INFORMS Journal on Computing
Vol. 23, No. 1, Winter 2011, pp. 75–89
ISSN 1091-9856 | EISSN 1526-5528 | 11 | 2301 | 0075

informs®

DOI 10.1287/ijoc.1100.0380

The Robust Network Loading Problem Under Hose Demand Uncertainty: Formulation, Polyhedral Analysis, and Computations

Ayşegül Altın
Department of Industrial Engineering, TOBB University of Economics and Technology, 06560 Sögütözü, Ankara, Turkey, aaltin@etu.edu.tr

Hande Yaman, Mustafa Ç. Pınar
Department of Industrial Engineering, Bilkent University, 06800 Bilkent, Ankara, Turkey
{hyaman@bilkent.edu.tr, mustafap@bilkent.edu.tr}

We consider the network loading problem (NLP) under a polyhedral uncertainty description of traffic demands. After giving a compact multicommodity flow formulation of the problem, we state a decomposition property obtained from projecting out the flow variables. This property considerably simplifies the resulting polyhedral analysis and computations by doing away with metric inequalities. Then we focus on a specific choice of the uncertainty description, called the "hose model," which specifies aggregate traffic upper bounds for selected endpoints of the network. We study the polyhedral aspects of the NLP under hose demand uncertainty and use the results as the basis of an efficient branch-and-cut algorithm. The results of extensive computational experiments on well-known network design instances are reported.

Key words: network loading problem; polyhedral demand uncertainty; hose model; robust optimization; polyhedral analysis; branch and cut
History: Accepted by S. Raghavan, Area Editor for Telecommunications and Electronic Commerce; received February 2008; revised November 2008, May 2009, October 2009; accepted December 2009. Published online in *Articles in Advance* March 23, 2010.

1. Introduction

Consider the problem of deciding the optimal (i.e., resulting in the least total installation cost) number of devices of unit capacity to be installed on the links of the simple network in Figure 1(a) to support the communication demands between the nodes. The number on each edge gives the installation cost of a unit capacity device on that edge. Each pairwise demand is cited with its source and destination; i.e., AB is the demand from A to B, whereas BA is the demand in the reverse direction. Suppose that all communication demands except AD, DA, AE, and EA are forecasted to be one unit of traffic flow. The aforementioned four pairs are not expected to exchange any traffic, and hence these demands are zero.

Suppose that we seek a design where link capacities are sufficient to accommodate the total flow on each link in both directions and we allow multipath routing. Then, an optimal capacity installation is given in Figure 1(b) with a total cost of 13. Now suppose that the communication demands are realized to be different than expected, namely, AD, AE, BD, and BE are one unit more than forecasted, whereas AB, BA, DE, and ED are one unit fewer than forecasted. As a result, the current capacity of link CD would not be sufficient to route all traffic requests simultaneously. In telecommunications networks, such a deficiency causes a delay whose consequences become more severe as the deviation from expectations and the strategic value of the data traffic increase.

In this paper, we discuss the design of networks that can support changing communication patterns in the least costly manner. More precisely, we study the robust *network loading problem* (NLP) under a polyhedral uncertainty definition of possible traffic demands. The traditional NLP assumes that pairwise demands are known. The purpose is to determine the least costly allocation of discrete units of capacitated facilities on the links of the given network. In this work, we do not assume that demands are known a priori, but we consider a polyhedral definition of feasible demands. Our motivation for this study is to design networks robust to fluctuations in demand estimates, which are almost sure to happen in real-life applications. Hence, we want our least-cost design to remain operational for any feasible realization in a prescribed polyhedral set.

It is well accepted that data are always subject to some uncertainty in real-life problems. On some occasions researchers completely ignore uncertainty and

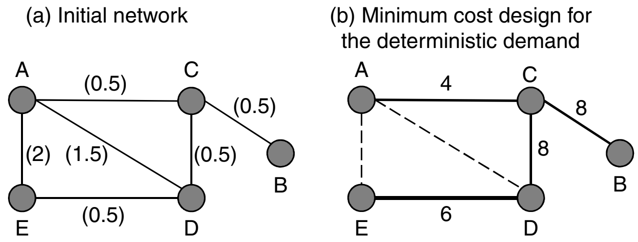

Figure 1 Example of Network Capacity Loading

use nominal values to represent the expected average behavior of the system. On the other hand, stochastic programming (SP) has been widely used to deal with uncertainty. SP yields decisions that might become infeasible with some probability, but in some cases, such a tolerance is not favorable, and robust optimization (RO) is more useful because it aims to make the best decision that remains "operational" for any realization of data within a prescribed uncertainty set. An overview of some topics in the RO domain is given by Ben-Tal and Nemirovski (2008).

