1 The VPN problem

The problem under consideration in this chapter is robust network design under the symmetric hose model, with single-path routing. This is defined in Chapter 1, and motivated and discussed thoroughly in §2.2.1. Here, we give a very brief self-contained description of the problem.

We are given an undirected graph $G = (V, E)$. Each edge $e \in E$ has an associated nonnegative cost $c(e)$, representing the cost per unit of bandwidth on that edge. We will use $d(\cdot, \cdot)$ to denote the shortest-path metric induced by these weights. In addition, a subset $W \subset V$ of *terminals* is given; these are the entities which the VPN must connect. Let $k = |W|$. Each terminal $i \in W$

has an associated *marginal* or *hose capacity* b_i. By scaling if necessary, we will assume that $b_i \in \mathbb{N}$ from now on (see [86]). The universe of feasible demands is defined as

$$\mathcal{H} = \{D \in \mathbb{R}_+^{\binom{k}{2}} : \textstyle\sum_j D_{ij} \leq b_i \ \forall\, i \in W\}.$$

This information defines the VPN problem instance; we may use the shorthand $(G, W, \boldsymbol{b})$ to describe a particular instance.

A solution is given by a routing template $\mathcal{P}$. Since we are considering single-path routing, this is given as $\mathcal{P} = \{P_{ij} : i, j \in W\}$, where P_{ij} is an i-j-path for every $i, j \in W$, and $P_{ij} = P_{ji}$.

The cost of the solution, which we seek to minimize, is given by $C_{\text{VPN}}(\mathcal{P}) := \sum_{e \in E} c(e)u(e)$, where

$$u(e) := \max_{D \in \mathcal{H}} \sum_{i<j \in W : e \in P_{ij}} D_{ij}.$$

3.1.2 The VPN Conjecture

Fingerhut et al. [58] in 1997 proposed the following simple algorithm, and demonstrated that it is a 2-approximation. It was rediscovered by Gupta et al. [77] in 2001. The proof of the approximation ratio is deferred to the next chapter, since details of the proof will be important there.

Definition 3.1. A *shortest path tree solution* (or *VPN tree solution*) with root $r \in V$ has routing template $\mathcal{R}^r = \{R^r_{ij} : i, j \in W\}$, where R^r_{ij} consists of a shortest path from i to r, followed by a shortest path from r to j.

Note that in the above definition, R^r_{ij} could be a walk rather than a simple path. If preferred, the obvious shortcutting procedure can be used to obtain simple paths (which no longer necessarily pass through r). This will not decrease the cost of the solution however, and in some applications it may be useful to route all flow through a single hub node; in particular, routing decisions need only be made at r (see also §6.2.1).

The algorithm simply returns $\mathcal{R}^t$, where $t \in V$ is chosen such that $C(\mathcal{R}^t)$ is minimal. This is clearly computable in polynomial time, with nk shortest path calculations.

Theorem 3.2 ([58, 77]). *This algorithm is a* 2*-approximation algorithm. Moreover, it returns a solution of cost at most twice that of the optimal solution with* dynamic *routing.*

Proof. Note that the cost of solution $\mathcal{R}^t$ is at most $\sum_{i \in W} d(i,t) b_i$, since the required capacity on the edges of the path from any terminal i to t is at most b_i.

Let $B := \sum_{i \in W} b_i$. Let OPT be the cost of the optimal solution; it will in fact not matter whether single-path, multipath or dynamic routing is used. Consider the single demand matrix $\bar{D}$ defined by

$$\bar{D}_{ij} = \begin{cases} \frac{b_i b_j}{B} & \text{if } i \neq j \\ 0 & \text{otherwise} \end{cases}.$$

We have that for any $i \in W$,

$$\sum_{j \in W} \bar{D}_{ij} = \frac{b_i}{B} \sum_{j \neq i} b_j = b_i \frac{B - b_i}{B} \leq b_i. \tag{3.1}$$

Hence $\bar{D} \in \mathcal{H}$, and so $\text{OPT} \geq C^*(\bar{D})$, the cost to optimally route just the single demand matrix $\bar{D}$; note that this is true even when dynamic routing is allowed. So we have

$$\text{OPT} \geq \sum_{i<j \in W} d(i,j) \bar{D}_{ij} = \tfrac{1}{2} \sum_{i \in W} \sum_{j \in W} d(i,j) \frac{b_i b_j}{B}. \tag{3.2}$$

Now consider the following weighted average over the costs of shortest path tree solutions, with roots in W:

$$\begin{aligned} \frac{1}{B} \sum_{j \in W} b_j C(\mathcal{R}^j) &\leq \frac{1}{B} \sum_{j \in W} b_j \sum_{i \in W} d(i,j) b_i \\ &\leq 2 \cdot \text{OPT} \qquad\qquad \text{by (3.2)}. \end{aligned}$$

Since the cheapest $\mathcal{R}^j$ for $j \in W$ is at most this average, the result follows. □

The proof can be easily modified to give the slightly stronger bound $2(1 - b/B)$, where $b = \min_i b_i$ and $B = \max_i b_i$. Simply replace $\bar{D}$ throughout by the matrix $\hat{D} = (1-b/B)^{-1}\bar{D}$; we will still have by (3.1) that $\hat{D} \in \mathcal{H}$. In particular, if $b_i = 1$ for all $i \in W$, it shows that the algorithm is a $2(1 - \frac{1}{k})$-approximation algorithm.

As shown in [101, 77], this algorithm in fact returns the optimal solution using tree routing. The following conjecture then says that this algorithm is optimal even for single-path routing, making the VPN problem polynomially solvable:

VPN Conjecture ([2, 54, 78, 87])**.** *For any symmetric VPN problem instance, there exists an optimal single-path routing solution in the form of a tree.*

Before this work, the conjecture had been established only for the following cases:

(i) when the graph is a tree (this is trivial),

(ii) on a complete graph with all edge lengths equal [58, 86],

(iii) on a graph with at most 4 nodes [86],

(iv) when the graph is a ring network, i.e., a single cycle [86], and

(v) on graphs built up from the above via taking 1-*sums* [86]; this is simply the gluing together of two graphs by taking the disjoint union, but identifying one node from each.

Hurkens et al. actually proved thiat in all of the above cases, a stronger result holds; even if multipath routing is allowed, there is always an optimal tree solution in any of the above cases.[1] They do this by explicitly constructing an LP solution, and a matching dual solution certifying its optimality.

Simultaneously with this work, Fiorini et al. [59] proved (via completely different methods) that the conjecture holds for outerplanar graphs.[2]

[1] They also conjectured this to be true in general; cf. [54].

[2] An outerplanar graph is one that can be embedded as a planar map with all nodes adjacent to the outer face.

Grandoni, Kaibel, Oriolo and Skutella [71] gave a much shorter proof of the VPN Conjecture on ring networks. Their method is to define a new problem, *pyramidal routing*, and reduce the VPN Conjecture to a new one involving this routing scheme. They then give a proof of this new Pyramidal Routing Conjecture for ring networks. Their result is crucial for our proof, since we proceed by solving the Pyramidal Routing Conjecture on general networks. (In fact, the Pyramidal Routing Conjecture is equivalent to the VPN Conjecture, as we show in §3.2.) We describe their results as part of §3.2.

We prove that the VPN Conjecture is true for arbitrary instances:

Theorem 3.3. *For any symmetric VPN problem instance, there exists an optimal single-path routing solution in the form of a tree.*

This result also settles the complexity of the single path VPN problem, since the optimal VPN tree can easily be computed in polynomial time.

Making use of our results, Fiorini et al. [60] proved that tree routing remains optimal if costs are not linear, but given by an increasing concave cost function (as in the uniform buy-at-bulk problem).

3.1.3 Reducing to unit marginals

As was noted in [58, 86], it is sufficient to consider unit marginals. We reproduce the proof here.

Lemma 3.4 ([58, 86])**.** *If the VPN Conjecture holds for instances with unit marginals ($b_i = 1$ for all $i \in W$), then it holds for all instances.*

Proof. Begin with an arbitrary VPN instance $(G, W, \boldsymbol{b})$. We now construct a new graph G' as follows. For each terminal $i \in W$, we add b_i pendant nodes adjacent to i (see Figure 3.1); the added edges have zero cost. We say that these new nodes are copied from i. The set of terminals W' in the new instance consists of exactly the new nodes we added, and each of these has marginal 1 in the new instance.

Let $\mathcal{P} = \{P_{ij} : i, j \in W\}$ be any solution template to the original instance. This induces a solution template $\mathcal{Q} = \{Q_{i'j'} : i', j' \in W'\}$ to the new instance in the obvious way: for any $i', j' \in W'$, which are copies of $i, j \in W$ respectively, let $Q_{i'j'} = \{i', i\} \cup P_{ij} \cup \{j, j'\}$. Then notice that the cost of routing $\mathcal{Q}$ in the new instance is the same as the cost of routing $\mathcal{P}$ in the old one. For take any edge $e \in E(G)$, and any demand matrix D that respects the hose constraints induced by $\boldsymbol{b}$. Then define the demand matrix D' between terminals in W' by $D_{i'j'} = D_{ij}/(b_i b_j)$ for each pair $i', j' \in W'$ with i' a copy of i, j' a copy of j, and $i \neq j$; all other entries of D' are zero. Then D' is feasible for the new instance, and puts the same load on edge e. Conversely, given a feasible D' for the new instance, we may define D by

$$D_{ij} = \sum_{i' \text{ copy of } i} \sum_{j' \text{ copy of } j} D_{i'j'}.$$

Then D is feasible for the original instance, and again puts the same load on e.

Assuming the VPN Conjecture for the unit marginal case, it follows that there exists an optimal solution $\mathcal{Q}^* = \{Q_{i'j'} : i', j' \in W'\}$ to the new instance in the form of a tree. Let T^* be such an optimal tree (so $Q_{i'j'}$ is exactly the unique path between i' and j' in T^*). But consider any $i, j \in W$. For *any* copies i' of i and j' of j, the route between i' and j' in the solution template to the new instance will be the same. This follows since the solution paths all pass through i and j, and there must be a unique path in T^* between i and j. Thus we can unambiguously define a solution template to the original instance based on T^* (see Figure 3.1). This has the same cost as solution $\mathcal{Q}^*$ for the new instance, and so must be optimal: for if there was a cheaper solution to the original instance, it would induce a cheaper solution to the new instance, contradicting the optimality of $\mathcal{Q}^*$. □

We will assume unit marginals from this point forward.

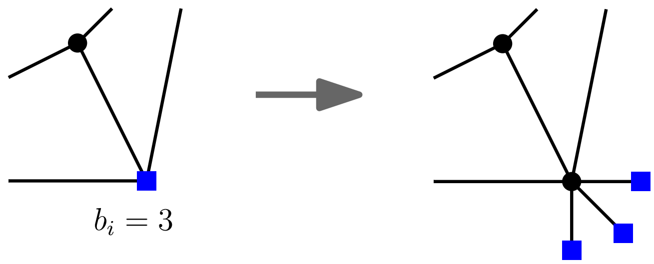

Figure 3.1: Reducing to the case of unit marginals.

3.2 Pyramidal routing

A *pyramidal routing* (PR) problem instance, as introduced in [71], is also defined by an undirected graph G with costs, and a set of terminals W; in addition, one node $t \in W$ is specified as the root. A routing template consists of a set $\mathcal{P}_t$ of simple i-t paths P_{it}, one for each $i \in W \setminus \{t\}$. Define $\ell(e, \mathcal{P}_t)$ to be the total flow through edge e, i.e., $\ell(e, \mathcal{P}_t) := |\{i \in W : e \in P_{it}\}|$. The bandwidth requirement $y(e, \mathcal{P}_t)$ is instead given by the function

$$y(e, \mathcal{P}_t) := \min\{\ell(e, \mathcal{P}_t), k - \ell(e, \mathcal{P}_t)\},$$

where recall $k = |W|$. When it is clear from the context, we may omit the routing template and just write, e.g., $y(e)$. Note that the function $y(e)$ can be viewed as a concave function of $\ell(e)$; the pyramidal shape of this dependence gives this problem its name (see Figure 3.2). The total cost is then

$$C_{\text{PR}}(\mathcal{P}_t) := \sum_{e \in E} c(e) y(e, \mathcal{P}_t).$$

The non-monotone nature of this cost is quite unusual; an edge which is used very heavily, by most of the terminals, is quite cheap!

We can also define an analogous fractional version of the pyramidal problem, where instead of paths P_{it}, the routing template consists of a set of unit i-t flows f_{it}.

Grandoni et al. [71] show that the VPN Conjecture is implied by the following conjecture:

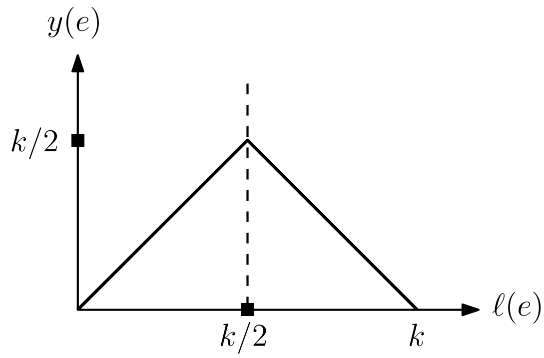

Figure 3.2: Non-monotone dependence of $y(e)$ on the load $\ell(e)$.

Pyramidal Routing Conjecture. *For every integral pyramidal routing instance, there exists a minimal cost solution where the associated routing template is a tree.*

A crucial part of their argument is the following lemma, the proof of which we reproduce below. Their proof is a slight extension of one given by Gupta et al.; cf. [77, Theorem 3.1].

Lemma 3.5 ([71])**.** *Given an* SPR *instance, and a routing template* $\mathcal{P} = \{P_{ij} : i \neq j \in W\}$*, there exists a terminal* $t \in W$ *so that* $C_{\text{VPN}}(\mathcal{P}) \geq C_{\text{PR}}(\mathcal{P}_t)$*, where* $\mathcal{P}_t = \{P_{it} : i \in W \setminus \{t\}\}$*.*

Proof. The strategy of the proof is to derive a lower bound for $u(e)$ for each e in the instance by judiciously selecting a demand matrix, which will then give us the desired lower bound on $C_{\text{VPN}}(\mathcal{P})$. Fix an edge e. The choice D^e of demand matrix for e is given by

$$D^e_{ij} := \begin{cases} \frac{1}{k}\left(\frac{y(e,\mathcal{P}_i)}{\ell(e,\mathcal{P}_i)} + \frac{y(e,\mathcal{P}_j)}{\ell(e,\mathcal{P}_j)}\right) & \text{if } e \in P_{ij}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.3}$$

Claim 3.6. *For all edges* e*,* D^e *is a valid hose demand matrix.*

Proof. We need to show that $\sum_{j \in W} D_{ij}^e \leq 1$ for all $i \in W$. We have

$$\begin{aligned}
\sum_{j \in W} D_{ij}^e &= \sum_{j \in W : e \in P_{ij}} \frac{1}{k} \left(\frac{y(e, \mathcal{P}_i)}{\ell(e, \mathcal{P}_i)} + \frac{y(e, \mathcal{P}_j)}{\ell(e, \mathcal{P}_j)} \right) \\
&\leq \sum_{j \in W : e \in P_{ij}} \frac{1}{k} \left(\frac{k - \ell(e, \mathcal{P}_i)}{\ell(e, \mathcal{P}_i)} + \frac{\ell(e, \mathcal{P}_j)}{\ell(e, \mathcal{P}_j)} \right) \\
&= \sum_{j \in W : e \in P_{ij}} \frac{1}{k} \cdot \frac{k}{\ell(e, \mathcal{P}_i)} \\
&= 1. \qquad \square
\end{aligned}$$

Claim 3.7. *For every edge e we have*

$$u(e) \geq \frac{1}{k} \sum_{i \in W} y(e, \mathcal{P}_i). \tag{3.4}$$

Proof. The claim follows from the definitions of D_{ij}^e and $\ell(e, \mathcal{P}_i) = |\{j \in W : e \in P_{ij}\}|$.

$$\begin{aligned}
u(e) &\geq \sum_{i < j \in W : e \in P_{ij}} \frac{1}{k} \left(\frac{y(e, \mathcal{P}_i)}{\ell(e, \mathcal{P}_i)} + \frac{y(e, \mathcal{P}_j)}{\ell(e, \mathcal{P}_j)} \right) \\
&= \sum_{i < j \in W} \mathbb{1}_{e \in P_{ij}} \cdot \frac{1}{k} \cdot \frac{y(e, \mathcal{P}_i)}{\ell(e, \mathcal{P}_i)} + \sum_{i > j \in W} \mathbb{1}_{e \in P_{ji}} \cdot \frac{1}{k} \cdot \frac{y(e, \mathcal{P}_i)}{\ell(e, \mathcal{P}_i)} \\
&= \sum_{i \in W} \sum_{j \in W} \mathbb{1}_{e \in P_{ij}} \cdot \frac{1}{k} \cdot \frac{y(e, \mathcal{P}_i)}{\ell(e, \mathcal{P}_i)} \\
&= \frac{1}{k} \sum_{i \in W} y(e, \mathcal{P}_i). \qquad \square
\end{aligned}$$

The lemma now follows by multiplying the inequality in (3.4) by $c(e)$ and summing over all $e \in E$:

$$\begin{aligned}
\sum_{e \in E} c(e) u(e) &\geq \sum_{e \in E} c(e) \frac{1}{k} \sum_{i \in W} y(e, \mathcal{P}_i) \\
&= \frac{1}{k} \sum_{i \in W} \sum_{e \in E} c(e) y(e, \mathcal{P}_i) \\
&\geq \min_{i \in W} \sum_{e \in E} c(e) y(e, \mathcal{P}_i). \qquad \square
\end{aligned}$$

The pyramidal routing problem has some interesting features that often make it more pleasant to work with. One such is the following.

Lemma 3.8. *There exists an optimal solution to a fractional pyramidal routing instance that is integral, i.e., f_{it} is an i-t-path for all $i \in W \setminus \{t\}$.*

Proof. We show this by proving that the problem consists of minimizing a concave function over a 0-1 polytope $\mathcal{B}$. A routing template $\mathcal{P}_t = \{f_{it} : i \in W\}$ induces a vector $\boldsymbol{x}^{\mathcal{P}_t}$ in $\mathbb{R}^{k\times|E|}$, by setting $x^{\mathcal{P}_t}_{i,e} = f_i(e)$. Let the polytope $\mathcal{B} \subset \mathbb{R}^{k\times|E|}$ be the set of vectors induced by feasible solution templates. By the max-flow min-cut theorem, the extreme points of $\mathcal{B}$ are 0-1 vectors.