In RO, one decides on an uncertainty set $\mathbb{U}$, which defines all likely data realizations for which one is willing to be prepared, without making any assumption on the stochastic model of the data. Then, a robust design is the one whose worst performance over $\mathbb{U}$ is the best. There are various ways of defining the uncertainty set $\mathbb{U}$: a set of finite/infinite number of scenarios, finite intervals, or a polyhedral or an ellipsoidal set (see, e.g., Atamtürk 2006; Atamtürk and Zhang 2007; Ben-Tal and Nemirovski 1998, 1999, 2008; Ben-Tal et al. 2004; Bertsimas and Sim 2003, 2004; Mudchanatongsuk et al. 2008; Ordoñez and Zhao 2007; Yaman et al. 2007).

An important uncertain component in network design problems is the traffic matrix, i.e., the demand between origin–destination pairs. In practice, it is not likely for network designers to have a precise estimate of the traffic matrix, and ignoring this uncertainty may lead to a failure to meet service-level agreements. To overcome this obstacle, Duffield et al. (1999) and Fingerhut et al. (1997) independently proposed a flexible model (hose model) that specifies aggregate traffic upper bounds for selected endpoints of the network. Since then, the hose model has gained significant popularity because of its ease of specification (Fingerhut et al. 1997) as well as the resource-sharing flexibility and multiplexing gains it provides (Duffield et al. 1999). The hose model is initially used to design virtual private networks (VPNs). Among these efforts, Gupta et al. (2001), Italiano et al. (2002), Grandoni et al. (2008), and Goyal et al. (2008) address the computational complexity of the resulting combinatorial optimization problems; Goyal et al. (2008) prove that the VPN design problem with fractional link capacities and single-path routing of symmetric traffic matrices can be solved in polynomial time. Similarly, Gupta et al. (2003), Kumar et al. (2001), and Swamy and Kumar (2002) develop approximation algorithms for the problem with different hose definitions. In the same vein, Ben-Ameur and Kerivin (2005) discuss the *polyhedral model*, where the feasible demand realizations are defined by an arbitrary polyhedron. They develop an iterative algorithm based on enumerating the vertices of the demand polyhedron so as to determine robust minimum-cost splittable routing and edge capacity configurations. Later, Altın et al. (2007) propose a compact mixed-integer programming model for VPN design with continuous capacity expansion under unsplittable routing along with a branch-and-price-and-cut algorithm. Their model considers all traffic matrices simultaneously. On the other hand, the growth in the size and application types in IP networks has inspired several works in this domain as well (Belotti and Pınar 2008, Altın et al. 2010).

The number of different facility types available for installation, the use of different cost functions with flow costs, and technical restrictions on the routing of demands give rise to variants of the deterministic NLP (Atamtürk and Rajan 2002; Avella et al. 2007; Berger et al. 2000; Bienstock and Günlük 1996; Bienstock et al. 1998; Günlük 1999; Brockmüller et al. 2004; Magnanti and Mirchandani 1993; Magnanti et al. 1993, 1995; Mirchandani 2000; Rardin and Wolsey 1993; van Hoesel et al. 2002). The capacity expansion problem (CEP), where the decision is to determine a capacity expansion plan for a given network, is also closely related with NLP (Atamtürk and Günlük 2007, Atamtürk and Rajan 2002, Berger et al. 2000, Bienstock and Günlük 1996, Günlük 1999).

Because NLP is strongly NP-hard, there have been various efforts for solving it as efficiently as possible through the use of alternative formulations and heuristics, and by a thorough polyhedral analysis (Magnanti and Mirchandani 1993, Magnanti et al. 1993, van Hoesel et al. 2002, Atamtürk and Günlük 2007). The most common approach in the literature to handle NLP efficiently is to define some strong valid inequalities to strengthen the linear programming relaxations. Projection of the feasible set onto the space of discrete design variables has also been a common point of interest (Atamtürk and Rajan 2002; Avella et al. 2007; Bienstock et al. 1998; Bienstock and Günlük 1996; Magnanti and Mirchandani 1993; Magnanti et al. 1993, 1995; Mirchandani 2000; Rardin and Wolsey 1993).

Because the demand between each origin–destination pair can be considered as a single commodity, NLP is of a multicommodity flow nature. Although the problem for single-commodity flow with two facility types is very well studied, and the polyhedra of feasible flows is fully characterized (Mirchandani 2000), the multicommodity flow version remains hard, and metric inequalities are used to define the projection of the corresponding polyhedron on the space of discrete design variables (Onaga and Kakusho 1971).

Against this background, the main contribution of this paper to the existing body of literature on single-stage robust NLP is to relax the assumption of known traffic demands prior to designing the network. Whereas NLP with known (deterministic) demands is well studied, the literature on robust NLP is rather limited. For the single-stage robust NLP under polyhedral uncertainty, we are not aware of any other attempt with the exception of an earlier reference by Karaşan et al. (2005), where uncertainty was incorporated into the design of fiber optic networks with an emphasis on modeling rather than on a detailed polyhedral analysis and branch and cut. On the other hand, Atamtürk and Zhang (2007) study the two-stage robust NLP, where the capacity is reserved on network links before observing the demands and the routing decision is made afterwards in the second stage. Furthermore, Mudchanatongsuk et al. (2008) study an approximation to the robust CEP with recourse, where the routing of demands (recourse variables) is limited to a linear function of demand uncertainty.

Our formulation for NLP with polyhedral uncertainty is interesting because we avoid using metric inequalities because of a decomposition property obtained from a projection on the design components. A similar projection is used in Mirchandani (2000) for deterministic single- and multicommodity NLP, where all extreme rays of the related projection cone for the single-commodity case were characterized. However, only necessary conditions were obtained for the deterministic multicommodity variant. The latter problem is difficult because the coupling bundle constraints prevent the decomposition of the problem into single-commodity subproblems. However, we bypass that difficulty by observing that we can decompose the projection problem into many smaller single-commodity problems for which the results of Mirchandani (2000) remain valid. This observation considerably simplifies the formulations, but the problem still remains difficult and requires intensive efforts for developing an efficient solution algorithm. Consequently, it opens the way to a thorough polyhedral analysis based on which we develop a branch-and-cut algorithm along with a simple but effective heuristic, and we use it to solve several well-known network design instances.

Studies on the polyhedral properties of deterministic NLP are mostly limited to the case of at most three facility types where the capacity of a facility is an integer multiple of the capacity of the smaller facility. Atamtürk (2002) gives valid inequalities for the deterministic problem with general capacity modularities and an arbitrary number of facilities. More recently, Raack et al. (2010) derive a general definition of flow-cutset inequalities as mixed-integer rounding inequalities for deterministic NLP with directed, bidirected, and undirected networks. They also consider arbitrary capacity structures for multiple facilities, where they study the facial structure of the cutset polyhedra and its relation to the deterministic NLP. The second main contribution of this paper is that we present valid inequalities for robust NLP with an arbitrary number of facilities and arbitrary capacity structures.

The rest of this paper is organized as follows. In §2 we describe our problem and give a compact mixed-integer programming formulation and its projection onto the space of design variables. We move on to the hose model in §2.2 and carry out a thorough polyhedral analysis for NLP under hose uncertainty in §3. Then we continue with separation algorithms for various valid inequalities and heuristics, all incorporated into a branch-and-cut algorithm in §4. We give a summary of our computational results in §5 and conclude in §6 with some directions for future work.

2. Problem Definition

The deterministic NLP is defined as follows. Let $G=(V,E)$ be an undirected graph where V is the set of nodes and E is the set of edges. Let Q denote the set of commodities, i.e., the set of origin–destination pairs with traffic demand. The origin of commodity $q\in Q$ is $s(q)$ and its destination is $t(q)$. A set of facility alternatives with different capacities and costs can be used to carry flow through the network. The problem is to determine the number of facilities installed on the edges such that all demand can be routed and the installation cost is minimized. Then NLP can be modelled as

$$\min \sum_{\{h,k\}\in E}\sum_{l\in L} p^l_{hk} y^l_{hk} \tag{1}$$

$$\text{s.t.} \sum_{k:\ \{h,k\}\in E}(f^q_{hk}-f^q_{kh}) = \begin{cases} 1 & h=s(q),\\ -1 & h=t(q),\\ 0 & \text{otherwise,}\end{cases} \qquad \forall h\in V,\ q\in Q, \tag{2}$$