The objective function is $C_{\text{PR}}(\mathcal{P}) = \sum_{e\in E} c(e)y(e,\mathcal{P}_t)$. As noted before, $y(e,\mathcal{P}_t)$ is a concave function in the load on e. Since the load is a linear function of the routing template (thought of as a vector $\boldsymbol{f} = (f_{it})_{i\in W}$ of flows), $y(e,\mathcal{P}_t)$ is concave over $\mathcal{B}$; since the sum of concave functions remains concave, $C_{\text{PR}}(\mathcal{P})$ is too. It is well known that a minimizer of a concave objective always exists at a vertex of the polytope, which corresponds to an integral routing template. □

We have seen that the VPN cost is lower bounded by some pyramidal routing cost. In fact, a converse result holds too; the cost of optimal SPR solutions can also be upper bounded by the cost of an associated pyramidal routing problem. To do this, for each solution $\mathcal{P}_t$ to a PR problem instance with root t, we define an oblivious routing template, called the *truncated hub template* associated with $\mathcal{P}_t$. This is defined as the template $\mathcal{Q} = \{Q_{ij} : i,j \in W\}$, where Q_{ij} is any i-j-path in the component of $P_{it} \triangle P_{jt}$ (where $\triangle$ denotes symmetric difference) containing i and j. Note that since i and j are the only odd-degree nodes in $P_{it} \triangle P_{jt}$, they will indeed be in the same component.

Lemma 3.9. *Given a solution $\mathcal{P}_t$ to a PR problem instance with root t, the capacity on any edge e required by its truncated hub template $\mathcal{Q}$ is at most $y(e,\mathcal{P}_t)$. In particular, $C_{\text{VPN}}(\mathcal{Q}) \le C_{\text{PR}}(\mathcal{P}_t)$.*

Proof. Let $D \in \mathcal{H}$ be any valid demand matrix. Consider any edge e and define the set of nodes which route through e by $R_e := \{i \in W : e \in P_{it}\}$. Note that $\ell(e, \mathcal{P}_t) = |R_e|$.

Now notice that if we have a pair $i, j \in W$ where $e \in Q_{ij}$, then exactly one of i and j is in R_e, because of the symmetric difference construction. So we have

$$\begin{aligned}\sum_{\{i,j\}:e\in Q_{ij}} D_{ij} &= \sum_{i\in R_e}\sum_{j\notin R_e} D_{ij} \\ &\leq \sum_{i\in R_e}\sum_{j\in W} D_{ij} \\ &\leq \sum_{i\in R_e} 1 \\ &= |R_e|.\end{aligned}$$

Similarly,

$$\sum_{\{i,j\}:e\in Q_{ij}} D_{ij} = \sum_{j\notin R_e}\sum_{i\in R_e} D_{ij} \leq \sum_{j\notin R_e} 1 = |W \setminus R_e|.$$

Thus we have that

$$\sum_{\{i,j\}:e\in Q_{ij}} D_{ij} \leq \min\{\ell(e, \mathcal{P}_t), k - \ell(e, \mathcal{P}_t)\} = y(e, \mathcal{P}_t).$$

But then the required capacity on edge e is

$$u(e) = \max_{D\in\mathcal{H}} \sum_{\{i,j\}:e\in Q_{ij}} D_{ij} \leq y(e, \mathcal{P}_t),$$

as required. □

Note that by Lemma 3.5 the optimal SPR cost is at least a convex combination of costs $C_{\text{PR}}(\mathcal{P}_t)$. Lemma 3.9 shows that it is also at most the cost of any given optimal PR solution. Thus we have the following.

Theorem 3.10. *For any pair $t, t' \in W$, the optimal solutions to the PR problems with root t and t' are the same, and have the same value as the* *optimal solution of the associated* SPR *problem. Hence, the* PR *Conjecture and the* VPN *Conjecture are equivalent.*

It should be noted, however, that no such correspondence holds between the MPR problem and the fractional PR problem.

3.3 Proof of the Pyramidal Routing Conjecture

In this section, we give a proof of the Pyramidal Routing Conjecture, thus proving Theorem 3.3. The first step will be a reduction, via a certain cost-sharing scheme, to a problem involving T-joins:

Definition 3.11. Given a graph $G = (V, E)$ and any subset $T \subseteq V$ of even cardinality, a *T-join* of G is a set $J \subset E$ of edges such that the odd degree nodes in the subgraph defined by J is precisely T. A *T-cut* is a subset $S \subset V$ such that $|S \cap T|$ is odd.

T-joins are very well understood combinatorial objects, and we are able to resolve this question in §3.3.2 using some uncrossing techniques. András Sebő [122] subsequently gave a nice shorter proof of this T-join result, which we include in §3.3.3.

3.3.1 A reduction to T-joins

Begin with an instance of the pyramidal routing problem, with root t. Let $\mathcal{P}_t$ be a routing template for this instance.

Definition 3.12. Call an edge $e \in E$ *heavy* if $\ell(e) \geq k/2$.

Let H be the set of heavy edges determined by $\mathcal{P}_t$. Note that

$$y(e) = \begin{cases} \ell(e) & \text{if } e \notin H \\ k - \ell(e) & \text{if } e \in H. \end{cases}$$

Let T' be the set of odd degree vertices in the subgraph induced by H. Now define $T = T' \triangle \{t\}$, and $T_u := T \triangle \{u\}$ for all $u \in W$. Note that $|T'|$ is even, so $|T|$ is odd, and so $|T_u|$ is even for all $u \in W$.

Let M_u be the minimum cost T_u-join on G. $C(M_u) := \sum_{e \in M_u} c(e)$ is defined to be the cost of M_u.

Define $C'(u) := C(P_{ut} \triangle H)$. This we think of as being u's contribution to the total cost of the pyramidal routing. Notice that u pays for light edges on its path, and heavy edges not on its path. We also have

$$\begin{aligned}\sum_{u \in W} C'(u) &= \sum_{u \in W} \sum_{e \in P_{ut} \triangle H} c(e) \\ &= \sum_{e \in E \setminus H} \ell(e) \cdot c(e) + \sum_{e \in H} (k - \ell(e)) c(e) \\ &= C_{\text{PR}}(\mathcal{P}_t).\end{aligned}$$

So this really is a division of the total cost between the terminals.

Theorem 3.13. *A lowerbound for the cost $C_{\text{PR}}(\mathcal{P}_t)$ of solution template $\mathcal{P}_t$ is $\sum_{u \in W} C(M_u)$.*

Proof. We want to show $C'(u) \geq C(M_u)$ for all $u \in W$. Consider the symmetric difference $H_u := P_{ut} \triangle H$. Since P_{ut} has even degree at every node except u and t, and H is a T'-join, it follows that H_u is a T_u-join. So $C(H_u) \geq C(M_u)$. But by definition, $C'(u) = C(H_u)$. Thus

$$C_{\text{PR}}(\mathcal{P}_t) = \sum_{u \in W} C'(u) = \sum_{u \in W} C(H_u) \geq \sum_{u \in W} C(M_u).$$

$\square$

Note that the right hand side of this inequality depends on H only via T. We also have the following pleasant result. Define the truncated template $\mathcal{Q} = \{Q_{uv} : u, v \in W\}$, where Q_{uv} is any u-v path contained within the component of $M_u \triangle M_v$ containing u and v. Such a component must exist, because u and v are the only odd-degree nodes in $M_u \triangle M_v$. Then:

Theorem 3.14. *The truncated template $\mathcal{Q}$ satisfies*

$$C_{\text{VPN}}(\mathcal{Q}) \leq \sum_{u \in W} C(M_u).$$

Proof. The proof is very similar to the proof of Lemma 3.9. Consider any edge e. Let $R_e = \{u : e \in M_u\}$. If $e \in Q_{uv}$, exactly one of u and v must be in R_e. Consider any feasible demand matrix D. Then

$$\sum_{\{u,v\}: e \in Q_{uv}} D_{uv} \leq \sum_{u \in R_e} \sum_{v \in W \setminus R_e} D_{uv} \leq \sum_{u \in R_e} \sum_{v \in W} D_{uv} \leq |R_e|.$$

Thus $u_{\mathcal{Q}}(e) \leq |R_e|$, and so the total cost satisfies

$$\begin{aligned} \sum_{e \in E} u_{\mathcal{Q}}(e) c(e) &\leq \sum_{e \in E} |R_e| c(e) \\ &= \sum_{e \in E} \sum_{u : e \in M_u} c(e) \\ &= \sum_{u \in W} \sum_{e \in M_u} c(e) \\ &= \sum_{u \in W} C(M_u) \end{aligned}$$

□

Note that if we had $T = \{t\}$, then M_u is a shortest path from u to t. So if t is the centre of the minimum cost VPN tree, $\sum_{u \in W} C(M_u)$ is exactly the cost of the optimal tree solution. In the next section, we show that no other choice of T improves upon this.

3.3.2 A T-join inequality

For each node v in G, define $C_{\text{SP}}(v)$ to be the cost of the VPN tree from the terminals to v, i.e.,

$$C_{\text{SP}}(v) = \sum_{u \in W} d(u, v),$$

where $d(u, v)$ is the shortest path distance between u and v according to edge weights c.

We prove the following inequality, which in turn proves the Pyramidal Routing Conjecture by the reduction in the previous section:

Theorem 3.15. *There exists a node $v \in V$ so that*

$$\sum_{u \in W} C(M_u) \geq C_{\text{SP}}(v).$$

In fact, we prove the following slightly stronger theorem:

Theorem 3.16. *Let F be the multigraph obtained by taking the disjoint union of the M_u's. Then there exists a node $v \in V$ so that there are edge-disjoint paths in F from all the vertices in W to v.*

Let T be an arbitrary subset of V of odd cardinality, induced by some routing template as in the previous section. Call a set $S \subseteq V$ T*-even* (respectively T*-odd*) if $|S \cap T|$ is even (respectively odd). Note that since $|T|$ is odd, exactly one of S and $V \setminus S$ is T-even for any $S \subseteq V$.

We will need the following lemma:

Lemma 3.17. *For any set $S \subseteq V$ which is T-even, $|\delta_F(S)| \geq |S \cap W|$.*

Proof. Consider any $u \in S \cap W$. Since S is T-even, it is T_u-odd, and so M_u must intersect $\delta_F(S)$. □

Proof of Theorem 3.16. Construct the graph F' from F by adding an extra node s, and edges su for all $u \in W$. The statement of the theorem is equivalent to showing that there exists a node v so that there is an s-v flow of size k on F', taking all the edges to have unit capacity.

For $z \in V$, define D_z to be the side of a minimum s-z cut containing z. Suppose for a contradiction that $|\delta_{F'}(D_z)| < k$ for all $z \in V$, since otherwise we have a valid routing by the max-flow min-cut theorem. Since $|\delta_{F'}(D_z)| = |\delta_F(D_z)| + |D_z \cap W|$, this gives

$$|\delta_F(D_z)| < k - |D_z \cap W| \qquad \forall z \in V. \tag{3.5}$$

Note that D_z is T-even, since otherwise $V \setminus D_z$ would be T-even, which, by Lemma 3.17, would imply

$$|\delta_F(D_z)| = |\delta_F(V \setminus D_z)| \geq |W \setminus D_z| = k - |D_z \cap W|,$$

contradicting (3.5).

Suppose inductively that all intersections of less than l sets, each of type D_z for some $z \in V$ are T-even. This is true for $l = 2$, since D_z is T-even for all $z \in V$. Now suppose for a contradiction that for some arbitrary collection $D_1, \ldots, D_l$, the set $D_1 \cap D_2 \cdots \cap D_l$ is T-odd. Let $D'_i = D_i \setminus (\cup_{j \neq i} D_j)$. It follows from the inclusion-exclusion principle, and our inductive assumption, that

Claim 3.18. *Under the assumptions in the preceding paragraph, D'_i is T-odd for all i.*

Proof. Note that $D'_i = D_i \setminus (\cup_{j \neq i}(D_j \cap D_i))$. Now by the inclusion-exclusion principle, we have

$$\begin{aligned}\Big|\bigcup_{j\neq i}(D_j \cap D_i) \cap T\Big| = \sum_{j \neq i} \Big|(D_j \cap D_i) \cap T\Big| \\ - \sum_{j_1 < j_2 : j_1, j_2 \neq i} \Big|(D_{j_1} \cap D_{j_2} \cap D_i) \cap T\Big| + \cdots \\ \cdots + (-1)^{l-1} \Big|(D_1 \cap D_2 \cap \ldots D_l) \cap T\Big|.\end{aligned}$$

The last term on the right is odd and the rest are even by our assumptions, and thus $\cup_{j \neq i}(D_j \cap D_i)$ is T-odd; since D_i is T-even, it follows that D'_i is T-odd. □

Claim 3.19.

$$\sum_{i=1}^{l} |\delta_F(D_i)| \geq \sum_{i=1}^{l} |\delta_F(D'_i)|$$

Proof. Consider any edge e that contributes to the right hand side. If it has endpoints in D'_i and D'_j for some $i \neq j$, then it will contribute twice to the right hand side; such an edge will also contribute at least twice to the left hand side, in $\delta_F(D_i)$ and $\delta_F(D_j)$. If e has an endpoint in D'_i only, and not in any other D'_j, then it is counted once on the right hand side, and at least once on the left hand side. $\square$

Now we have

$$\begin{aligned}
\sum_{i=1}^{l} |\delta_F(D_i)| &\geq \sum_{i=1}^{l} |\delta_F(D'_i)| && \text{by Claim 3.19} \\
&= \sum_{i=1}^{l} |\delta_F(V \setminus D'_i)| \\
&\geq \sum_{i=1}^{l} (k - |D'_i \cap W|) && \text{by Lemma 3.17, as } V \setminus D'_i \text{ is } T\text{-even} \\
&\geq \sum_{i=1}^{l} (k - |D_i \cap W|) && \text{as } D'_i \subseteq D_i.
\end{aligned}$$

But this is a contradiction because (3.5) implies

$$\sum_{i=1}^{l} |\delta_F(D_i)| < \sum_{i=1}^{l} (k - |D_i \cap W|).$$

So our assumption that $D_1 \cap \cdots \cap D_l$ is T-odd was incorrect. Thus inductively, arbitrary intersections of the D_u's are T-even. It follows that $\cup_{u \in V} D_u$ is T-even, again by the inclusion-exclusion principle. But $u \in D_u$, so $\cup_{u \in V} D_u = V$, which is T-odd. This contradiction implies the result.

$\square$

3.3.3 A simpler proof for the final step

The following alternative proof of Theorem 3.16 is due to András Sebő, which he found after hearing our result. We include it here for completeness, since it is substantially shorter. Sebő in fact proved the following:

Theorem 3.20 ([122])**.** *Let $G = (V, E)$ be a connected multigraph, and $T \subseteq V$ a vertex set of even size. Let H be the disjoint union of k T-joins in G. Then for any node $s \in T$, there is another node $v \in V$ such that there are k edge-disjoint s-v-paths in H.*

Let us first see how this implies Theorem 3.16. First, construct F' as described in the earlier proof of Theorem 3.16, by adding a new node s connected to all the terminals; Theorem 3.16 asserts that there is a node $v \in V$ such that there are $|W|$ edge-disjoint s-v-paths in F'. Let $M'_u := M_u \cup \{(u, s)\}$ for all $u \in W$. By construction, F' is the disjoint union of all the M_u's. But M'_u is a T_s-join, since M_u is a T_u-cut and

$$T_u \triangle \{u, s\} = T' \triangle \{u, t\} \triangle \{u, s\} = T_s.$$

Thus F' is disjoint union of k different T_s-joins. Now applying Theorem 3.20 to the multigraph F' and the set T_s gives the conclusion of Theorem 3.16.

Proof of Theorem 3.20. By the max-flow min-cut theorem it suffices to prove that there is a node v such that the minimum s-v cut in H has size at least k. A basic property of T-cuts and T-joins is that every T-cut intersects every T-join. It follows that since H is the disjoint union of k T-joins, $|\delta_H(S)| \geq k$ for any T-cut S.

Consider a tree R with node set $V(H)$ (edges of R need not be edges of H). Any edge $a \in E(R)$ induces a natural cut $\delta(U_a)$ in the graph, by taking U_a to be one of the components of $R - a$. R is a *Gomory-Hu tree* of H if for any pair of distinct nodes $u, v \in V$, there is an edge $a \in E(R)$ on the unique u-v path in R so that $\delta(U_a)$ is a minimum u-v-cut in H. In particular, if $uv \in E(R)$ then $\delta(U_{uv})$ is a minimum u-v-cut. A Gomory-Hu tree exists for any graph [68].

Removing s from the Gomory-Hu tree R leaves several connected components. At least one of these connected components must have odd intersection with T; call this component C. Let sv be the edge in H connecting s to C. Then the cut defined by the edge sv in R is a T-cut and hence has size at least k. But since R is a Gomory-Hu tree, it is also a minimum s-v cut. Thus we have shown that the minimum cut between s and v has size at least k, and so there are k edge-disjoint s-v-paths, as required. $\square$

3.4 On another definition of the symmetric VPN problem

There is another, slightly different, definition of the symmetric model which is ocassionally used. The authors of [58] and [101] call an instance of the asymmetric VPN problem *symmetric* if $b_i^+ = b_i^-$ for all terminals i. In other words, there is still a differentiation between demand from i to j and demand from j to i, it is just the hose capacities that are symmetric. Thus in a single-path routing, a different route might be used from i to j than from j to i—we do not require that $P_{ij} = P_{ji}$. Another (less important) difference is a matter of scaling; assume that $b_i^+ = b_i^- = 1$ for all $i \in W$. Then the total incoming demand at a terminal i can be as large as 1, and the same for the outgoing demand, so the "total" demand terminating at the terminal is 2. Thus if we take a solution template with $P_{ij} = P_{ji}$ for all pairs $i, j \in W$, a solution to this half-symmetric problem will require exactly twice the capacity as for the symmetric VPN problem with $b_i = 1$ for all $i \in W$.

In this section, we demonstrate that this slight extra flexibility does not help, by using a slight variation of Lemma 3.5 of Grandoni et al. [71].

Theorem 3.21. *For an instance of the asymmetric VPN problem with symmetric hose capacities, the cheapest solutions still take the form of a truncated hub template $\mathcal{Q} = \{Q_{ij} : i, j \in W\}$, which is in particular symmetric: $Q_{ij} = Q_{ji}$ for all $i, j \in W$.*

It is sufficient to prove this for unit marginals; $b_i^+ = b_i^- = 1$ for all $i \in W$. The naturally corresponding symmetric VPN instance then has $b_i = 1$ for all $i \in W$. Note that a solution to such an instance using a symmetric routing template will cost twice as much when used as the solution template for the corresponding symmetric instance, since flow from i to j and from j to i will both require capacity.

The proof now follows immediately from Lemma 3.23 below, combined with Lemma 3.9. Of course, combined with the resolution of the PR conjecture, this implies the optimal solution still has the form of a tree.