$$\sum_{q\in Q}(f^q_{hk}+f^q_{kh})d_q \le \sum_{l\in L} C^l y^l_{hk} \quad \forall \{h,k\}\in E, \tag{3}$$

$$y^l_{hk}\ge 0 \text{ and integer} \quad \forall \{h,k\}\in E,\ l\in L, \tag{4}$$

$$f^q_{hk}, f^q_{kh}\ge 0 \quad \forall \{h,k\}\in E,\ q\in Q, \tag{5}$$

where d_q is the forecasted demand for commodity $q\in Q$, L is the set of facility alternatives, p^l_{hk} is the cost of installing one facility of type $l \in L$ on edge $\{h, k\} \in E$, and C^l is the transmission capacity of type $l \in L$ facility. Variables of the model are y^l_{hk} for the number of type $l \in L$ facilities loaded on the edge $\{h, k\} \in E$ and f^q_{hk} for the fraction of d_q routed on the edge $\{h, k\} \in E$ in the direction from h to k. Constraints (2) are the usual flow conservation constraints for each demand pair at each node. Finally, the constraints (3) are the edge capacity constraints, which ensure that the total capacity installed on each edge is enough to support the total flow on it in both directions.

2.1. Robust Network Loading Problem with Polyhedral Demands

Demand forecasts may not be precise and the realized demand is very likely to be different from what is expected. Our aim is to design a network that is viable for any demand realization in the polyhedral set $D = \{d \in \mathbb{R}^{|Q|}: Ad \le \alpha, d \ge 0\}$, where $A \in \mathbb{R}^{m\times|Q|}$ and $\alpha \in \mathbb{R}^m$. We assume that D is bounded and nonempty. This leads to the following polyhedral NLP model (NLP_{POL}):

$$\min \sum_{\{h,k\}\in E} \sum_{l\in L} p^l_{hk} y^l_{hk}$$
$$\text{s.t. } (2),\ (4),\ (5),$$
$$\max_{d\in D} \sum_{q\in Q} (f^q_{hk} + f^q_{kh}) d_q \le \sum_{l\in L} C^l y^l_{hk} \quad \forall \{h, k\} \in E. \tag{6}$$

Unlike the deterministic case, NLP_{POL} is a semi-infinite optimization model as a result of the infinite number of inequalities we need to consider over the demand polyhedron for each edge $\{h, k\} \in E$. However, following the method commonly used in robust optimization (see, e.g., Altın et al. 2007, Ben-Tal and Nemirovski 1999, Bertsimas and Sim 2003), we can give a compact linear mixed-integer programming (MIP) formulation for NLP_{POL}. In NLP_{POL}, for a given flow vector f and an edge $\{h, k\} \in E$, the worst-case capacity requirement can be found by solving

$$\max \sum_{q\in Q} (f^q_{hk} + f^q_{kh}) d_q \tag{7}$$
$$\text{s.t. } \sum_{q\in Q} a^q_z d_q \le \alpha_z \quad \forall z = 1, \ldots, m, \tag{8}$$
$$d_q \ge 0 \quad \forall q \in Q. \tag{9}$$

Notice that (7)–(9) is a linear programming model and its dual is

$$\min \sum_{z=1}^{m} \alpha_z \lambda^{hk}_z \tag{10}$$
$$\text{s.t. } \sum_{z=1}^{m} a^q_z \lambda^{hk}_z \ge f^q_{hk} + f^q_{kh} \quad \forall q \in Q, \tag{11}$$
$$\lambda^{hk}_z \ge 0 \quad \forall z = 1, \ldots, m, \tag{12}$$

where λ^{hk}_z is the dual variable corresponding to (8). Since (7)–(9) is feasible and bounded, we can use a duality transformation similar to the one of Soyster (1973). Hence for each edge $\{h, k\} \in E$, we can replace (6) with

$$\sum_{l\in L} C^l y^l_{hk} \ge \min\left\{\sum_{z=1}^{m} \alpha_z \lambda^{hk}_z\text{: (11) and (12)}\right\}.$$

Then, we can omit the min since we try to minimize the sum of the design variables y^l_{hk} with nonnegative weights. Hence, assuming that demand is subject to polyhedral uncertainty, NLP_{POL} can be reformulated as the following linear MIP model (NLP_{GD}):

$$\min \sum_{\{h,k\}\in E} \sum_{l\in L} p^l_{hk} y^l_{hk} \tag{13}$$
$$\text{s.t. } (2),\ (4),\ (5),$$
$$\sum_{z=1}^{m} \alpha_z \lambda^{hk}_z \le \sum_{l\in L} C^l y^l_{hk} \quad \forall \{h, k\} \in E, \tag{14}$$
$$f^q_{hk} + f^q_{kh} \le \sum_{z=1}^{m} a^q_z \lambda^{hk}_z \quad \forall q \in Q,\ \{h, k\} \in E, \tag{15}$$
$$\lambda^{hk}_z \ge 0 \quad \forall z = 1, \ldots, m,\ \{h, k\} \in E. \tag{16}$$

As there is no flow cost in our model, we can obtain a formulation of our problem in the space of $\lambda \in \mathbb{R}^{m|E|}$ and design variables $y \in \mathbb{Z}^{|L||E|}$. Mirchandani (2000) characterized all extreme rays of the projection cone related to the single-commodity NLP. However, only necessary conditions for the multicommodity variant are given. In this case, the resulting projection inequalities are the well-known metric inequalities.

Although we do not provide the complete machinery of the projection process, we note here a particular decomposition property for NLP_{GD}. Observe that after the duality transformation we have used above, there are no constraint bundling flow variables associated with different commodities in NLP_{GD}. Hence, the existence of a multicommodity flow f can be certified by checking the existence of $|Q|$ single-commodity flows; i.e., the projection cone for the multicommodity problem can be decomposed into $|Q|$ cones with one cone for each commodity $q \in Q$. Based on this observation and using the extreme rays mentioned in Mirchandani (2000) for the single-commodity problem, we obtain the following mathematical model (NLP_{PRO}) in the space of λ and y variables:

$$\min \sum_{e\in E} \sum_{l\in L} p^l_e y^l_e$$
$$\text{s.t. } (4),\ (14),\ (16),$$
$$\sum_{z=1}^{m} a^q_z \lambda^e_z \ge 0 \quad \forall e \in E, \forall q \in Q, \tag{17}$$

$$\sum_{e\in\delta(S)}\sum_{z=1}^{m} a_z^q \lambda_z^e \geq 1 \qquad \forall\, q\in Q,\ S\subset V\colon s(q)\in S,\ t(q)\in V\backslash S, \tag{18}$$

where (17) and (18) are the related projection inequalities. We denote an edge $\{h,k\}$ as e when there is no need to specify its endpoints. For $S\subset V$, $\delta(S)$ denotes the set of edges with only one endpoint in S.

To conclude this section, we remark that model NLP_{GD} has an interesting property. Consider the case where $D=\{d\in\mathbb{R}^{|Q|}\colon Id=\alpha,\ d\geq 0\}$ and I is an identity matrix of size $|Q|$. Note that this corresponds to the deterministic case where $d_q=\alpha_q$ for each $q\in Q$. For this particular definition of D, constraints (14)–(16) in the model NLP_{GD} become

$$\sum_{q\in Q} d_q\lambda_q^{hk} \leq \sum_{l\in L} C^l y_{hk}^l \quad \forall\{h,k\}\in E,$$

$$f_{hk}^q + f_{kh}^q \leq \lambda_q^{hk} \quad \forall\, q\in Q,\ \{h,k\}\in E.$$

Here, the variable $d_q\lambda_q^{hk}$ can be interpreted as the capacity on edge $\{h,k\}\in E$ allocated to commodity $q\in Q$. Rardin and Wolsey (1993) use similar variables to express the flow requirements using cut constraints and obtain an extended formulation for the uncapacitated fixed-charge network flow problem. Then they project out these variables and obtain the so-called "dicut collection inequalities." Labbé and Yaman (2004) do a similar analysis on the flow formulations for the uncapacitated hub location problem and show that the family of dicut collection inequalities contains the metric inequalities. Notice that for a general demand polyhedron D, in our model, the variables λ_z^{hk} are not additional variables that are used to get an extended formulation; rather, they come out of the duality transformation that is used to convert the semi-infinite optimization model NLP_{POL} to a mixed-integer programming model NLP_{GD}. Still, the same duality transformation results in a system where flow variables related to different commodities are not bundled together any more and permits the use of cut inequalities to model the flow requirements as we did in NLP_{PRO}.