First we need the following asymmetric analogues of the load and pyramidal cost:

Definition 3.22. Let

$$\begin{aligned}
\ell^+(e,\mathcal{P}_t) &:= |\{j \in W : e \in P_{tj}\}|, \\
\ell^-(e,\mathcal{P}_t) &:= |\{i \in W : e \in P_{it}\}|, \\
y^+(e,\mathcal{P}_t) &:= \min\{\ell^+(e,\mathcal{P}_t), k - \ell^+(e,\mathcal{P}_t)\}, \\
y^-(e,\mathcal{P}_t) &:= \min\{\ell^-(e,\mathcal{P}_t), k - \ell^-(e,\mathcal{P}_t)\}.
\end{aligned}$$

Lemma 3.23. *Given an instance of the asymmetric VPN problem $(G, W, \boldsymbol{b}^+, \boldsymbol{b}^-)$ with $b_i^+ = b_i^- = 1$ for all $i \in W$, and a routing template $\mathcal{P} = \{P_{ij} : i, j \in W\}$, there exists a terminal $t \in W$ so that $C_{\text{VPN}}(\mathcal{P}) \geq 2\min\{C_{\text{PR}}(\mathcal{P}_t^+), C_{\text{PR}}(\mathcal{P}_t^-)\}$, where $\mathcal{P}_t^+ = \{P_{tj} : j \in W\}$ and $\mathcal{P}_t^- = \{P_{it} : i \in W\}$.*

Proof. The proof is completely analagous to the proof of Lemma 3.5, with some slight changes to handle the extra asymmetry. For any edge $e \in E$, we define

$$D_{ij}^e := \begin{cases} \frac{1}{k}\left(\frac{y^+(e,\mathcal{P}_i)}{\ell^+(e,\mathcal{P}_i)} + \frac{y^-(e,\mathcal{P}_j)}{\ell^-(e,\mathcal{P}_j)}\right) & \text{if } e \in P_{ij}, \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

Note that this matrix is no longer necessarily symmetric.

Claim 3.24. *D^e is a valid hose demand matrix for all edges.*

Proof. We need to show that $\sum_{j \in W} D_{ij}^e \leq 1$ for all $i \in W$, and also that $\sum_{i \in W} D_{ij}^e \leq 1$ for all $j \in W$. The calculation is essentially the same as before:

$$\begin{aligned}
\sum_{j \in W} D^e_{ij} &= \sum_{j \in W : e \in P_{ij}} \frac{1}{k} \left(\frac{y^+(e, \mathcal{P}_i)}{\ell^+(e, \mathcal{P}_i)} + \frac{y^-(e, \mathcal{P}_j)}{\ell^-(e, \mathcal{P}_j)} \right) \\
&\leq \sum_{j \in W : e \in P_{ij}} \frac{1}{k} \left(\frac{k - \ell^+(e, \mathcal{P}_i)}{\ell^+(e, \mathcal{P}_i)} + \frac{\ell^-(e, \mathcal{P}_j)}{\ell^-(e, \mathcal{P}_j)} \right) \\
&= \sum_{j \in W : e \in P_{ij}} \frac{1}{k} \cdot \frac{k}{\ell^+(e, \mathcal{P}_i)} \\
&= 1.
\end{aligned}$$

The calculation for $\sum_{i \in W} D^e_{ij}$ is similar. □

Claim 3.25. *For every edge e we have*

$$u(e) \geq \frac{1}{k} \left(\sum_{i \in W} y^+(e, \mathcal{P}_i) + \sum_{i \in W} y^-(e, \mathcal{P}_i) \right). \tag{3.7}$$

Proof.

$$\begin{aligned}
u(e) &\geq \sum_{i,j \in W : e \in P_{ij}} \frac{1}{k} \left(\frac{y^+(e, \mathcal{P}_i)}{\ell^+(e, \mathcal{P}_i)} + \frac{y^-(e, \mathcal{P}_j)}{\ell^-(e, \mathcal{P}_j)} \right) \\
&= \frac{1}{k} \sum_{i \in W} \frac{y^+(e, \mathcal{P}_i)}{\ell^+(e, \mathcal{P}_i)} \sum_{j \in W} \mathbb{1}_{e \in P_{ij}} + \frac{1}{k} \sum_{j \in W} \frac{y^-(e, \mathcal{P}_j)}{\ell^-(e, \mathcal{P}_j)} \sum_{i \in W} \mathbb{1}_{e \in P_{ij}} \\
&= \frac{1}{k} \sum_{i \in W} \frac{y^+(e, \mathcal{P}_i)}{\ell^+(e, \mathcal{P}_i)} \cdot \ell^+(e, \mathcal{P}_i) + \frac{1}{k} \sum_{j \in W} \frac{y^-(e, \mathcal{P}_j)}{\ell^-(e, \mathcal{P}_j)} \cdot \ell^-(e, \mathcal{P}_j) \\
&= \frac{1}{k} \left(\sum_{i \in W} y^+(e, \mathcal{P}_i) + \sum_{i \in W} y^-(e, \mathcal{P}_i) \right).
\end{aligned}$$

□

The lemma now follows as before, by multiplying (3.7) by $c(e)$ and summing over all $e \in E$, and using that the minimum is no more than the sum. □

We also observe that with multipath routing, we also do not gain anything by allowing asymmetric routing templates. The argument is much more direct:

Lemma 3.26. *Any asymmetric VPN instance with symmetric hose constraints has an optimal solution where the routing template is symmetric.*

Proof. Let $\mathcal{P} = \{f_{ij} : i, j \in W\}$ be an arbitrary solution template. Now define $\mathcal{Q} = \{g_{ij} : i, j \in W\}$, where g_{ij} is the reverse of the flow f_{ji}, for all $i, j \in W$. Clearly $C(\mathcal{P}) = C(\mathcal{P}$; also, the template $\mathcal{R} = \{\frac{1}{2}(f_{ij} + g_{ij}) : i, j \in W\}$ is symmetric. But $C(\mathcal{R}) \leq \frac{1}{2}(C(\mathcal{P}) + C(\mathcal{Q}))$ by convexity: for any edge e,

$$\begin{aligned} u_{\mathcal{R}}(e) &= \max_{D \in \mathcal{H}} \sum_{i,j \in W} \tfrac{1}{2}(f_{ij}(e) + g_{ij}(e))D_{ij} \\ &\leq \frac{1}{2}\left(\max_{D \in \mathcal{H}} \sum_{i,j \in W} f_{ij}(e)D_{ij} + \max_{D \in \mathcal{H}} \sum_{i,j \in W} g_{ij}(e)D_{ij}\right) \\ &= u_{\mathcal{P}}(e) + u_{\mathcal{Q}}(e). \end{aligned}$$

□

Chapter 4

The multipath VPN Conjecture

It has been conjectured [86] (cf. [54]) that the multipath version of the VPN problem, where routing templates may be fractional, also has an optimal solution in the form of a tree. In this chapter we show that this conjecture is false, by demonstrating some small examples where the multipath optimum is cheaper than the cheapest tree solution (and hence, by the result of the previous chapter, the cheapest single-path solution).

We also commence an investigation into the worst-case gap between multipath and single-path routing. One of the examples in §4.2 yields a gap of $\mathrm{OPT}_{\mathrm{SPR}}/\mathrm{OPT}_{\mathrm{MPR}} = 9/8$. Theorem 3.2 actually shows that the optimal tree routing is in fact within a factor 2 of the optimal *dynamic* routing, not just the optimal single-path routing (this is implicit in [58] and explicit in [77]). Comparisons with dynamic routing will be discussed in Chapter 7; for now, we are more interested in the weaker implication, that the gap between multipath routing and single-path routing is at most 2. We do not improve this bound here, but we give some results in §4.4 that may be useful towards this goal.

We in fact give two quite different forms of counterexample. The examples in §4.2 are both simpler and have a larger gap than the construction of §4.3. The latter construction is included for completeness (this construction was used in the published version [69]). Perhaps the ideas in these two forms of counterexample can be combined to obtain a construction with a larger gap.

4.1 The compact LP formulation

Before describing the counterexamples, it will be useful to introduce the compact LP formulation, as defined by [86, 2]. This can also be obtained by applying Theorem 2.6 to the robust linear program that defines multipath routing in the hose model.

The formulation is:

$$\begin{array}{llll} & \min & \sum_{e\in E}\sum_{i\in W} c(e)y_i(e) & \\ \text{s.t.} & y_i(e)+y_j(e)\geq f_{ij}(e) & & \forall e\in E, i<j\in W, \\ & f_{ij} & \text{is a unit } i\text{-}j\text{-flow} & \forall i<j\in W \\ & y_i(e)\geq 0 & & \forall i\in W, e\in E \end{array}$$

The flows f_{ij} are defined on the bidirection of G, and we use $f_{ij}(e)$ to refer to the amount of flow on edge e, irrespective of direction. We also define f_{ij} for $i>j$ in a symmetric fashion: f_{ij} is the reverse of the flow f_{ji}.

This formulation yields a convenient description of the capacity reservation of a solution as a cost sharing between the terminals. We think of y_i as the capacity paid for by terminal i. We see that $\boldsymbol{y} := \{y_i : i \in W\}$ is a valid capacity reservation if and only if for every pair $i \neq j \in W$, $y_i + y_j$ supports a unit i-j-flow. Thus we may think of $\boldsymbol{y}$ as specifying a MPR solution, of cost

$$C(\boldsymbol{y}) = \sum_{i\in W} C(y_i) = \sum_{e\in E}\sum_{i\in W} c(e)y_i(e).$$

4.2 Two simple counterexamples

We begin with the smallest counterexample we know of (which is likely the smallest possible). It consists of only 7 nodes, and is shown in Figure 4.1; edge lengths are indicated. Half a unit of capacity is bought on each edge; the colours (and numbers) show which terminal pays for each edge, with the edges paid for by terminal 1 shown in bold. Since $y_i + y_j$ has enough capacity

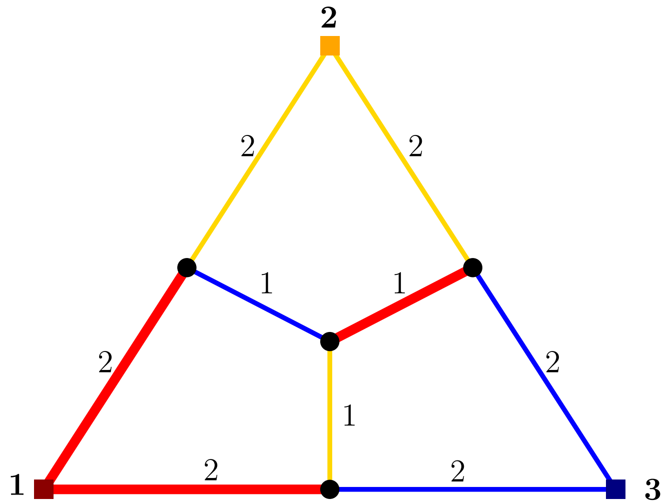

Figure 4.1: An instance where $\textsc{opt}_{\textsc{mpr}}$ is cheaper than $\textsc{opt}_{\textsc{spr}}$. Edges paid for by terminal 1 (red) are shown in bold.

for a unit i-j flow for every $i, j \in W$, this is indeed a feasible MPR solution of cost 15/2. The cheapest shortest path tree on the other hand (found just by checking all possible choices of root) has cost 8. By Theorem 3.3, this is the cost of the optimal SPR solution, and so we have a counterexample. The ratio between $\textsc{opt}_{\textsc{spr}}$ and $\textsc{opt}_{\textsc{mpr}}$ in this example is 16/15.

This example also shows that the multipath VPN Conjecture is false even for planar graphs.

We can obtain a slightly larger gap with a slightly larger construction, using the same idea. In Figure 4.2, all edges have cost 1. For the MPR solution, 1/3 capacity is bought on each edge; the edges paid for by terminal 1 (red) are shown in bold, and the rest are symmetric. This is again a feasible MPR solution, of cost 16/3. The cheapest tree solution on the other hand costs 6; this gives a ratio $\textsc{opt}_{\textsc{spr}}/\textsc{opt}_{\textsc{mpr}} = 9/8$.

The natural extensions of this construction to larger graphs do not yield any improvement to the gap.

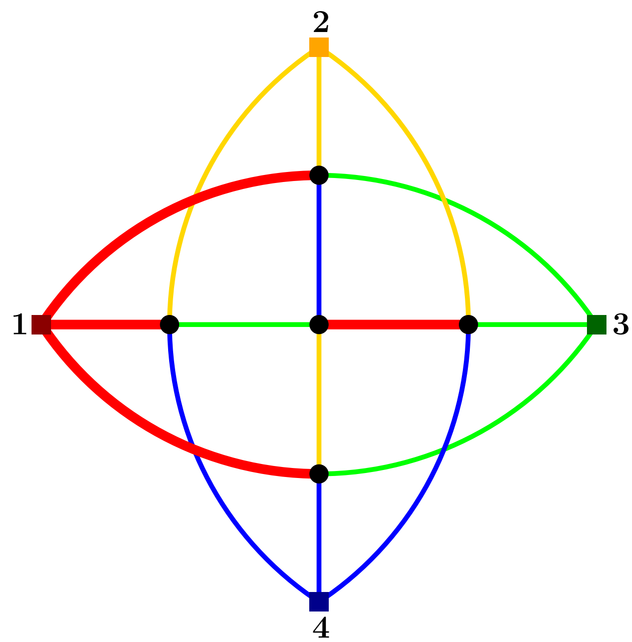

Figure 4.2: An instance with a larger gap. Edges paid for by terminal 1 (red) are shown in bold.

4.3 A construction based on combinatorial designs

Recall that a *projective plane* of order n is a $(n^2+n+1, n+1, 1)$ block design. It consists of a set W of n^2+n+1 *points*, and a collection $\mathcal{L}$ of subsets of points (the *lines*). Every line contains exactly $n+1$ points, every point is in exactly $n+1$ lines, every pair of lines determines a unique point, and every pair of points lies on a unique line. It is well known that projective planes exist for all orders that are powers of primes.

The complement $(W, \mathcal{B})$ of a projective plane $(W, \mathcal{L})$ is obtained by replacing each line with its complement: $\mathcal{B} = \{W \setminus L : L \in \mathcal{L}\}$. Call the sets in $\mathcal{B}$ *blocks*. This is an (n^2+n+1, n^2, n^2-n) design: every block contains n^2 points, every point is in n^2 blocks, every pair of blocks have exactly n^2-n points in common,

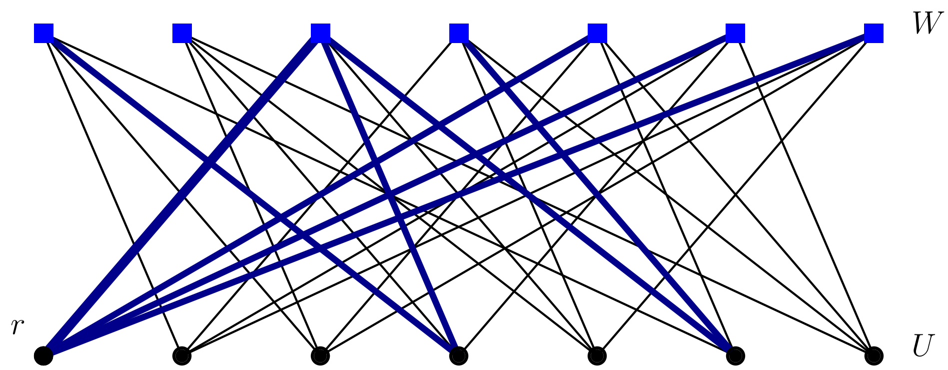

Figure 4.3: The construction for $n = 2$; the incidence graph of the complement of the Fano plane. (Note that this is not itself a counterexample; $n \geq 3$ is needed). The bold edges indicate a typical tree solution, rooted at r.

and every pair of points is contained in exactly $n^2 - n$ blocks.

We now construct the bipartite graph $G = (W \cup U, E)$ from the complement $(W, \mathcal{B})$ of a projective plane of order n as follows. The nodes in W correspond to points, and the nodes in U correspond to the blocks. An edge (w, B) exists if and only if w is contained in block B. Our construction implies that $|\delta(v)| = n^2$ for all $v \in W \cup U$, and $|\delta(v) \cap \delta(w)| = n^2 - n$ for $v, w \in W$ and $v, w \in U$. We also have $|W| = |U| = n^2 + n + 1$.

We now consider the MPR instance on this graph, where all edges have unit cost, and W is the set of terminals. Figure 4.3 shows the instance for the case $n = 2$. Notice that for $n \geq 2$, the optimal shortesst path tree solution has cost

$$\mathrm{OPT}_{\mathrm{SPR}} = n^2 + 3(n + 1),$$

by rooting at any node in U; a VPN tree routed on a node in W has cost $2(n^2 + n)$, which is larger. But consider the solution where terminal i pays for only edges in $\delta(\{i\})$, buying capacity $1/(n^2 - n)$ on each. Then since every $i, j \in W$ have $n^2 - n$ common neighbours, it follows that $y_i + y_j$ supports a unit i-j-flow, and so $\boldsymbol{y}$ is a feasible MPR solution, of cost

$$C(\boldsymbol{y}) = \frac{1}{n^2 - n} \cdot (n^2 + n + 1) \cdot n^2 = \frac{n(n^2 + n + 1)}{n - 1}.$$

A quick calculation shows that $C(\boldsymbol{y}) < \text{OPT}_{\text{SPR}}$ for $n \geq 3$, thus giving another class of counterexamples. The gaps are smaller than the examples of the previous section however.

The same technique yields a counterexample for any (v, l, λ) block design satisfying

$$\frac{v(v-1)}{l-1} < 3v - 2l.$$

Some examples are the complements of Steiner triple systems, and the unique $(9, 6, 5)$ design.

4.4 Towards a tighter bound on the gap

In this section, we consider the upper bound on the gap. We first note that in investigating this gap, it is sufficient to consider unit marginals. This follows by considering again the transformation given in §3.1.3. From the discussion there, any multipath routing for the original instance also gives a multipath routing in the transformed instance with the same cost; thus the optimal multipath routing can only get cheaper. By Theorem 3.3, it follows that the optimal single-path routing, being a tree, is unaffected by the transformation. Hence the gap in the transformed instance must be at least as large as in the original, and we consider only unit marginals from now on.

Can the upper bound of 2 be improved? In other words, does there exist an $\epsilon > 0$ so that $\text{OPT}_{\text{SPR}}/\text{OPT}_{\text{MPR}} \leq 2 - \epsilon$ for all VPN instances? We don't answer this question here, but we show that it in order to answer it, attention can be restricted to solutions of a particular form:

Definition 4.1. For any vector $\boldsymbol{b}$ indexed by nodes of G, and satisfying $\sum_v b_v = 1$, an MPR solution $\boldsymbol{y}$ is called a $\boldsymbol{b}$-*pleasant solution* if for all $j \in W$, y_j supports a flow with demands given by $\boldsymbol{b} - \boldsymbol{e}_j$. A solution which is $\boldsymbol{b}$-pleasant for some $\boldsymbol{b}$ is called just a *pleasant solution*.

There is an analogy with the T-join solutions of the previous chapter. There, we came across solutions defined by a subset T of the nodes, with terminal i paying for a T_i-join, where $T_i = T \triangle \{i\}$. Pleasant instances are much like a fractional analogue of this. The vector $\boldsymbol{b}$ takes the place of T, and the T_i-join becomes a $(\boldsymbol{b} - \boldsymbol{e}_i)$-flow.