2.2. The Hose Demand Uncertainty Case

Duffield et al. (1999) proposed the hose model to carry out flexible resource management in VPN. Independently, Fingerhut et al. (1997) discuss the same flexible specification of nonsimultaneous traffic requirements for a more effective design of broadband networks. Since then, the hose model has become popular in the telecommunications community. Rather than the point-to-point demand estimations, it uses the traffic bandwidth of some special nodes called *VPN terminals* to characterize the feasible demand matrix realizations. The difficulty of the VPN design problem (with continuous link capacities) depends on the bandwidth definition (*symmetric, asymmetric*, and *sum-symmetric*) and the technical constraints on the routing scheme (*single-path, multipath, tree,* and *terminal tree* routing). An intriguing question is the complexity of the symmetric case with single-path routing. Hurkens et al. (2007) prove that it can be solved in polynomial time if the backbone network of the VPN is a circuit. However, NLP with symmetric demands remains a challenging problem as our test results in §5 show. In the rest of this paper, we consider the following symmetric hose model of demand uncertainty:

$$D_{\mathrm{hose}} = \left\{ d\in\mathbb{R}^{|Q|}\colon \sum_{q\in Q\colon\ s(q)=i\ \text{or}\ t(q)=i} d_q \leq b_i\ \forall\, i\in W,\quad d_q\geq 0\ \forall\, q\in Q \right\}, \tag{19}$$

where $W\subseteq V$ is the set of VPN terminals; i.e., $W=\{i\in V\colon \exists q\in Q \text{ with } s(q)=i \text{ or } t(q)=i\}$ and b_i is the bandwidth capacity of the terminal node $i\in W$. In the classical symmetric model; demand is undirected; i.e., the demand from s to t is equal to the demand from t to s. However, in (19), we allow directed demand as long as the cumulative bounds are respected.

The importance of the hose model can be demonstrated by returning to the simple example in Figure 1, where we consider a single-facility type with unit capacity. Recall that the optimal capacity allocation would be as shown in Figure 1(b) with a total cost of 13 when the demands are assumed to be known. Now consider the corresponding hose model where the bandwidth of nodes from A to E are 4, 8, 8, 6, and 6 units, respectively. Then, the optimal design for the hose polyhedron is as shown in Figure 2(a) with a total cost of 15. Notice that even though the total design cost has increased slightly, the polyhedral design is more robust to fluctuations in demand. Recall the scenario we discussed in §1 where some of the pairwise demands have deviated by one unit from their expectations. Although the deterministic design fails in that case, the robust one in Figure 2(a) remains operational.

Next, consider the demand uncertainty definition that we call the BS model, developed by Bertsimas and

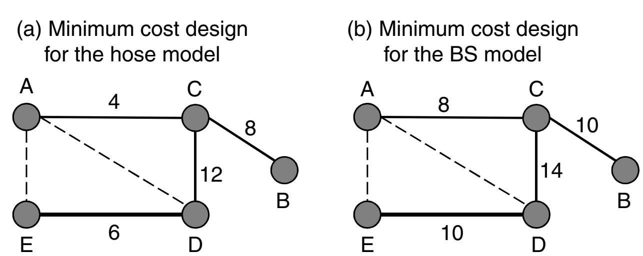

Figure 2 **Minimum Cost Robust Designs**

Sim (2003), where each demand d_q takes a value in the range $[\bar{d}_q - \hat{d}_q, \bar{d}_q + \hat{d}_q]$ such that at most Γ commodities would attain their maximum values. For the example above, we let the mean demand estimations $\bar{d}_q$ and deviations $\hat{d}_q$ be one unit so that both the expected and realized demand matrices belong to the demand polyhedron. Then, even for the not-so-conservative case with $\Gamma = 2$, the optimal design is as in Figure 2(b) with a total cost of 22. Although this design also remains operational for the aforementioned scenario, it leads to a significant increase in the design cost. An increase in the total design cost is a natural consequence of having a robust design. We provide some experimental results on this issue later in §5. However, this example shows that the hose model can be more advantageous than some other uncertainty definitions. The hose model enables the transfer of unused capacity for a pairwise demand to another demand that goes beyond its estimation. Hence, capacities of edges for the hose model can be less than required by the point-to-point pipes as a result of statistical multiplexing.

The next proposition gives a formulation of NLP under hose demand uncertainty.