We will prove the following:

Theorem 4.2. *For any $\gamma > 0$, If there exists an instance of the symmetric* VPN *problem where the ratio between single-path and multipath routing is at least $2 - \gamma$, then there exists a pleasant solution to this instance where this ratio is at least $(2-\gamma)/(1+2\gamma)$.*

This shows that if no upper bound better than 2 is possible, i.e., there exist instances where γ is arbitrarily small, then this can be demonstrated using pleasant solutions. In other words, for any $\epsilon > 0$ there would exist instances where the ratio between the optimal SPR solution and some pleasant solution is at least $2 - \epsilon$. These solutions seem much easier to deal with, so this could aid the search for such a construction. Conversely, to show an upper bound of some constant less than 2, it is sufficient to prove such a bound for all pleasant solutions.

Proof of Theorem 4.2. Let $\boldsymbol{y}$ be the optimal MPR solution. Let $Y_i := \operatorname{supp}(y_i)$; without loss of generality, we may assume that the Y_i's are disjoint, by making parallel copies of edges if necessary.

Let $\bar{f} := \frac{1}{k-1}\sum_{i \neq r} f_{ri}$. Define the MPR solution $\boldsymbol{y'}$ by $y'_r = y_r$, and for $j \neq r$,

$$y'_j(e) = \begin{cases} |f_{rj}(e) - \bar{f}(e)| & e \in Y_r \\ y_j(e) & e \notin Y_r \end{cases}.$$

We prove the following two lemmas:

Lemma 4.3. *The solution $\boldsymbol{y'}$ is a $\boldsymbol{b}$-pleasant solution, where*

$$b_v := \mathbb{1}_{r=v} + \sum_{a \in \delta^+(v) \cap Y_r} \bar{f}(a) - \sum_{a \in \delta^-(v) \cap Y_r} \bar{f}(a) \qquad \forall v \in V.$$

Lemma 4.4. *The additional cost of solution $\boldsymbol{y'}$ compared to $\boldsymbol{y}$ is at most*

$$C(\boldsymbol{y'}) - C(\boldsymbol{y}) \leq 2\sum_{j \neq r}(C(y_r) + C(y_j) - d(r,j)). \tag{4.1}$$

Once these lemmas are proven, the theorem is completed as follows. The above augmentation can be done for any choice of $r \in W$; in particular, we can choose r^* that minimizes (4.1). The additional cost of augmentation for this choice is then certainly not more than the average of (4.1) over all $r \in W$:

$$\begin{aligned} C(\boldsymbol{y'}) - C(\boldsymbol{y}) &\leq \frac{2}{k}\sum_{r \in W}\sum_{j \neq r}(C(y_r) + C(y_j) - d(r,j)) \\ &\leq \frac{2}{k}\sum_{j,r \in W}(C(y_r) + C(y_j) - d(r,j)) \\ &= 4C(\boldsymbol{y}) - \frac{2}{k}\sum_{r \in W} C(\mathcal{R}^r) \\ &\leq 4C(\boldsymbol{y}) - 2\textsc{opt}_{\textsc{spr}}, \end{aligned}$$

where recall that $\mathcal{R}^r$ is the shortest path solution rooted at r, which necessarily costs at least as much as the optimal SPR solution. But now since $\textsc{opt}_{\textsc{spr}}/C(\boldsymbol{y}) = 2 - \gamma$, we have $2C(\boldsymbol{y}) - \textsc{opt}_{\textsc{spr}} = \gamma C(\boldsymbol{y})$, and so

$$\begin{aligned} C(\boldsymbol{y'}) &\leq C(\boldsymbol{y} + 2\gamma C(\boldsymbol{y}) \\ &= (1 + 2\gamma)C(\boldsymbol{y}) \\ &= \frac{1 + 2\gamma}{2 - \gamma}\textsc{opt}_{\textsc{spr}}. \end{aligned}$$

Thus

$$\frac{\textsc{opt}_{\textsc{spr}}}{C(\boldsymbol{y'})} \geq \frac{2 - \gamma}{1 + 2\gamma},$$

as required.

Proof of Lemma 4.3. Let $\hat{f}_j$ be the portion of f_{rj} on Y_j only (i.e., excluding the portion on Y_r). This is a $\boldsymbol{b}^j$-flow, where

$$b_v^j = \mathbb{1}_{v=r} - \mathbb{1}_{v=j} + \sum_{a \in \delta^+(v) \cap Y_r} f_{rj}(a) - \sum_{a \in \delta^-(v) \cap Y_r} f_{rj}(a).$$

Now define h_j by $h_j(a) = f_{rj}(a) - \bar{f}(a)$ for all $a \in Y_r$, and $h_j(a) = 0$ for all $a \notin Y_r$. Then the total demand at any node v satisfied by $\hat{f}_j + h_j$ is

$$b_v^j - \sum_{a \in \delta^+(v) \cap Y_r} (f_{rj}(a) - \bar{f}(a)) + \sum_{a \in \delta^-(v) \cap Y_r} (f_{rj}(a) - \bar{f}(a)) \tag{4.2}$$

$$= \mathbb{1}_{r=v} - \mathbb{1}_{j=v} + \sum_{a \in \delta^+(v) \cap Y_r} \bar{f}(a) - \sum_{a \in \delta^-(v) \cap Y_r} \bar{f}(a) \tag{4.3}$$

$$= b_v - \mathbb{1}_{j=v}. \tag{4.4}$$

Hence $\hat{f}_j + h_j$ is a $(\boldsymbol{b} - \boldsymbol{e}_j)$-flow. Since y'_j supports $\hat{f}_j + h_j$ for each j, it is a $\boldsymbol{b}$-pleasant solution. □

Before calculating the cost of our augmentation, we note the following simple observation:

Claim 4.5. *If $x_1, x_2, \ldots, x_N$ are real numbers with $|x_i| \leq R$ for all i, then $\sum_i |x_i - \bar{x}| \leq 2N(R - \bar{x})$, where $\bar{x}$ is the average of the x_i's.*

Proof.

$$\begin{aligned}
\sum_i |x_i - \bar{x}| &= \sum_{i: x_i \geq \bar{x}} (x_i - \bar{x}) + \sum_{i: x_i < \bar{x}} (\bar{x} - x_i) \\
&\leq \sum_{i: x_i \geq \bar{x}} (R - \bar{x}) + \sum_{i: x_i < \bar{x}} (R - x_i) \\
&\leq \sum_i (R - \bar{x}) + \sum_i (R - x_i) \\
&= 2N(R - \bar{x}).
\end{aligned}$$

□

Proof of Lemma 4.4. The extra cost of $\boldsymbol{y'}$ is

$$C(\boldsymbol{y'}) - C(\boldsymbol{y}) = \sum_{e \in Y_r} c(e) \sum_{j \neq r} (f_{rj} - \bar{f})(e).$$

Consider a term corresponding to a fixed $e \in Y_r$ in the above. Let $x_j = f_{rj}(e)$ if f_{rj} uses edge e in the same direction as $\bar{f}$, and let $x_j = -f_{rj}(e)$ otherwise;

so $|x_j| \leq y_r(e)$. Then the average over all $j \neq r$ of x_j is $\bar{x} = \bar{f}(e)$, and $|x_j - \bar{x}| = (f_{rj} - \bar{f})(e)$. Thus by Claim 4.5,

$$\sum_{j \neq r} (f_{rj} - \bar{f})(e) \leq 2(k-1)(y_r(e) - \bar{f}(e)).$$

So

$$\begin{aligned} C(\boldsymbol{y'}) - C(\boldsymbol{y}) &\leq 2(k-1) \sum_{e \in Y_r} c(e)(y_r(e) - \bar{f}(e)) \\ &= 2 \sum_{e \in Y_r} c(e) \left(\sum_{j \neq r} y_j(e) + (k-1)(y_r(e) - \bar{f}(e)) \right), \end{aligned}$$

since $y_j(e) = 0$ for $j \neq r, e \in Y_r$. Now for any $e \notin Y_r$ and $j \neq r$, $y_j(e) \geq f_{rj}(e)$, implying $\sum_{j \neq r} y_j(e) \geq (k-1)\bar{f}(e)$. Thus including terms for $e \notin Y_r$ only increases the above sum, and we obtain

$$C(\boldsymbol{y'}) - C(\boldsymbol{y}) \leq 2 \sum_{e \in E} c(e) \left((k-1)(y_r(e) - \bar{f}(e)) + \sum_{j \neq r} y_j(e) \right).$$

Note that the flow $(k-1)\bar{f} = \sum_{j \neq r} f_{rj}$ routes 1 unit of flow from each terminal to r, and hence

$$(k-1) \sum_{e \in E} c(e) \bar{f}(e) \geq \sum_{j \neq r} d(r,j).$$

Finally,

$$C(\boldsymbol{y'}) - C(\boldsymbol{y}) \leq 2 \sum_{j \neq r} (C(y_r) + C(y_j) - d(r,j)). \qquad \square$$

This completes the proof of the theorem. $\square$

Chapter 5

The inapproximability of robust network design

5.1 Introduction

In the survey of Chekuri [41], one of the main open problems asked is the approximability status of the general robust network design problem with single-path routing, as defined in Chapter 1, where the demand universe is essentially arbitrary. Of course, it cannot be completely arbitrary, else a negative result is trivial; the universe could be too complicated to even have a polynomial-size description. It is sensible to consider the problem for "reasonable" universes—in particular, universes described as convex sets or polytopes that can be separated in polynomial time. Recall that a set $\mathcal{X}$ is separable if there is a polynomial time algorithm which for any x, determines whether or not x belongs to $\mathcal{X}$, and if not, returns a separating hyperplane. Slightly more restrictive, another possible definition for a "reasonable" universe is one that has an extended formulation, i.e., is the projection of a polytope with a compact LP formulation.

The general RND problem with single-path routing is of course APX-hard, since it includes Steiner tree (see §2.4). On the positive side, it was observed by Gupta in 2004 (see [41]) that there is an $O(\log n)$ approximation using metric embedding techniques. The question is to close this gap; is there in fact a constant factor approximation for any reasonable polytope, or is the problem much harder to approximate than the special cases considered thus far?

In this chapter, we give a negative result to this question. We show that the uniform buy-at-bulk network design problem can be simulated by a robust network design problem with a separable polytope (which also has an extended formulation). This immediately gives us a strong inapproximability result, since by the seminal work of Andrews [7], this problem is hard to approximate within a polylogarithmic factor (under suitable complexity assumptions).

For completeness, we begin with Gupta's proof that the problem is approximable within a logarithmic factor. We then define and discuss the uniform buy-at-bulk problem, before finally describing our reduction from it.

5.2 A logarithmic approximation algorithm via metric embeddings

Theorem 5.1 ([75]). *Given an instance of robust network design on a weighted graph $G = (V, E)$, with $|V| = n$, terminal set W, and separable universe $\mathcal{U}$, there is an $O(\log n)$-approximation algorithm for the* SPR *problem, and the cost ratio between the optimal* SPR *and* FR *solutions is $O(\log n)$.*

Proof. Let $c(e)$ refer to the edge weight of $e \in E$, and let $d(\cdot,\cdot)$ be the shortest path metric according to these weights. Now take K_G to be the metric completion of G, so that $e = vw \in E(K_G)$ has weight $d(v, w)$. There is a direct correspondence between solutions on K_G and solutions on G, by replacing an edge $vw \in E(K_G)$ with any shortest v-w-path in G.

From Theorem 2.11 of Fakcharoenphol et al. [57], we can find (in polynomial time) a distribution $\mathcal{D}$ over tree metrics such that for every ρ in the support of $\mathcal{D}$, $\rho(v, w) \geq d(v, w)$ for all $v, w \in V$ and $\mathbb{E}_{\rho\in\mathcal{D}}(\rho(v, w)) \leq \alpha d(v, w)$, where $\alpha = O(\log n)$. Moreover, $\mathcal{D}$ can be taken to contain only $O(n \log n)$ tree metrics.

The cost of a capacity reservation u is $C(u) := \sum_{v,w \in V} d(v,w)u(vw)$. We can also define the cost of this reservation with respect to another metric ρ by $C_\rho(u) := \sum_{v,w \in V} \rho(v,w)u(vw)$. Let u^* be the optimal capacity reservation using the *dynamic* routing model, so $C(u^*) = \textsc{opt}_{\textsc{fr}}$. We have by linearity of expectation that

$$\mathbb{E}_{\rho \in \mathcal{D}} C_\rho(u^*) \leq \alpha C(u^*).$$

So there exists a $\tau \in \mathcal{D}$ such that $C_\tau(u^*) \leq \alpha C(u^*)$; since $\mathcal{D}$ contains so few trees, we can in polynomial time find this τ. Let T be the tree associated with τ. On a tree, there is a unique simple path between every pair of terminals, and so all routing schemes coincide (and hence have the same optimum). Let $\mathcal{Q}_T := \{Q_{ij} : i, j \in W\}$ be the routing template induced by T, and let $u_T^* : E(T) \to \mathbb{R}_+$ be the corresponding optimal capacity vector. Since $d(v,w) \leq \tau(v,w)$ for all $v, w \in V$, it follows that $C(\mathcal{Q}_T) \leq C_\tau(\mathcal{Q}_T)$. We also have $C(\mathcal{Q}_T) = C_\tau(u_T^*) \leq C_\tau(u^*)$ by optimality of u_T^*. So finally,

$$C(\mathcal{Q}_T) \leq \alpha C(u^*) = O(\log n)\textsc{opt}_{\textsc{fr}},$$

as required. □

5.3 The uniform buy-at-bulk problem

Recall the definition of the (multicommodity) uniform buy-at-bulk problem given in §2.4. We are given an undirected graph G with nonnegative edge lengths $c : E \to \mathbb{R}_+$, as well as a single nonnegative, increasing and concave cost 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 load on edge e, i.e., the number of demand pairs using edge e in their routing path. Since f is concave, buying a large capacity on a single edge may be much cheaper than buying small capacities on many edges.

Andrews [7] showed that even on undirected graphs, this uniform buy-at-bulk problem is hard to approximate; in particular, it cannot be approximated within a ratio of $\Omega(\log^{1/4-\epsilon} n)$ for any $\epsilon > 0$, unless $\mathrm{NP} \subset \mathrm{ZPTIME}(n^{\mathrm{polylog}\, n})$. (This is in fact true even if f is restricted to be of the form $f(x) = L + x$, for some suitable L.)

5.4 A reduction from buy-at-bulk

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

Let Π be the set of permutation of the integers 1 through k, and let $\pi = (\pi_1, \pi_2, \ldots, \pi_k)$ be any such permutation. For notational convenience, we also define $\pi_0 = 0$. Define the demand matrix D^π by

$$D^\pi_{uv} = \begin{cases} f(\pi_i) - f(\pi_i - 1) & \text{if } \{u, v\} = \{s_i, t_i\} \text{ for some } 1 \leq i \leq k \\ 0 & \text{otherwise} \end{cases}. \tag{5.1}$$

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

$$\mathcal{B} := \mathrm{conv}\{D^\pi : \pi \in \Pi\}. \tag{5.2}$$

Theorem 5.2. *The buy-at-bulk problem on graph $G = (V, E)$ with lengths $c : E \to \mathbb{R}_+$ and cost function $f(\cdot)$, has the same optimum as the robust network design 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 $s_i t_i$ paths P_i for each $1 \leq i \leq k$. Let $\ell(e)$ be the number of demand pairs which use edge e on their path. Then for any edge e, the cost of this edge in the buy-at-bulk instance is

$c_e f(\ell_e)$. In the robust instance, the required capacity u_e is

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

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

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

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

It remains to show that this choice of $\mathcal{B}$ can be separated. In fact, we show:

Claim 5.3. *The polytope $\mathcal{B}$ defined in (5.2) has a compact extended formulation.*

Proof. Let $\mathcal{I} = \{\{s_i, t_i\} : 1 \le i \le k\}$. Any $D \in \mathcal{B}$ must satisfy $D_{uv} = 0$ for any pair $\{u, v\} \notin \mathcal{I}$. 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 $\boldsymbol{d}$, defined as $d_i = D_{s_i t_i}$ for all i. Also define $\boldsymbol{\delta}^\pi$ by $\delta_i^\pi = D_{s_i t_i}^\pi$; note that these are fixed vectors.

A matrix D is in $\mathcal{B}$ 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,

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

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

$D \in \mathcal{B}$ if and only if $\boldsymbol{d}$ is a convex combination of elements in $\{P^{\pi}\boldsymbol{\delta}^1 : \pi \in \Pi\}$; equivalently, there is some doubly stochastic matrix[1] M so that

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

This is clearly a linear system in the unknowns M and $\boldsymbol{d}$. The complete formulation is:

$$\begin{aligned}
D_{uv} &= 0 && \forall\, \{u,v\} \notin \mathcal{I} \\
D_{s_i t_i} &= d_i && \forall\, 1 \le i \le k \\
\boldsymbol{d} - M\boldsymbol{\delta}^1 &= \mathbf{0} && \\
\textstyle\sum_j M_{ij} &= 1 && \forall\, 1 \le i \le k \\
\textstyle\sum_i M_{ij} &= 1 && \forall\, 1 \le j \le k \\
0 \le M_{ij} &\le 1 && \forall\, 1 \le i, j \le k.
\end{aligned}$$

□

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

Chapter 6

Tree demands: generalizing the hose model

6.1 Motivation and definitions

As we have seen in the previous chapter, the general robust network design problem, with single-path routing, is hard to even approximate; specifically, under some complexity assumptions, it cannot be approximated to within polylogarithmic factors. As such, it is interesting to ask for relevant special cases or classes where the situation is better.

Positive results in robust network design have been given for relatively few demand matrix universes. The hose model, in both its symmetric and asymmetric forms, has received most of the attention. From the results in Chapter 3, the symmetric VPN problem is polynomially solvable. From Gupta et al. [81], the asymmetric VPN problem is approximable within a constant factor. Eisenbrand and Happ [50] considered the following generalization of the symmetric hose model. The terminals W are partitioned into groups $W_1, W_2, \ldots, W_m$. The demand universe is defined similarly to the hose model, except that there is no demand between terminals within the same group. In other words, the universe is

$$\mathcal{H}' := \mathcal{H} \cap \{D \in \mathbb{R}_+^{\binom{|W|}{2}} : \text{ for all } r, D_{ij} = 0\ \forall\, i, j \in W_r\},$$

where $\mathcal{H}$ is the usual symmetric hose universe. In fact, this is also a generalization of the *asymmetric* hose model. We may think of the asymmetric hose as a symmetric model, but where terminals are partitioned into *senders* and *receivers*; this is exactly the Eisenbrand-Happ model with 2 groups. They show a constant factor algorithm for this model.

In this chapter, we consider another natural generalization of the symmetric hose model, which we call the *tree demand model*.

Definition 6.1. Let T be any capacitated tree whose leaves are indexed by the terminals W. There are no other restrictions on the tree; it need not be a subgraph of the network, and the internal nodes of the tree do not correspond to nodes in the network. A symmetric demand matrix D_{ij} whose rows and columns are indexed by W is called a *T-demand* if it can be routed on T without violating the capacities on the edges of T. The set of T-demands defines a polytope that we denote by $\mathcal{U}_T$.

The *tree demand problem* (for a given T) is defined as the robust network design problem induced by G and the universe $\mathcal{U}_T$. Thus, we seek an oblivious routing for the terminals which minimizes the total capacity cost required to support all T-demands.

The usual symmetric hose model corresponds to the case where the tree is simply a star. For in this case, a symmetric demand D is feasible if for every $i \in W$, the leaf edge $ir \in E(T)$ (taking r to be the root of T) is not overloaded; in other words, $\sum_j D_{ij} \leq b_{ir}$. This is exactly the symmetric hose model with hose capacities $\boldsymbol{b'}$ given by $b'_i = b_{ir}$. Note however that the tree demand model and the asymmetric hose model are incomparable.

This appears to be a very natural definition, but there is some further motivation for this choice. 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

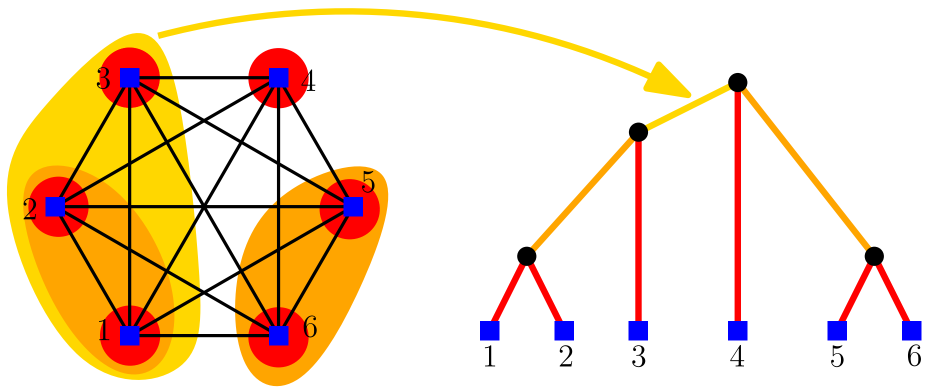

Figure 6.1: A laminar family of cut constraints, and associated tree demand.

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". In other words, we are given a family $\mathcal{S}$ of nontrivial subsets of W; for each $S \in \mathcal{S}$, an upper bound $b_S \in \mathbb{N}$ is prescribed, and any symmetric feasible demand D must satisfy

$$\sum_{i \in S, j \notin S} D_{ij} \leq b_S \qquad \forall S \in \mathcal{S}.$$

These extra cut constraints could be used to more accurately define the requirements of a VPN, possibly yielding a cheaper final network. Tree demands correspond exactly to the case where the sets in $\mathcal{S}$ form a nested family; the demand tree corresponds exactly to the nesting structure of the sets in $\mathcal{S}$ (see Figure 6.1).

It is interesting to compare this model with the Eisenbrand-Happ model. In their model, terminals within the same group do not communicate. In our model, if we think of a tree of two levels, this gives a natural grouping of terminals according to their parent node. But now the communication requirements between different groups will in general be smaller than between terminals in the same group.

Results We will 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 2 for the unit capacity case. In fact, the proofs imply something stronger: given the optimal network that supports each tree demand via dynamic routing, we can find an oblivious routing which costs no more than 8 times as much, or twice as much for unit capacities.

We do not, however, show that these bounds are tight; it is possible that the algorithm described is in fact optimal. This is the case if the tree is a star, by the result of Chapter 3.

6.2 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* [94, 37] on a tree.

Given our tree T with edge capacities $\boldsymbol{b}$, consider any mapping $h : V(T) \to V(G)$ such that $h(v) = v$ for each leaf $v \in W$. We think of an edge $uv \in E(T)$ as being mapped to some shortest path between $h(u)$ and $h(v)$ in G. Call this a hierarchical hubbing solution. The hierarchical hubbing problem asks for the hierarchical hubbing solution that minimizes

$$\sum_{uv \in E(T)} b_{uv} d_G(h(u), h(v)).$$

Recall that in the zero-extension problem we are given a set of terminals W within a weighted graph H and a metric ρ on W. We consider mappings h from $V(H)$ to W; we wish to find the mapping that minimizes the weighted sum $\sum_{uv \in E(H)} \rho(h(u), h(v))$. Thus in the case $W = V(G)$, 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 the cheapest shortest path tree solution for the instance.

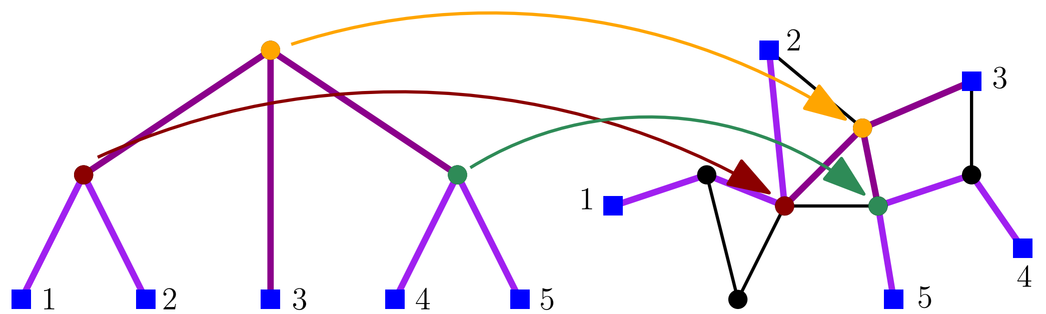

Figure 6.2: An example of a hierarchical hubbing.

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 concatenating the shortest paths between $h(x_i)$ and $h(x_{i+1})$ for each $1 \le i \le t-1$. This motivates the name "hierarchical hubbing".

Lemma 6.2. *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 nodes of T, i.e., by the positions of the hierarchical hubs. For any subtree S of T, and any node $v \in V$, let $C(S, v)$ be the cost of an optimal hierarchical hubbing solution for S, but with the root of S mapped to node v. For $S = \{i\}$ a leaf of T, define $C(\{i\}, i) = 0$ and $C(\{i\}, v) = \infty$ if $v \neq i$, i.e., mapping i to v is not valid.

We calculate these costs using dynamic programming. Let s be a node of T, and S the subtree rooted at s. Label the children of s as $s_1, s_2, \ldots, s_k$, and let S_i be the subtree rooted at s_i. Let e_i denote the edge from s to s_i. Suppose we know $C(S_i, w)$ for $1 \le i \le k$ 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 time algorithm. □

6.2.1 Another application of hierarchical hubbing

Shepherd and Winzer [124] describe an application of robust network design to optical networking. They first remark that the shortest path tree solution for the VPN problem, as given by Definition 3.1, has enough edge capacity to route all the hose matrices via the root node r without shortcutting. In other words, the solution template may be defined by taking P_{ij} as the union of a shortest i-r path combined with a shortest r-j path, possibly yielding a nonsimple path, but always passing through r. The advantage of this is that it avoids the need for expensive routing equipment at every node as used in the standard hop-by-hop architecture: instead, only the node r needs to do IP lookups. One problem with this approach is the single point of failure at r. To handle this, Shepherd and Winzer propose load balancing across multiple hubs, and hence multiple trees (called *selective randomized load balancing*). In the extreme case, where one balances across all possible hubs, we essentially have the classical Valiant randomized load balancing scheme (RLB) discussed in §2.3.4. Proposals to use randomized load balancing to minimize network performance measures such as congestion had been proposed around the same time [99]. In [124], the cost of hop-by-hop routing and hub routing via multiple hubs was empirically compared; costs included both optical and data (router) costs. They found that while the hop-by-hop IP routing architecture was cheaper than using RLB, it was considerably more expensive then using selective RLB across a limited number of hubs.

In [124] 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 described above. We define two extensions of the problem, *hub-constrained* and *leaf-constrained* hierarchical hubbing. In the hub-constrained version, each edge $uv \in E(T)$ has an associated bound $R(u,v)$ 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 i and node v on the path from i to r, we require that $d(h(i), h(v)) \leq R(i,v)$. The algorithm described above can be easily modified to find the optimal solution subject to these extra constraints. By enforcing these constraints, we may arrive at a solution which is more expensive, but with much better latency properties.

6.3 Analysis

The high level view of the analysis is as follows. We define a class of demand matrices D^ℓ; the index ℓ will be a so-called *connected labelling* of T. Each ℓ, and hence D^ℓ, will in turn be associated with a particular oblivious routing template $\mathcal{P}^\ell$. The important property of every D^ℓ is that it "dominates" the universe $\mathcal{T}$, in the sense that any capacity reservation on G that is sufficient to route D^ℓ, is sufficient to route all demand matrices in $\mathcal{T}$ obliviously, using the template $\mathcal{P}^\ell$. However, each D^ℓ will *not* be a valid T-demand. Instead, we define a distribution over all the D^ℓ's such that $\bar{D} := \mathbb{E}(D^\ell)$ lies in the scaled up polytope $\alpha \cdot \mathcal{U}_T$, for some constant α. It follows that for some ℓ, the cost of routing D^ℓ is within a factor α of the optimal robust network cost. Finding such a D^ℓ may not be so easy, so instead, we show that the cost of routing *any* D^ℓ is at least the cost of the optimal hierarchical hub routing; we have seen that this can be found in polynomial time. The hierarchical hub routing is thus a feasible solution to the tree demand problem that gives an α approximation. We will demonstrate a distribution that yields $\alpha = 8$; for the case where the capacities on T are all unit, we obtain $\alpha = 2$.

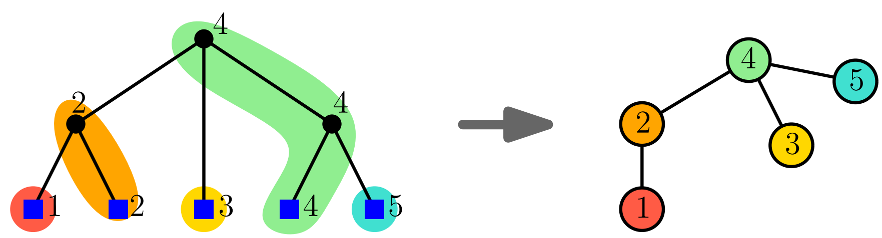

Figure 6.3: A connected labelling, and the associated T^ℓ obtained by contracting.

6.3.1 Connected labellings and hub routings

Definition 6.3. A *connected labelling* of a tree T is a function $\ell : V(T) \to W$, satisfying

(i) $\ell(w) = w$ for all $w \in W$,

(ii) $\ell^{-1}(w)$ 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 $\ell^{-1}(w)$ to obtain a new tree T^ℓ, with $V(T^\ell) = W$ (see Figure 6.3). The edges of T^ℓ determine the nonzero demands—if $uv \notin T^\ell$, then $D^\ell_{uv} = 0$. Now consider $uv \in T^\ell$; there is a unique edge $e \in T$ that connects the components $\ell^{-1}(u)$ and $\ell^{-1}(v)$. Define $D^\ell_{uv} = b_e$.

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

Lemma 6.4. *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 $u_\mathcal{Q} : E \to \mathbb{R}_+$ be the capacity reservation associated with this static routing. We define a routing template as follows. For any given pair u, v of terminals, consider the path between u and v in T^ℓ; let it be $v_0 v_1 \cdots v_m$, where $v_0 = u$ and $v_m = v$. Then for each edge $v_i v_{i+1}$ of this path, there is an associated route $Q_{v_i v_{i+1}}$ in $\mathcal{Q}$. We define P_{uv} to be a simple u-v path contained in the union $Q_{vv_1} \cup Q_{v_1 v_2} \cup \cdots \cup Q_{v_{m-1} v}$, and take $\mathcal{P}^\ell$ to be the routing template given by the P_{uv}'s.

Lemma 6.5. *The capacity reservation $u_\mathcal{Q}$ is enough to support the routing of any $D \in \mathcal{U}_T$ via $\mathcal{P}^\ell$.*

Proof. Let D be any T-demand, and let f be any edge of G. Let E' be the set of edges $e \in E(T^\ell)$ such that Q_e contains f. Note that since T^ℓ was obtained from T by contracting edges, we can think of an edge in 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 $\mathcal{P}^\ell$ 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 $D^\ell_{ij} = b_{ij}$ for each edge $ij \in E(T^\ell)$, the total load does not exceed $\sum_{ij \in E'} D^\ell_{ij} \le u_\mathcal{Q}(f)$ as required. □

6.3.2 Distributions over connected labellings

For any connected labelling ℓ, D^ℓ induces a load on edges in the original T. For edge $e = uv \in T$, this is $\sum_{uv \in S(e)} D^\ell_{uv}$, where recall $S(e)$ is the set of terminal pairs separated by e in T. If $e \in T^\ell$, the only pair in $S(e)$ with nonzero demand in D^ℓ is between $\ell^{-1}(u)$ and $\ell^{-1}(v)$, and this gives a load of b_e. For other edges, the load may generally exceed the edge's capacity b_e, and so D^ℓ may not be a valid T-demand. But suppose we manage to find a distribution so that the *expected load* across on any edge of T exceeds its capacity only by a constant factor α. Then consider the demand matrix $\bar{D}$ obtained by averaging the demand matrices D^ℓ over this distribution, i.e., the demand matrix given by $\bar{D}_{uv} = \mathbb{E}(D^\ell_{uv})$ for all $u, v \in W$. 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_{SPR}, i.e.,

$$C^*(\bar{D}) \leq \alpha \cdot \text{OPT}_{\text{SPR}}.$$

Since static routings are on shortest paths, we have a simple formula for $C^*(\bar{D})$:

Claim 6.6. $C^*(\bar{D}) = \mathbb{E}(C^*(D^\ell))$.

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

$$C^*(\bar{D}) = \sum_{u,v \in W} \bar{D}_{uv} d(u, v). \tag{6.1}$$

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

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

Taking expectations of both sides, and then using (6.1), we have

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

It follows from this claim that there must be some ℓ s.t. $C^*(D^\ell) \leq C^*(D)$. By Lemma 6.4, 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.

6.3.3 Expected loads for a distribution

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

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

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

So the total contribution is

$$\begin{aligned}
&\sum_{f \in E(T)} b_f \cdot \mathbb{1}_{(\text{one endpoint of } f \text{ has label in } L(e)\text{, the other in } R(e))} \\
&= \sum_{(x,y) \in \boldsymbol{A}_{\boldsymbol{e}}(T)} b_{xy} \cdot \mathbb{1}_{\ell(x) \in L(e) \wedge \ell(y) \in R(e)}.
\end{aligned}$$

The last line follows by the connectedness of the labellings, and our choice of orientation; it is not possible for $\ell(x)$ to be in $R(e)$ and $\ell(y)$ to be in $L(e)$.

Now consider any distribution over the labellings. We're interested in the

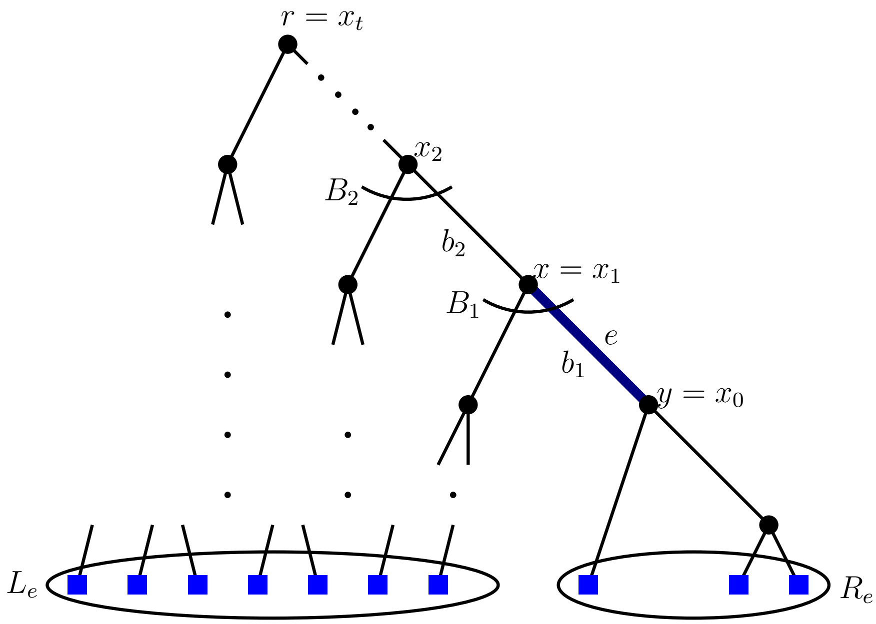

Figure 6.4: Calculating the expected load on edge e.

expected load on edges of T. By linearity of expectations, this is

$$\begin{aligned} &\sum_{(x,y)\in \boldsymbol{A}(e)} b_{xy}\mathbb{P}(\ell(x) \in L(e) \wedge \ell(y) \in R(e)) \\ &= \sum_{(x,y)\in \boldsymbol{A}(e)} b_{xy}\Big(\mathbb{P}(\ell(y) \in R(e)) - \mathbb{P}(\ell(x) \in R(e))\Big). \end{aligned} \tag{6.2}$$

This follows since there are only three possible events for the pair x, y (see Figure 6.4):

(i) $\ell(x), \ell(y) \in L(e)$,

(ii) $\ell(x) \in L(e)$, $\ell(y) \in R(e)$, or

(iii) $\ell(x), \ell(y) \in R(e)$.

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

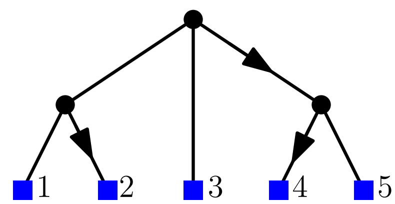

Figure 6.5: A choice of arrows leading to the connected labelling shown in Figure 6.3.

general capacities, this distribution does not yield a constant expected load; however, it is the starting point for constructing a distribution that does.

Define the random labelling ℓ using a coupled random walk scheme as follows. First, pick an arbitrary non-leaf node of T to be the root; call it r. For every non-leaf node s, pick one of its children at random, weighting the choices according to the edge capacities, and draw an arrow to it from s (see Figure 6.5). Now for any node s of T, define $\ell(s)$ to be the terminal reached by following the arrows from s. This clearly gives a (random) connected labelling.

Fix an edge $e \in E(T)$. We must compute the expected load on e, as given in Equation (6.2). Let us choose to orient e away from the root, so that $R(e)$ is the component of $T \setminus \{e\}$ below e, i.e., not containing the root. It is clear that any edges below e do not contribute to the sum, since walks from x and y definitely end up in $R(e)$ (the walks can't go up the tree). Likewise, any edge that is not on, or touching, the path from e to the root cannot contribute—x and y would both have to end up in $L(e)$.