Proposition 2.1. *The projection of NLP_{GD} onto the space of (λ, y) variables for the hose model (NLP_{hose}) is as follows*:

$$\min \sum_{e\in E}\sum_{l\in L} p_e^l y_e^l$$

$$\text{s.t.} \sum_{i\in W} b_i \lambda_i^e \le \sum_{l\in L} C^l y_e^l \quad \forall\, e\in E, \tag{20}$$

$$\sum_{e\in\delta(S)} (\lambda_{s(q)}^e + \lambda_{t(q)}^e) \ge 1$$

$$\forall\, q\in Q,\, S\subset V\colon s(q)\in S,\, t(q)\in V\backslash S, \tag{21}$$

$$y_e^l \ge 0 \textit{ and integer} \quad \forall\, e\in E,\, l\in L,$$

$$\lambda_i^e \ge 0 \quad \forall\, i\in W,\, e\in E.$$

3. Polyhedral Analysis

In this section we present results on the facets of the polyhedron associated with the network loading problem under hose uncertainty NLP_{hose}. In the sequel, we assume that C^l is a positive integer for $l\in L$ and that the set L is ordered such that for l_1 and l_2 in L such that $l_1 < l_2$ we have $C^{l_1} < C^{l_2}$. Let $F = \{(\lambda, y)\in \mathbb{R}_+^{|W||E|}\times \mathbb{Z}_+^{|E||L|}$: (20) and (21)$\}$ and $P = \text{conv}(F)$. Observe that adding constraints

$$\lambda_i^e \le 1 \quad \forall\, i\in W,\, e\in E \tag{22}$$

does not change the validity of the model when the costs are nonnegative (see Karaşan et al. 2005). Let $F' = F\cap\{(\lambda, y)\in \mathbb{R}_+^{|W||E|}\times \mathbb{Z}_+^{|E||L|}$: (22)$\}$ and $P' = \text{conv}(F')$.

First, we investigate the dimension of the polyhedra P and P'.

The proofs of all the results presented in this section as well as two lemmas are given in the Online Supplement at http://joc.pubs.informs.org/ecompanion.html.

Proposition 3.1. *The dimension of P and P' is* $(|W|+|L|)|E|$.

Proof. See the Online Supplement. □

3.1. Projection onto the Subspace of λ

Let $F_\lambda = \text{Proj}_\lambda(F) = \{\lambda \in \mathbb{R}_+^{|W||E|}$: (21)$\}$ and $F'_\lambda = \text{Proj}_\lambda(F') = F_\lambda \cap \{\lambda\in\mathbb{R}_+^{|W||E|}$: (22)$\}$. Now, we relate facet defining inequalities of F_λ and F'_λ with those of P and P'.

Proposition 3.2. *Inequality $\sigma\lambda \ge \sigma_0$ is facet defining for P (respectively, for P') if and only if it is facet defining for F_λ (respectively, for F'_λ).*

Proof. See the Online Supplement. □

3.2. Projection into the Subspace of (λ_e, y_e)

For $e\in E$, define $F_e = \{(\lambda^e, y_e)\in\mathbb{R}_+^{|W|}\times\mathbb{Z}_+^{|L|}$: (20)$\}$, $P_e = \text{conv}(F_e)$, $F'_e = F_e\cap\{(\lambda^e, y_e)\in\mathbb{R}_+^{|W|}\times\mathbb{Z}_+^{|L|}$: (22)$\}$, and $P'_e = \text{conv}(F'_e)$. Observe that if $\delta(S)\backslash\{e\}\ne\varnothing$ for every $S\subset V$ such that there exists $q\in Q$ with $s(q)\in S$ and $t(q)\in V\backslash S$, then $F_e = \text{Proj}_{(\lambda^e, y_e)}(F)$ and $F'_e = \text{Proj}_{(\lambda^e, y_e)}(F')$. In the following theorem, we investigate how the facet-defining inequalities of P_e and P'_e are related to those of P and P'.

Theorem 3.1. *Let $e\in E$ be such that $\delta(S)\backslash\{e\}\ne\varnothing$ for every $S\subset V$ such that there exists $q\in Q$ with $s(q)\in S$ and $t(q)\in V\backslash S$. Inequality $\alpha\lambda^e + \beta y_e \ge \gamma$ is facet defining for P_e (respectively, for P'_e) if and only if it is facet defining for P (respectively, for P').*

Proof. See the Online Supplement. □

3.3. Projection into the Subspace of Design Variables Associated with the Edges of a Cut

For $S\subseteq V$, define $b(S) = \sum_{i\in S\cap W} b_i$ and $B(S) = \min\{b(S), b(V\backslash S)\}$. Notice that in the worst case all terminals in $S\subset V$ would want to use all of their bandwidths to exchange traffic with the nodes in $V\backslash S$. As a result, the worst-case traffic on the cut $\delta(S)$ would be the minimum of these requirements, i.e., $B(S)$ (see Gupta et al. 2001, Karaşan et al. 2005).