Now label the nodes on the path from e to the root by $x_0 = y, x_1 = x, \ldots, x_t = r$. Let B_i be the sum of the capacities of the downward edges from x_i, and write $b_i := b_{x_i x_{i-1}}$ (see again Figure 6.4). There are two types of edges to consider:

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

$$\begin{aligned}
&b_i\Big(\mathbb{P}(\ell(x_{i-1}) \in R(e)) - \mathbb{P}(\ell(x_i) \in R(e))\Big)\\
&= b_i\Big(B_i/b_i \cdot \mathbb{P}(\ell(x_i) \in R(e)) - \mathbb{P}(\ell(x_i) \in R(e))\Big)\\
&= (B_i - b_i)\mathbb{P}(\ell(x_i) \in R(e)).
\end{aligned}$$

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

$$\begin{aligned}
&b_g\Big(\mathbb{P}(\ell(x_i) \in R(e)) - \mathbb{P}(\ell(z) \in R(e))\Big)\\
&= b_g\mathbb{P}(\ell(x_i) \in R(e)),
\end{aligned}$$

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

$$(B_i - b_i)\mathbb{P}(\ell(x_i) \in R(e)).$$

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

$$\begin{aligned}
&\sum_{i=1}^{t} 2(B_i - b_i)\mathbb{P}(\ell(x_i) \in R(e))\\
&= 2\sum_{i=1}^{t}(B_i - b_i)\prod_{j=1}^{i}\frac{b_j}{B_j}. \qquad (6.3)
\end{aligned}$$

6.3.4 Trees with unit capacities

If $b_e = 1$ for all $e \in E(T)$, then we have from Eq. (6.3) that the expected load on any edge is at most

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

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

6.3.5 Trees with arbitrary capacities

The same distribution does not work for arbitrary capacities. Consider a complete binary tree of height h, with all edges at height i having capacity 2^i. Then the expected load of an edge e adjacent to a leaf node is, by (6.3),

$$2\sum_{i=1}^{h-1}(2^{i+1}-2^i)\prod_{j=1}^{i}\frac{2^j}{2^{j+1}} = 2\sum_{i=1}^{h-1}2^i2^{-i} = 2(h-1).$$

So the expected load of this edge is $\Theta(\log n)$.

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

$$b_e \geq \sum_{e' \in \delta_T(y)\setminus\{e\}} b_{e'}, \tag{6.4}$$

then $\mathcal{U}_T$ 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 (6.4) does not occur for any such edge, we have that this contraction will not create any new wide edges. Repeat this process until we have a new tree $\hat{T}$, with associated demand polytope $\mathcal{U}_{\hat{T}}$. Since we only contracted wide edges of T, one easily checks that for any $D \in \mathcal{U}_{\hat{T}}$, $D/2 \in \mathcal{U}_T$. 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

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

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

Almost certainly, the constant 8 can be improved by a better choice of distribution. (We do not believe this argument can be extended to obtain a factor 2 in the general case however, and unlike for the hose model, there is no reduction to the unit capacity case).

Chapter 7

Comparing routing schemes: oblivious vs. dynamic routing

7.1 Introduction

Different possible routing strategies have already been mentioned in Chapter 1 and §2.3. This chapter is concerned with comparing the efficiency of different routing strategies.

Recall that the main dichotomy is between *dynamic* routing schemes, and *oblivious* routing schemes. In a dynamic scheme, the routing may adapt to the particular demand pattern currently being experienced by the network. The major disadvantage of this approach, as discussed in §2.2.2, is the difficulty of implementing it in practice. With oblivious routing on the other hand, the routing for a particular pair is fixed and can be set up in advance.

We restate the four routing schemes that will concern us in this chapter.

- **Dynamic routing** (FR): no solution template is specified; rather, the capacity reservation u must be sufficient so that for any demand matrix D in the universe, the fractional multicommodity flow problem for routing D with capacities u is feasible.

- **Multipath routing** (MPR): the solution is specified by a unit i-j flow

f_{ij}, for each pair $i, j \in W$; any demand between i and j is then routed according to these proportions. The routing template $\mathcal{P}$ is defined by $\mathcal{P} := \{f_{ij} : i, j \in W\}$.

- **Single-path routing** (SPR): the flows in the routing template are restricted to be integral, i.e., each pair i,j routes along a single path P_{ij}.
- **Tree routing** (TR): the flow template is again integral, but is also restricted to having support in the form of a tree.

We define again the general robust network design problem discussed already in Chapter 1, but with more emphasis on the role of the routing scheme.

Definition 7.1. Given a weighted graph $G = (V, E)$ on n nodes with edge costs $c : E \to \mathbb{R}_+$, a separable polytope $\mathcal{U}$ of demand matrices, and a routing model (FR, SPR, MPR or TR), the robust network design problem is:

> Find a minimum cost capacity installation $u : E \to \mathbb{R}_+$ so that all demand matrices in $\mathcal{U}$ can be routed according to the given routing model.

For a given instance of robust network design $(G, W, \mathcal{U})$, we use the notation $\text{OPT}_{\text{FR}}(G, W, \mathcal{U})$, $\text{OPT}_{\text{MPR}}(G, W, \mathcal{U})$, $\text{OPT}_{\text{SPR}}(G, W, \mathcal{U})$ and $\text{OPT}_{\text{TR}}(G, W, \mathcal{U})$ to denote the corresponding cost of an optimally designed robust network for the four routing models. If the context is clear, we may simply write, for instance, OPT_{FR}.

We clearly have

$$\text{OPT}_{\text{FR}} \le \text{OPT}_{\text{MPR}} \le \text{OPT}_{\text{SPR}} \le \text{OPT}_{\text{TR}}, \tag{7.1}$$

since the requirements on the routing scheme become stricter as we go from left to right. It was already known that the gap between OPT_{FR} and OPT_{SPR} is $O(\log n)$ [75], via tree embedding methods; this was discussed and proven in Chapter 5 as Theorem 5.1. A very similar argument, but using a theorem of Abraham et al. [1] which gives a probabilistic embedding of *spanning trees*, yields $\text{OPT}_{\text{TR}} = \tilde{O}(\log n)\text{OPT}_{\text{FR}}$, where $\tilde{O}$ hides an $O(\text{poly} \log \log n)$ factor.

Our Results In this chapter, our goal is to understand to what extent these gaps are realizable; in other words, for any pair of routing methods, what is the maximum possible gap between the costs of their optimal solution? In particular: how much do we lose by using oblivious routing rather than dynamic routing for robust design problems?

In short, the answer is that except for the pair $\{\mathrm{OPT}_{\mathrm{MPR}}, \mathrm{OPT}_{\mathrm{SPR}}\}$, the gap between any pair in (7.1) can be as large as $\Omega(\log n)$; this is essentially tight. The exception, the gap between $\mathrm{OPT}_{\mathrm{MPR}}$ and $\mathrm{OPT}_{\mathrm{SPR}}$, we show is at least polylogarithmically large.

We begin with the gap between single-path and tree routing in §7.2. We show that this gap can be logarithmically large by reducing to a well-known negative result on metric embeddings.

In §7.3, we consider multipath and single-path routing. This result will essentially be a corollary of the inapproximability result of Chapter 5, combined with an integrality gap obtained from the hard instances of uniform buy-at-bulk demonstrated by Andrews [7, 8].

The bulk of the content in this chapter will be demonstrating the gap between $\mathrm{OPT}_{\mathrm{FR}}$ and $\mathrm{OPT}_{\mathrm{MPR}}$; this is given in §7.4.

Discussion It is implicit in Fingerhut et al. [58] and explicit in Gupta et al. [77] that in the symmetric hose model, $\mathrm{OPT}_{\mathrm{MPR}} \leq \mathrm{OPT}_{\mathrm{SPR}} \leq 2 \cdot \mathrm{OPT}_{\mathrm{FR}}$. This is given with a proof as Theorem 3.2. This says nothing about the asymmetric hose model however; in fact, the gap instance between $\mathrm{OPT}_{\mathrm{MPR}}$ and $\mathrm{OPT}_{\mathrm{FR}}$ that we demonstrate is an instance of the asymmetric VPN problem, thus giving a logarithmic gap for this important case.[1]

This result shows that for at least some robust network design problems of practical interest, the routing model used may have a serious impact on the solution cost. While completely dynamic routing is typically infeasible for reasons mentioned previously, perhaps some tradeoff between the two extremes

[1]This rectifies an earlier assertion (cf. Theorem 4.6 in [41]).

of dynamic and oblivious routing could produce significantly better results while remaining practical: see Chapter 8 for a brief discussion.

It turns out that the problem of designing an SPR routing template for our gap instance corresponds to the rent-or-buy problem. In this problem (see §2.4) there is only one demand matrix instead of a polytope of demands, but the cost function is concave; it is truncated at some maximum value M. We sketch the lower bound argument for OPT_{SPR} separately in §7.4.3 since it is much simpler; it proceeds by showing that the optimal SPR templates may be assumed to be tree templates for our gap instance.

The lower bound for OPT_{MPR} is more involved. We show that the cost of an MPR template for our gap instance can be characterized by a network design problem that we call *buy-and-rent*. Again there is only one demand to be satisfied, but the cost function is more complex. The buy-and-rent cost function seems to be new and natural: briefly, instead of asking that each edge be either rented or bought, it allows that capacity may be partially bought and the rest rented. This new cost function is more amenable to analysis, and leads to our lower bound for OPT_{MPR}.

Relation to congestion lower bounds We remark that our lower bounds for the total cost model also imply lower bounds for minimizing the maximum congestion, since if every edge had congestion at most α times the dynamic optimum, the total cost would also be at most a factor α away. Since the polytope $\mathcal{H}^r$ we use is a subset of the single-sink demands routable in G, this also implies a result in [82] which gives an $\Omega(\log n)$ bound for congestion via oblivious routing of single sink demands (although their analysis also extends to the case of lower bounding performance of a general online algorithm). As discussed in §2.3.4, congestion minimization problems can be seen as equivalent to a robust optimization where one uses maximum edge congestion as a cost function; simply take the polytope consisting of all single-sink demands which are routable in G (this is a superset of our choice $\mathcal{H}^r$). The construction in [82] uses meshes (grids), building on work of [19, 107]. This construction does not seem to extend to the total cost model however, and we use instead a construction based on expanders, extending and simplifying a connection shown in earlier work [42].

7.2 Single-path routing vs. tree routing

As discussed in the introduction, for any robust network design problem we have $\text{OPT}_{\text{TR}} = \tilde{O}(\log n)\text{OPT}_{\text{FR}}$. We now show that this is essentially (up to $O(\text{poly} \log\log n)$ factors) best possible, by exhibiting a problem instance such that $\text{OPT}_{\text{TR}} = \Omega(\log n)\text{OPT}_{\text{SPR}}$, and so also $\text{OPT}_{\text{TR}} = \Omega(\log n)\text{OPT}_{\text{FR}}$.

We will consider a universe consisting of only a single demand matrix; the gap has nothing to do with robustness at all. Rather, it is simply a consequence of a negative result on metric embeddings.

Take any graph $G = (V, E)$ on n nodes, pick $c(e) = 1$ for all $e \in E$, and choose every node to be a terminal: $W = V$. Consider the demand matrix where every pair of adjacent terminals routes one unit of demand between them. Our universe consists of only this single demand, and so an optimal SPR template routes each communicating pair along a shortest path—in other words, the edge between them. Thus the optimal SPR template buys one unit of capacity on each edge, for a total cost of $|E|$.

If we restrict ourselves to a tree routing on the other hand, we must pick some spanning tree T of G, and route every pair via T. The total amount paid for routing a pair $u, v \in V$ is then $d_T(u, v)$, where d_T is the metric induced by T. The total cost is then $\sum_{uv \in E} d_T(u, v)$, giving a ratio between single-path and tree routing of

$$\frac{1}{|E|} \sum_{uv \in E} d_T(u, v),$$

or in other words, the average stretch over all edges of the metric induced by T. But it is well known that there exist graphs where this average stretch is $\Omega(\log n)$ [17]; in particular, since tree metrics are ℓ_1-metrics, expander graphs give such a lower bound [105].

7.3 Multipath vs. single-path routing

In Chapter 5, we showed that (under suitable complexity assumptions), the general robust network design problem is hard to approximate within a polylog factor. We should thus expect a similar gap between single-path routing and its fractional relaxation—i.e., multipath routing. Indeed, the hardness construction of Andrews [7] for buy-at-bulk does also yield an integrality gap of $\Omega(\log^{1/4-\epsilon} n)$ [8], which translates into the same gap between MPR and SPR using the polytope $\mathcal{B}$ defined in Chapter 5.

One might expect that there should be examples demonstrating a stronger $\Omega(\log n)$ gap, but we do not know of any such examples.

7.4 Oblivious vs. dynamic routing

7.4.1 A robust network design instance

Let $G = (V, E)$ be a graph on n nodes with constant degree $d \geq 3$ and edge expansion at least 1; in other words, we have that $|\delta_G(S)| \geq |S|$ for all $S \subseteq V$ with $|S| \leq n/2$. For d chosen large enough, expander graphs with these parameters exist for all n (see, e.g., [85]). Now add a special sink node r to V to obtain our instance $\bar{G} = (\bar{V}, \bar{E}) = (V \cup \{r\}, E \cup \{vr : v \in V\})$; see Figure 7.1.

Definition 7.2. The *single-sink hose model* with sink r is a special case of the asymmetric hose model, where the marginals $\boldsymbol{b}$ satisfy $b_r^+ = 0$, $b_r^- > 0$ (r is a receiver) and for all $v \in W \setminus \{r\}$, $b_v^- = 0$, $b_v^+ > 0$ (all other terminals are senders). We will always take sender to have unit marginals: $b_v^+ = 1$ for all $v \in W \setminus \{r\}$; the value b_r^- we call the *sink capacity*. The associated RND problem we call a *single-sink* VPN *problem*.

The demand universe we use is given by a single-sink hose polytope (this was also used in [42] for the hardness construction). All nodes are terminals, so $W = \bar{V}$, and r is the sink. For some $\beta < 1$ yet to be specified, we set $b_r^- = \beta n$

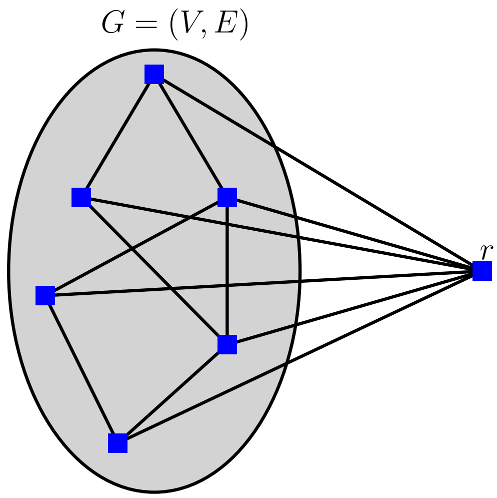

Figure 7.1: The gap instance. G is a d-regular expander

(and as stated, all the senders have unit marginals). We denote this universe by $\mathcal{H}^r$.

We assume throughout that $b_r^- = \beta n$ is an integer; when we write, e.g., $\beta = 1/\log n$, this may be read as choosing $\beta \approx 1/\log n$ with βn integral.

Notice that:

Observation 7.3. *If b_r^- is an integer, then our network is robust for $\mathcal{H}^r$ and a given routing model if and only if for each subset X of b_r^- nodes in G, there is enough capacity to route one unit from each node in X to r, using the prescribed routing model.*

We use this fact below. Finally, we also assign costs to the edges: each edge of G has cost 1, and each edge in $\delta_{\bar{G}}(r)$ has cost $1/\beta$.

Our main result is the following theorem:

Theorem 7.4. *For $\beta = 1/\log n$, there is a dynamic routing for the single-sink hose model instance (defined above) of cost $O(n)$, but every* MPR *solution (and hence every* SPR *solution) has cost $\Omega(n \log n)$.*

The first assertion is proved in the next section. In §7.4.3, we see that determining OPT_{SPR} for single-sink hose models is equivalent to the single-sink rent-or-buy problem, and that this problem always has a tree solution that is optimal. This can be used to show that $\text{OPT}_{\text{SPR}} = \Omega(n \log n)$ for our instance with $\beta = 1/\log n$. We give a sketch proof of this since it is considerably simpler than (but implied by) the proof of the corresponding bound for MPR. This MPR lower bound is demonstrated in §7.4.4.

We mention that if instead b_r^- is set to 1, there is no gap at all between OPT_{MPR} and OPT_{FR}, since Frangioni et al. [63] showed that MPR and FR coincide in the single-sink hose model when all marginals are 1.

7.4.2 A solution for the dynamic routing model

Put capacity β on each edge of $\delta_{\bar{G}}(r)$, and capacity 1 on each edge of G. Clearly, the cost of this reservation is $O(n)$ independent of β. We show that this is a valid FR capacity reservation. From Observation 7.3, it suffices to show that for any subset of βn nodes X in G, all nodes in X can simultaneously route a unit flow to r. To this end, we add a new node t to $\bar{G}$ and edges vt for $v \in X$ with unit capacity to form graph G'. We show that G' supports a t-r flow of size $|X| = \beta n$. By the max-flow min-cut theorem it suffices to show that all r-t cuts in G' have size at least βn, i.e., that for each $S \subseteq V$ we have $|\delta_{G'}(S \cup \{t\})| \geq \beta n$.

We have

$$|\delta_{G'}(S \cup \{t\})| = \beta|S| + |X \setminus S| + |\delta_G(S)|.$$

Now, if $|S| \leq n/2$ then using the fact that for G we have $|\delta_G(S)| \geq |S|$ we get

$$\begin{aligned}|\delta_{G'}(S \cup \{t\})| &\geq \beta|S| + |X \setminus S| + |S| \\ &\geq \beta|S| + |X| \\ &\geq |X|.\end{aligned}$$

If on the other hand $|S| > n/2$, then since $|\delta_G(S)| \geq n - |S|$ we get

$$\begin{aligned}
|\delta_{G'}(S \cup \{t\})| &\geq \beta|S| + |X \setminus S| + n - |S| \\
&\geq \beta|S| + |X \setminus S| + \beta(n - |S|) \\
&= \beta n + |X \setminus S| \\
&\geq \beta n.
\end{aligned}$$

Hence the above capacity reservation can support the FR routing model and costs $O(n)$.

7.4.3 Rent-or-buy: lower bounds for SPR

Recall the single-sink rent-or-buy problem defined in §2.4. Given our undirected graph $G = (V, E)$ with edge weights $c(e)$, a set of terminals $W \subset V$, and a distinguished root node $r \in W$, the goal is to route all terminals unsplittably to the sink. However, each edge may be either rented or bought; if rented, we pay $c(e)$ for each terminal using the edge in their path to r, and if bought, we pay $M \cdot c(e)$, and all terminals may use the edge.

Lemma 7.5. *Given an instance of the single-sink* VPN *problem with sink r, the cost of routing any* SPR *template $\mathcal{P} = \{P_{ir} : i \in W\}$ is the same as the cost of this same routing for a single-sink rent-or-buy instance with root r and cutoff $M = b_r^-$.*

Proof. Consider any edge e. With respect to the asymmetric VPN instance, the capacity requirement $u(e)$ is given by

$$u(e) = \max_{D \in \mathcal{H}^r} \sum_{i \in W : e \in P_{ir}} D_{ir}.$$

Let $\ell(e) := |\{i : e \in P_{ir}\}|$; if $\ell(e) \leq b_r^-$, then we see that $u(e) = \ell(e)$, by setting $D_{ir} = \mathbb{1}_{e \in P_{ir}}$ for all $i \in W$. If $\ell(e) > b_r^-$, then clearly $u(e) = b_r^-$. Thus the total cost is

$$\sum_{e \in E} c(e) \min\{\ell(e), b_r^-\},$$

exactly the SSROB cost of the same routing. □

The following is immediate from the concavity of the rent-or-buy cost function:

Proposition 7.6. *Any SSROB problem has an optimal solution whose support is a tree.*

Proof. Let f be a flow giving an optimal solution to the rent-or-buy instance, chosen so that $\mathrm{supp}(f)$ is setwise minimal. We show that then $\mathrm{supp}(f)$ must form a tree.

Let us consider f as a directed flow, where each terminal sends flow to the sink. If there is any directed cycle in the support of f, then we may simply reduce flow on this cycle until some arc becomes zero; this does not increase the cost since our cost function is nondecreasing. So we may assume our support is acyclic in the directed sense. Suppose now that there is some undirected cycle K in the support which by assumption corresponds to some forward (traversing K in order) arcs F and some reverse arcs R. Let $\epsilon = \min\{f(a) : a \in R \cup F\}$. Define two solutions f^+, f^- by $f^{\pm}(a) = f(a) \pm \epsilon$ for $a \in F$, and $f^{\pm}(a) = f(a) \mp \epsilon$ for $a \in R$. By concavity of the rent-or-buy cost function, $C(f) \geq (C(f^+) + C(f^-))/2$. Then since f was an optimal solution, $C(f^+) = C(f^-) = C(f)$. Hence both f^+ and f^- are optimal, and one of them must have smaller support than f, a contradiction. □

Note that the preceding result shows that in the case of single-sink hose models, $\mathrm{OPT}_{\mathrm{SPR}} = \mathrm{OPT}_{\mathrm{TR}}$. It is not the case that $\mathrm{OPT}_{\mathrm{MPR}} = \mathrm{OPT}_{\mathrm{TR}}$ in this setting however: if that were the case, SSROB would be polynomially solvable. Because of this tree structure, arguing why the gap holds in the case of SPR is considerably simpler than for MPR. The argument contains some intuition as to why the gap also holds for MPR, so we outline this approach now.

Suppose the optimal SPR solution uses only one edge rv from $\delta(r)$. Then in the SPR solution, everyone must route to v in G; by Proposition 7.6, this solution has the form of some tree T. Since G was bounded degree this means that a constant fraction of the terminals must use long paths to r, of length $\log_d(n)$. If these all had to pay one unit along their whole path then this already costs $\Omega(n \log n)$. But it is not as easy as that; if we have a subtree T_w rooted at node w that contains at least $b_r^- = \beta n$ nodes, then in fact we only need to pay for b_r^- units on the edge out of w.

Imagine removing these "heavy" edges of T which are used by more than βn terminals. This leaves a number of subtrees, each containing at most βn terminals. If T is fairly balanced, there are around $\Theta(n/(\beta n)) = \Theta(1/\beta)$ such subtrees. (If T is very unbalanced on the other hand, there could be many more—consider a caterpillar. For the full proof, one must use the increased cost of the heavy edges to obtain the required bound.)

In each such subtree, a good fraction of the leaves are a distance roughly $\log \beta n$ from the root of this subtree. Since there is no cost sharing within this subtree, these nodes really do pay $\beta n \log(\beta n)$. Thus the subtrees combined pay

$$\Omega\left(1/\beta \cdot \beta n \log(\beta n)\right) = \Omega\left(n \log(\beta n)\right).$$

If we set $\beta = \frac{1}{\log n}$, this yields a cost of $\Omega(n \log n)$.

To make the above argument precise, we would need to deal with possibly multiple edges into r, as well as unbalanced solution trees. Since this does not extend to establish the gap between MPR and FR, we instead turn to this latter problem; this will immediately imply the gap between SPR and FR.

7.4.4 Buy-and-rent: a logarithmic gap between FR and MPR

Analyzing the MPR model is more difficult, partially because the analogue of Proposition 7.6 does not hold; we cannot assume that the solution has a convenient tree structure.

Let us first examine more closely the cost on edges induced by an MPR routing template for a single-sink VPN problem. As in Observation 7.3, a capacity allocation is feasible if it can support the routing of any βn terminals routing to r simultaneously, using the given routing template. If terminal i routes according to an i-r-flow f_i, then the capacity required for an edge e is

$$\max_{D \in \mathcal{H}^r} \sum_{i \in V} D_{ir} f_i(e) = \max_{W \subseteq V : |W| = \beta n} \sum_{i \in W} f_i(e), \tag{7.2}$$

where recall that $\mathcal{H}^r$ is the set of single-sink hose matrices. In other words, the capacity needed on edge e is just the sum of the βn largest values of $f_i(e)$.

We introduce a new routing cost model which we call (single-sink) *buy-and-rent* (BAR). This exactly models the MPR cost model defined above, but is more manageable in terms of analysis. In the buy-and-rent problem, there are costs on the edges, and unit demands from some subset W of terminal nodes. Each terminal wishes to fractionally route one unit of demand to the sink r. Apart from the costs $c(\cdot)$ on the edges, we also have a parameter M. The difference from rent-or-buy is that we may now purchase some capacity amount $\gamma(e) \in [0, 1]$ (in rent-or-buy we would buy an infinite capacity link), and the interpretation is that every terminal is allowed to use up to $\gamma(e)$ units of capacity on the edge. If a terminal chooses to route any more on that edge, then it must pay for the additional rental cost. The cost of purchasing the $\gamma(e)$ capacity on an edge e is $M\gamma(e)c(e)$.

Buy-and-rent can be considered as an LP relaxation of single-sink rent-or-buy. This formulation is in fact very similar to the LP relaxation used by Swamy and Kumar [129] to give constant factor approximation algorithms for connected facility location and single-sink rent-or-buy. Their formulation is stronger however (in that the optimum for their LP lies between the BAR and SPR optima), and so does not exactly model the MPR problem. In particular, in buy-and-rent, solutions may conceivably use flow paths that alternate several times between rented capacity and purchased capacity. In contrast, a solution to the LP of Swamy and Kumar [129] always has a connected "core" of purchased edges containing the sink node and terminals use rented capacity to route to that core.

Proposition 7.7. *On a graph G with edge costs $c(\cdot)$, terminal set W and sink $r \in W$, any multipath solution to the single-sink* VPN *problem with sink capacity $b_r^- = M$ has the same cost when considered as a solution to the buy-and-rent problem with parameter M.*

Proof. Let $k = |W|$, and label the terminals 1 through k. We may assume $M \le k$. Let $\mathcal{P} = \{f_i : i \in W\}$ be any MPR template for the single-sink VPN problem. We construct a BAR solution with parameter M using the same routing template; so terminal i will use flow f_i. For the bought capacity, consider an arbitrary edge e, and let π be a permutation of $\{1, 2, \ldots, k\}$ so that

$$f_{\pi(1)}(e) \ge f_{\pi(2)}(e) \ge \cdots \ge f_{\pi(k)}(e). \tag{7.3}$$

We then purchase $\gamma(e) = f_{\pi(M)}(e)$ units of capacity on edge e. This guarantees that for any edge, none of the terminals $\pi(j)$ with $j > M$ pays to route on edge e, since we purchased enough capacity for them to travel for free. Each terminal $\pi(j)$ with $j \le M$ must pay the rental cost to route an amount $f_{\pi(j)}(e) - f_{\pi(M)}(e) \ge 0$. This costs

$$c(e) \sum_{j \le M} \Big(f_{\pi(j)}(e) - f_{\pi(M)}(e)\Big) = c(e)\left(\sum_{j \le k} f_{\pi(j)}(e) \ - \ M f_{\pi(M)}(e)\right).$$

Since the purchased capacity costs $Mc(e)f_{\pi(M)}(e)$, the total buy-and-rent cost is

$$c(e) \sum_{j \le M} f_{\pi(j)}(e),$$

which is the cost of edge e in the MPR template using (7.2).

Conversely, suppose that we have a minimum cost solution for BAR, and consider the robust design cost for using the same routing template. Again, consider a fixed edge e and π satisfying (7.3).

We claim that we may assume $\gamma(e) = f_{\pi(M)}(e)$. For suppose $\gamma(e)$ were larger than this, and consider the effect of reducing it by some sufficiently small $\epsilon > 0$. The purchase costs would decrease by $\epsilon M c(e)$. The rental costs for terminals $\pi(j)$ for $j \geq M$ would be unaffected, and the rental costs of terminals $\pi(j)$ with $j < M$ would increase by at most $\epsilon c(e)$. Thus the total increase in rental costs are at most $M\epsilon c(e)$, which is no more than the savings in purchase costs. Similarly, if $\gamma(e) < f_{\pi(M)}(e)$, then increasing the bought capacity $\gamma(e)$ by some small $\epsilon > 0$ costs $\epsilon Mc(e)$. The reduction in rental costs is at least the reduction in rental cost of the first M terminals, which is $\epsilon Mc(e)$, and thus the overall cost does not increase as a result of increasing $\gamma(e)$.

So assuming this, the total cost paid on edge e is just the purchase cost $Mc(e)f_{\pi(M)}(e)$ plus the rental cost $c(e)\sum_{j\leq M}(f_{\pi(j)}(e) - f_{\pi(M)}(e))$. This is identical to the robust design cost when using the same template $\mathcal{P}$. □

We again take $\beta = 1/\log n$, so $M = \beta n = n/\log n$. We now prove that any solution to the BAR problem on our expander instance is expensive; this together with the preceding proposition implies our main result, Theorem 7.4.

Theorem 7.8. *Any solution to the* BAR *problem on the expander instance has cost* $\Omega(n \log n)$.

Proof. Consider an arbitrary BAR solution, determined by bought capacity γ_e on each edge, and a flow template $\mathcal{P} = \{f_i : i \in W\}$.

Let $\gamma(\delta(r)) := \sum_{v\in V}\gamma_{vr}$ be the total bought capacity on the *port edges* (these are the edges connecting r to the nodes in V), and let $\gamma(E) := \sum_{e\in E}\gamma_e$ be the capacity bought in the expander. The cost of buying capacity in the expander is then $M \cdot \gamma(E)$, so we may assume that $\gamma(E) < \log^2 n$, or else the solution already costs $\Omega(n \log n)$. A similar argument for port edges (but recalling that these edges cost $\log n$) allows us to assume that $\gamma(\delta(r)) < \log n$.

For a terminal v, let $B_i(v)$ be the set of nodes (or sometimes, their induced graph) in the expander that are a distance at most i from v. We are particularly interested in balls of radius $R := \lfloor \log_d \sqrt{n} \rfloor - 1 = \lfloor \log n/(2 \log d) \rfloor - 1$; we use $B(v)$ as shorthand for $B_R(v)$. Note that since G is d-regular,

$$|B(v)| \leq \sum_{i=0}^{R} d^i \leq d^{R+1} \leq n^{1/2}.$$

Let $\gamma^E(v) := \sum_{e\in E: e\subset B(v)} \gamma(e)$ and $\gamma^P(v) := \sum_{w\in B(v)} \gamma(wr)$. A single $\gamma(e)$ for an edge $e = u_1u_2$ contributes to many $\gamma^E(v)$'s, but not too many:

$$|\{v : e \subset B(v)\}| \le |\{v : u_1 \in B(v)\}| = |B(u_1)| \le n^{1/2}.$$

So we must have that

$$\sum_{v\in V} \gamma^E(v) \le n^{1/2}\gamma(E) \le n^{1/2}\log^2 n. \tag{7.4}$$

Similarly,

$$\sum_{v\in V} \gamma^P(v) \le n^{1/2}\log n. \tag{7.5}$$

Consider an arbitrary terminal v. The unit of flow from v can be divided up into three types depending on how the flow enters r:

- A fraction μ_v^r of flow that rents on the port edge it uses.
- A fraction μ_v^b of flow that uses bought port capacity, on a port within a distance R from v.
- A fraction μ_v^t representing all remaining flow; this flow must "travel" and use port edges that are further than R from v.

Clearly $\mu_v^r + \mu_v^b + \mu_v^t = 1$.

We now aim to find a lower bound on the total rental cost paid by the terminals. Flow that rents the port edge must pay $\log n$ just for this edge, giving a cost of $\mu_v^r \log n$. Now consider the μ_v^t fraction of flow that travels outside the ball $B(v)$ in the expander before using a port edge. This flow must cross each of the cuts $K_i := \delta(B_i(v))$, for $0 \le i \le R$.

The maximum amount of flow that can travel across cut K_i for free (using the bought capacity) is $\gamma(K_i)$, and so there is a rental cost of at least $\mu_v^t - \gamma(K_i)$ in crossing cut K_i. Summing over all the cuts, we find that the rental cost associated with this travelling flow is at least

$$\sum_{i=0}^{R-1} (\mu_v^t - \gamma(K_i)) \ge R\mu_v^t - \gamma^E(v).$$

Thus the rental cost associated with terminal v is at least

$$\log n \cdot \mu_v^r + R\mu_v^t - \gamma^E(v).$$

Summing this over all terminals v, we obtain a total rental cost of at least

$$\begin{aligned} C_{\text{RENT}} &\geq \sum_{v \in V} (\log n \cdot \mu_v^r + R \cdot \mu_v^t) - \sum_{v \in V} \gamma^E(v) \\ &\geq R \sum_{v \in V} (\mu_v^r + \mu_v^t) - \sum_{v \in V} \gamma^E(v) && \text{since } R \leq \log n \\ &\geq R \sum_{v \in V} (\mu_v^r + \mu_v^t) - n^{1/2} \log^2 n && \text{by (7.4).} \end{aligned}$$

Finally, note that

$$\begin{aligned} \sum_{v \in V} (\mu_v^r + \mu_v^t) &= \sum_{v \in V} (1 - \mu_v^b) \geq \sum_{v \in V} (1 - \gamma^P(v)) \\ &\geq n - n^{1/2} \log n && \text{by (7.5).} \end{aligned}$$

Thus

$$\begin{aligned} C_{\text{RENT}} &\geq R \cdot (n - n^{1/2} \log n) - n^{1/2} \log^2 n \\ &= \Omega(n \log n), \end{aligned}$$

since $R = \Theta(\log n)$. □

Chapter 8

Conclusion

We have made substantial progress on a number of questions related to robust network design. In particular, we have:

- resolved the VPN Conjecture (positively), as well as its multipath generalization (negatively),
- demonstrated that the general robust network design problem with a separable universe is hard to approximate with polylogarithmic factors,
- investigated the "tree demand" model that generalizes the symmetric hose model, obtaining positive approximation results, and
- compared dynamic and oblivious routing, to demonstrate a worst-case logarithmic gap, even for the asymmetric hose model.

However, there are still many interesting unanswered questions. We survey some of the most interesting ones here.

The gap between MPR and SPR in the symmetric hose model In Chapter 4, we showed that the optimal MPR solution may be fractional. A gap of 9/8 was demonstrated between the MPR and SPR models. An upper bound of 2 on this gap follows from the proofs of Fingerhut et al. and Gupta et al. It would be interesting to close this gap. Another, possibly easier, question is the following. Is there always an integral MPR optimum if the network is a series-parallel graph?

The approximability of single-sink robust network design Our inapproximability result for the general robust network design problem is via a reduction from multicommodity buy-at-bulk. However, as we have seen the *single-sink* buy-at-bulk problem is $O(1)$-approximable. It is then natural to ask: does the single-sink robust network design problem, where all nonzero demand terminates at a specified root node r, admit a constant-factor approximation algorithm?

The complexity of the tree demand problem We showed that the hierarchical hubbing algorithm proposed for the tree demand problem in Chapter 6 has a constant factor approximation ratio. This can be thought of as a generalization of the factor 2 result for the symmetric VPN problem [58, 77]. For the special case where the demand tree is simply a star, the hierarchical hubbing algorithm exactly finds the cheapest shortest path tree on the terminals; this is precisely the optimal tree solution, and hence by the result of Chapter 3, is optimal.

This raises the obvious question: is the hierarchical hubbing algorithm *always* optimal, for any demand tree? This would be a very pleasing generalization of the VPN result.

Generalizations of the tree demand model There is a natural progression from the tree demand model discussed above: instead of taking the polytope defined by demands routable on a given tree T, we consider demands routable on a given graph G. We are then trying to essentially simulate G with an oblivious routing on the network. Another possible generalization comes from the cut interpretation of tree demands. A tree demand universe is specified by a laminar family of cuts on the demand graph (i.e., a complete graph on the terminals) with an upper bound associated to each cut in the family. A feasible demand must not exceed any of these cut constraints. A natural generalization allows for upper bounds to be specified on any family of cuts, not necessarily nested. Can these generalizations still be approximated within a constant factor?

It may be sensible to define a directed version of tree demands, in correspondence with the asymmetric hose model. One possible way to do this is to simply orient the edges of the tree, while avoiding sinks or sources at interior nodes. More precisely: pick a subset R of the leaves, and orient all edges away from R. However, unlike with the asymmetric hose model, this model would be incomparable with the undirected tree demand model; a common generalization would be preferred.

Practical but competitive routing schemes We have seen that in some situations, oblivious routing may be much more expensive than dynamic routing. However, dynamic routing is not practical to implement. Is there any space between these extremes for a routing scheme that is somewhat practical, but competitive with dynamic routing in the worst case? This is a fairly ill-specified question: what is a "practical" routing scheme? One might ask for a routing where the information needed to determine the flow template for a particular pair is small or localized in some sense, so that the communication overhead required to implement the routing scheme is small.

This question falls within the framework of adaptive robust optimization; given that these problems are in general highly intractable, there is perhaps reason to be pessimistic. Żotkiewicz and Ben-Ameur [138] consider the possibility of partitioning the demand universe, and using different routing templates in different parts; theoretical results are however scarce. Scutellà [121] defines a model where two distinct routing templates can be supplied along with the capacity reservation; the requirement is that any feasible demand can be feasibly routed using at least one of the two templates. This kind of model provides a spectrum between oblivious and dynamic routing, by allowing more and more distinct routing templates. However, in [121] results are obtained only for a rather restricted version of this model.

Approximability of dynamic routing In [42], it is asked whether there exists a constant factor approximation algorithm for dynamic routing in the general RND model. This question is open even for the asymmetric hose model; currently, nothing better than an $O(\log n)$ factor is known. In particular, using the MPR solution as an approximation for the dynamic routing optimum can be a logarithmic factor off, as was shown in Chapter 7.

A 2-approximation for dynamic routing in the symmetric hose model follows from [58, 77] as already discussed. It is still open as to whether this is even NP-hard.

Bibliography

[1] I. Abraham, Y. Bartal, and O. Neiman. Nearly tight low stretch spanning trees. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 781–790, 2008.

[2] A. Altın, E. Amaldi, P. Belotti, and M. C. Pınar. Provisioning virtual private networks under traffic uncertainty. *Networks*, 49(1):100–115, January 2007. Preliminary version presented at 18th Int. Symp. Math. Programming, Copenhagen, 2003.

[3] A. Altın, P. Belotti, and M. C. Pınar. OSPF routing with optimal oblivious performance ratio under polyhedral demand uncertainty. *Optimization and Engineering*, November 2009.

[4] A. Altın, H. Yaman, and M. C. Pınar. A hybrid polyhedral uncertainty model for the robust network loading problem. Manuscript, April 2009.

[5] A. Altın, H. Yaman, and M. C. Pınar. The robust network loading problem under hose demand uncertainty: Formulation, polyhedral analysis, and computations. *INFORMS Journal on Computing*, March 2010.

[6] R. Andersen and U. Feige. Interchanging distance and capacity in probabilistic mappings, arXiv:0907.3631, July 2009.

[7] M. Andrews. Hardness of buy-at-bulk network design. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 115–124, 2004.

[8] M. Andrews. Private communication, 2010.

[9] M. Andrews and L. Zhang. Approximation algorithms for access network design. *Algorithmica*, 34(2):197–215, 2002. Preliminary version in Proc. 39th FOCS, 1998.

[10] M. Andrews and L. Zhang. Hardness of the undirected congestion minimization problem. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 284–293, 2005.

[11] M. Andrews and L. Zhang. Hardness of the undirected edge-disjoint paths problem. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 276–283, 2005.

[12] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In *Proceedings of ACM SIGCOMM*, pages 313–324, 2003.

[13] P. Assouad. Plongements isométriques dans L^1: aspect analytique. In G. Choquet, H. Fakhoury, and J. Saint-Raymond, editors, *Séminaire d'Initiation à l'Analyse*, number 14. Université Paris VI, 1980.

[14] Y. Aumann and Y. Rabani. An O(log k) approximate min-cut max-flow theorem and approximation algorithm. *SIAM Journal on Computing*, 27(1):291–301, 1998.

[15] B. Awerbuch and Y. Azar. Buy-at-bulk network design. In *Proceedings of the 38th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, volume 0, pages 542+, 1997.

[16] Y. Azar, E. Cohen, A. Fiat, H. Kaplan, and H. Räcke. Optimal oblivious routing in polynomial time. *Journal of Computer and System Sciences*, 69(3):383–394, November 2004.

[17] Y. Bartal. Probabilistic approximation of metric spaces and its algorithmic applications. In *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 184–193, 1996.

[18] Y. Bartal. On approximating arbitrary metrices by tree metrics. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC)*, pages 161–168, 1998.

[19] Y. Bartal and S. Leonardi. On-line routing in all-optical networks. *Theoretical Computer Science*, 221(1-2):19–39, June 1999.

[20] L. Becchetti, J. Könemann, S. Leonardi, and M. Pál. Sharing the cost more efficiently: Improved approximation for multicommodity rent-or-buy. *ACM Transactions on Algorithms*, 3(2), 2007. Preliminary version in Proc. 16th SODA, 2005.

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

[22] W. Ben-Ameur and H. Kerivin. Routing of uncertain traffic demands. *Optimization and Engineering*, 6(3):283–313, September 2005.

[23] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, August 2009.

[24] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376, March 2004.

[25] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, November 1998.

[26] A. Ben-Tal and A. Nemirovski. Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13, August 1999.

[27] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424, September 2000.

[28] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1):49–71, September 2003.

[29] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, January 2004.

[30] H.-G. Beyer and B. Sendhoff. Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering*, 196(33-34):3190–3218, July 2007.

[31] A. Bhaskara and A. Vijayaraghavan. Approximating matrix p-norms, arXiv:1001.2613, May 2010.

[32] A. Borodin and J. E. Hopcroft. Routing, merging and sorting on parallel models of computation. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing (STOC)*, pages 338–344, 1982.

[33] J. Bourgain. On Lipschitz embedding of finite metric spaces in Hilbert space. *Israel Journal of Mathematics*, 52(1):46–52, March 1985.

[34] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, March 2004.

[35] J. Byrka. An optimal bifactor approximation algorithm for the metric uncapacitated facility location problem. In *Proceedings of 10th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, volume 4627 of *Lecture Notes in Computer Science*, pages 29–43, 2007.

[36] J. Byrka, F. Grandoni, T. Rothvoß, and L. Sanità. An improved LP based approximation for Steiner tree. Accepted to the 42nd Annual ACM Symposium on Theory of Computing (STOC), 2010.

[37] G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. *SIAM Journal on Computing*, 34(2):358–372, 2005. Preliminary version in Proc. 12th SODA, 2001.

[38] C.-S. Chang, D.-S. Lee, and Y.-S. Jou. Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering. *Journal of Computer Communications*, 25(6):611–622, April 2002.

[39] M. Charikar, C. Chekuri, A. Goel, S. Guha, and S. Plotkin. Approximating a finite metric by a small number of tree metrics. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 379–388, 1998.

[40] A. Charnes and W. W. Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959.

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

[42] C. Chekuri, G. Oriolo, M. G. Scutellà, and F. B. Shepherd. Hardness of robust network design. *Networks*, 50(1):50–54, April 2007.

[43] M. Chlebík and J. Chlebíková. The Steiner tree problem on graphs: Inapproximability results. *Theoretical Computer Science*, 406(3):207–214, October 2008. Preliminary version in Proc. 8th SWAT, 2002.

[44] J. Chu and C.-T. Lea. Optimal link weights for IP-based networks supporting hose-model VPNs. *IEEE/ACM Transactions on Networking*, 17(3):778–788, 2009.

[45] M. M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Algorithms and Combinatorics. Springer, June 1997.

[46] 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. In *Proceedings of SIGCOMM*, pages 95–108, 1999.

[47] F. Eisenbrand and F. Grandoni. An improved approximation algorithm for virtual private network design. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 928–932, 2005.

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

[49] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Approximating connected facility location problems via random facility sampling and core detouring. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1174–1183, 2008.

[50] F. Eisenbrand and E. Happ. Provisioning a virtual private network under the presence of non-communicating groups. In *Proceedings of the 6th International Conference on Algorithms and Complexity*, volume 3998 of *Lecture Notes in Computer Science*, pages 105–114, 2006.

[51] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18:1035–1064, 1997.

[52] L. El Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52, 1998.

[53] M. Englert and H. Räcke. Oblivious routing for the L_p-norm. In *Proceedings of the 50th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 32–40, 2009.

[54] T. Erlebach and M. Rüegg. Optimal bandwidth reservation in hose-model VPNs with multi-path routing. In *Proceedings of INFOCOM*, pages 2275–2282, 2004.

[55] J. Fakcharoenphol, C. Harrelson, S. Rao, and K. Talwar. An improved approximation algorithm for the 0-extension problem. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 257–265, 2003.

[56] J. Fakcharoenphol, S. Rao, and K. Talwar. Approximating metrics by tree metrics. *SIGACT News*, 35(2):60–70, 2004.

[57] J. Fakcharoenphol, S. Rao, and K. Talwar. A tight bound on approximating arbitrary metrics by tree metrics. *Journal of Computer and System Sciences*, 69(3):485–497, November 2004. Preliminary version in Proc. 35th STOC, 2003.

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

[59] S. Fiorini, G. Oriolo, L. Sanità, and D. O. Theis. The VPN tree routing conjecture for outerplanar networks, arXiv:0711.2623, November 2007.

[60] S. Fiorini, G. Oriolo, L. Sanità, and D. O. Theis. The VPN problem with concave costs. Accepted to SIAM Journal on Discrete Mathematics, 2010.

[61] L. Fleischer, J. Könemann, S. Leonardi, and G. Schäfer. Simple cost sharing schemes for multicommodity rent-or-buy and stochastic Steiner tree. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC)*, pages 663–670, 2006.

[62] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[63] A. Frangioni, F. Pascali, and M. G. Scutellà. Static and dynamic routing under the single-source hose model. Technical report, Dipartimento di Informatica, Università di Pisa, 2009.

[64] N. Garg, R. Khandekar, G. Konjevod, R. Ravi, F. Salman, and A. Sinha. On the integrality gap of a natural formulation of the single-sink buy-at-bulk network design problem. In K. Aardal and B. Gerards, editors, *Proceedings of the 8th International Conference on Integer Programming and Combinatorial Optimization (IPCO)*, volume 2081 of *Lecture Notes in Computer Science*, pages 170–184, June 2001.

[65] A. Goel and D. Estrin. Simultaneous optimization for concave costs: single sink aggregation or single source buy-at-bulk. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 499–505, 2003.

[66] A. Goel and I. Post. An oblivious O(1)-approximation for single source buy-at-bulk. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 442–450, 2009.

[67] A. Goel and I. Post. One tree suffices: A simultaneous O(1)-approximation for single-sink buy-at-bulk, arXiv:1004.2291, April 2010.

[68] R. E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[69] N. Goyal, N. Olver, and B. Shepherd. The VPN conjecture is true. In *Proceedings of the 40th ACM Symposium on Theory of Computing (STOC)*, 2008.

[70] N. Goyal, N. Olver, and F. B. Shepherd. Dynamic vs. oblivious routing in network design. In *Proceedings of the 17th Annual European Symposium on Algorithms (ESA)*, volume 5757 of *Lecture Notes in Computer Science*, pages 277–288, 2009.

[71] F. Grandoni, V. Kaibel, G. Oriolo, and M. Skutella. A short proof of the VPN tree routing conjecture on ring networks. *Operations Research Letters*, 36(3):361–365, 2008.

[72] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, Berlin, 1988.

[73] M. Grötschel, C. L. Monma, and M. Stoer. Design of survivable networks. In M. O. Ball, C. L. Monma, and G. Nemhauser, editors, *Handbook in Operations Research and Management Science*, volume 7, pages 617–671. North Holland, 1995.

[74] S. Guha, A. Meyerson, and K. Munagala. A constant factor approximation for the single sink edge installation problems. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 383–388, 2001.

[75] A. Gupta. Private communication, 2004.

[76] A. Gupta, M. T. Hajiaghayi, and H. Räcke. Oblivious network design. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 970–979, 2006.

[77] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: a network design problem for multicommodity flow. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 389–398, 2001.

[78] A. Gupta and A. Kumar. Private communication, 2001.

[79] A. Gupta, A. Kumar, M. Pál, and T. Roughgarden. Approximation via cost sharing: Simpler and better approximation algorithms for network design. *Journal of the ACM*, 54(3):11, 2007. Combines results in Proc. 35th STOC, 2003, and Proc. 44th FOCS, 2003.

[80] A. Gupta, A. Kumar, and T. Roughgarden. A constant-factor approximation algorithm for the multicommodity rent-or-buy problem. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 333–344, 2002.

[81] A. Gupta, A. Kumar, and T. Roughgarden. Simpler and better approximation algorithms for network design. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 365–372, 2003.

[82] M. T. Hajiaghayi, R. D. Kleinberg, H. Räcke, and T. Leighton. Oblivious routing on node-capacitated and directed graphs. *ACM Transactions on Algorithms*, 3(4):51+, 2007.

[83] C. Harrelson, K. Hildrum, and S. Rao. A polynomial-time tree decomposition to minimize congestion. In *Proceedings of the 15th Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 34–43, 2003.

[84] P. Harsha, T. P. Hayes, H. Narayanan, H. Räcke, and J. Radhakrishnan. Minimizing average latency in oblivious routing. In *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 200–207, 2008.

[85] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(04):439–562, August 2006.

[86] C. A. J. Hurkens, J. C. M. Keijsper, and L. Stougie. Virtual private network design: A proof of the tree routing conjecture on ring networks. *SIAM Journal on Discrete Mathematics*, 21(2):482–503, 2007. Preliminary version in Proc. IPCO, 2005.

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

[88] G. F. Italiano, R. Rastogi, and B. Yener. Restoration algorithms for virtual private networks in the hose model. In *Proceedings.Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, pages 131–139, 2002.

[89] L. Jia, G. Lin, G. Noubir, R. Rajaraman, and R. Sundaram. Universal approximations for tsp, steiner tree, and set cover. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 386–395, 2005.

[90] W. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Conference in modern analysis and probability (New Haven, Conn., 1982)*, volume 26 of *Contemporary Mathematics*, pages 189–206. American Mathematical Society, 1984.

[91] A. Jüttner, I. Szabó, and Szentesi. On bandwidth efficiency of the hose resource management model in virtual private networks. In *Proceedings of IEEE INFOCOM*, pages 386–395, April 2003.

[92] C. Kaklamanis, D. Krizanc, and T. Tsantilas. Tight bounds for oblivious routing in the hypercube. In *Proceedings of the 2nd Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pages 31–36, 1990.

[93] D. Karger and M. Minkoff. Building steiner trees with incomplete global knowledge. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, volume 0, pages 613–623, 2000.

[94] A. Karzanov. Minimum 0-extensions of graph metrics. *European Journal of Combinatorics*, 19(1):71–101, January 1998.

[95] H. Kerivin and A. R. Mahjoub. Design of survivable networks: A survey. *Networks*, 46(1):1–21, 2005.

[96] L. G. Khachiyan. The problem of calculating the volume of a polyhedron is enumeratively hard. *Uspekhi Mat. Nauk*, 44(3(267)):179–180, 1989.

[97] S. Khuller, B. Raghavachari, and N. Young. Balancing minimum spanning trees and shortest-path trees. *Algorithmica*, 14(4):305–321, October 1995.

[98] M. Kodialam, T. V. Lakshman, J. B. Orlin, and S. Sengupta. Oblivious routing of highly variable traffic in service overlays and IP backbones. *IEEE/ACM Transactions on Networking*, 17(2):459–472, 2009.

[99] M. Kodialam, T. V. Lakshman, and S. Sengupta. Efficient and robust routing of highly variable traffic. In *Proceedings of Third Workshop on Hot Topics in Networks (HotNets-III)*, 2004.

[100] P. Kouvelis and G. Yu. *Robust Discrete Optimization and Its Applications (Nonconvex Optimization and Its Applications)*. Springer, 1st edition, November 1997.

[101] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener. Algorithms for provisioning virtual private networks in the hose model. *IEEE/ACM Transactions on Networking*, 10(4):565–578, August 2002. Preliminary version in Proc. SIGCOMM, 2001.

[102] F. T. Leighton, B. M. Maggs, and S. B. Rao. Packet routing and job-shop scheduling in O(congestion + dilation) steps. *Combinatorica*, 14(2):167–186, June 1994. Preliminary version in Proc. 29th FOCS, 1988.

[103] T. Leighton and S. Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, November 1999. Preliminary version in Proc. 28th FOCS, 1988.

[104] J. H. Lin and J. S. Vitter. ϵ-approximations with minimum packing constraint violation. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC)*, pages 771–782, 1992.

[105] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, June 1995.

[106] N. Linial, E. London, and Y. Rabinovich. The geometry of graphs and some of its algorithmic applications. In *Proceedings of the 35th Annual ACM Symposium on Foundations of Computer Science (FOCS)*, pages 577–591, 2004.

[107] B. Maggs, F. Meyer auf der Heide, B. Vöcking, and B. Westermann. Exploiting locality for data management in systems of limited bandwidth. In *Proceedings of the IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 284–293, 1997.

[108] A. Meddeb. Building cost effective lower layer VPNs: The ILEC/CLEC dilemma. *International Journal of Communication Systems*, 2010. Print version forthcoming.

[109] D. Mitra and R. A. Cieslak. Randomized parallel communications on an extension of the omega network. *Journal of the ACM*, 34(4):802–824, October 1987.

[110] N. Olver and F. B. Shepherd. Approximability of robust network design. In *Proceedings of the 21st ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1097–1105, 2010.

[111] G. Oriolo. Domination between traffic matrices. *Mathematics of Operations Research*, 33(1):91–96, February 2008.

[112] R. Ostrovsky and Y. Rabani. Universal $O(\text{congestion}+\text{dilation}+\log^{1+\epsilon} N)$ local control packet switching algorithms. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pages 644–653, 1997.

[113] G. Poo and H. Wang. Multi-path routing versus tree routing for vpn bandwidth provisioning in the hose model. *Computer Networks*, 51(6):1725–1743, April 2007.

[114] H. Räcke. Minimizing congestion in general networks. In *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 43–52, 2002.

[115] H. Räcke. Optimal hierarchical decompositions for congestion minimization in networks. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 255–264, 2008.

[116] H. Räcke. Survey on oblivious routing strategies. In K. Ambos-Spies, B. Löwe, and W. Merkle, editors, *Mathematical Theory and Computational Practice*, volume 5635, chapter 43, pages 419–429. Springer, Berlin, Heidelberg, 2009.

[117] R. Ravi and F. Salman. Approximation algorithms for the traveling purchaser problem and its variants in network design. In *Proceedings of the 7th Annual European Symposium on Algorithms (ESA)*, page 696, 1999.

[118] G. Robins and A. Zelikovsky. Improved steiner tree approximation in graphs. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 770–779, 2000.

[119] R. T. Rockafellar. Optimization under uncertainty. Lecture notes, 2001.

[120] T. Rothvoß and L. Sanità. On the complexity of the asymmetric VPN problem. In *Proceedings of the 12th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*, pages 326–338, 2009.

[121] M. G. Scutellà. On improving optimal oblivious routing. *Operations Research Letters*, 37(3):197–200, May 2009.

[122] A. Sebő. Private communication, 2008.

[123] F. B. Shepherd. Single-sink multicommodity flow with side constraints. In W. Cook, L. Lovász, and J. Vygen, editors, *Research Trends in Combinatorial Optimization*, chapter 20, pages 429–450. Springer, Berlin, Heidelberg, 2009.

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

[125] D. Shmoys. Approximation algorithms for facility location problems. In K. Jansen and S. Khuller, editors, *Approximation Algorithms for Combinatorial Optimization*, volume 1913 of *Lecture Notes in Computer Science*, chapter 4, pages 369–378. Springer, Berlin, June 2000.

[126] D. B. Shmoys, E. Tardos, and K. Aardal. Approximation algorithms for facility location problems. In *In Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pages 265–274, 1998.

[127] A. L. Soyster. Technical note–convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, September 1973.

[128] S. Srinivasagopalan, C. Busch, and S. S. Iyengar. An oblivious spanning tree for buy-at-bulk network design problems, arXiv:1004.0351, April 2010.

[129] C. Swamy and A. Kumar. Primal-dual algorithms for connected facility location problems. *Algorithmica*, 40(4):245–269, December 2004.

[130] K. Talwar. The single-sink buy-at-bulk lp has constant integrality gap. In *Proceedings of the 9th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 475–486, 2002.

[131] E. Tardos. A strongly polynomial algorithm to solve combinatorial linear programs. *Operations Research*, 34(2):250–256, 1986.

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

[133] L. G. Valiant and G. J. Brebner. Universal schemes for parallel communication. In *Proceedings of the 13th Annual ACM Symposium on Theory of Computing (STOC)*, pages 263–277, 1981.

[134] A. van Zuylen. Deterministic sampling algorithms for network design. *Algorithmica*, July 2009. Preliminary version in Proc. 16th ESA.

[135] D. Williamson and A. van Zuylen. A simpler and better derandomization of an approximation algorithm for single source rent-or-buy. *Operations Research Letters*, 35(6):707–712, November 2007.

[136] G. Yu and P. Kouvelis. Min-max optimization of discrete optimization problems. In D. Z. Du and P. M. Pardalos, editors, *Minimax and Its Applications*, pages 157–171. Kluwer Acadamic Publishers, 1995.

[137] R. Zhang-Shen and N. McKeown. Designing a fault-tolerant network using Valiant load-balancing. In *Proceedings of IEEE INFOCOM*, pages 2360–2368, April 2008.

[138] M. Żotkiewicz and W. Ben-Ameur. More adaptive robust stable routing. In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, pages 1–6, November 2009.