Let $S\subset V$ be such that the subgraphs induced by S and $V\backslash S$ are both connected. Let $y_{\delta(S)}$ be the restriction of the vector y to edges $e\in\delta(S)$, $F(S) = \{y_{\delta(S)}\in\mathbb{Z}_+^{|\delta(S)||L|}\colon \sum_{l\in L}\sum_{e\in\delta(S)} C^l y_e^l \ge B(S)\}$, and $P(S) = \text{conv}(F(S))$.

Proposition 3.3. *Let $S\subset V$ be such that the subgraphs induced by S and $V\backslash S$ are both connected and $B(S) > 0$. $F(S) = \text{Proj}_{y_{\delta(S)}}(F) = \text{Proj}_{y_{\delta(S)}}(F')$.*

Proof. See the Online Supplement. □

Now, we can relate facet-defining inequalities of $P(S)$ to those of P.

Theorem 3.2. *Let $S \subset V$ be such that the subgraphs induced by S and $V\backslash S$ are both connected and $B(S) > 0$. If inequality $\sum_{l\in L}\sum_{e\in\delta(S)} \beta_e^l y_e^l \geq \beta_0$ is facet defining for $P(S)$, and for each $e' \in \delta(S)$ there exists a vector $y_{\delta(S)} \in F(S)$ such that $\sum_{l\in L}\sum_{e\in\delta(S)} \beta_e^l y_e^l = \beta_0$ and $\sum_{l\in L} C^l y_{e'}^l > B(S)$, then the inequality is facet defining for P.*

Proof. See the Online Supplement. □

3.4. Cutset and Residual Capacity Inequalities

Now, we modify two well-known families of valid inequalities for NLP to render them valid for our problem. These inequalities are the cutset inequalities and arc residual capacity inequalities (see, e.g., Magnanti et al. 1993). Both inequalities can be generated as mixed-integer rounding (MIR) inequalities. Let $X = \{(x_1, x_2) \in \mathbb{R}_+ \times \mathbb{Z}\colon x_1 + x_2 \geq \mu\}$. The MIR inequality $x_1 \geq (\mu - \lfloor \mu \rfloor)(\lceil \mu \rceil - x_2)$ is valid for X (see, e.g., Wolsey 1998, Cornuéjols 2008).

The special cases of the cutset and residual capacity inequalities for the network loading problem under hose uncertainty with a single-facility type are presented and used in Karaşan et al. (2005) to strengthen the linear programming (LP) relaxation bound.

The set $F(S)$ is an integer knapsack cover set. Its convex hull is a special case of the single-commodity multifacility cutset polyhedron studied in Atamtürk (2002). Yaman (2007) gives a family of valid inequalities called the "lifted rounding inequalities" for the integer knapsack cover set. These inequalities generalize the cutset inequalities and are special cases of the multifacility cutset inequalities of Atamtürk (2002). As they are valid for $P(S)$, they are also valid for P and P'.

For $S \subset V$ and $l \in L$, let

$$Y^l(S) = \sum_{e\in\delta(S)} y_e^l,$$

$$r^l(S) = b(S) - \left\lfloor \frac{b(S)}{C^l} \right\rfloor C^l,$$

and

$$R^l(S) = B(S) - \left\lfloor \frac{B(S)}{C^l} \right\rfloor C^l.$$

For l_1 and l_2 in L, let

$$g(l_1, l_2) = C^{l_1} - \left\lfloor \frac{C^{l_1}}{C^{l_2}} \right\rfloor C^{l_2}.$$

Proposition 3.4. *For $S \subset V$ and $l^* \in L$ such that $R^{l^*}(S) > 0$, the cutset inequality*

$$\sum_{l\in L:\ C^l<B(S)} \left(R^{l^*}(S) \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor + \min\{g(l,l^*), R^{l^*}(S)\} \right) Y^l(S) + \sum_{l\in L:\ C^l\geq B(S)} R^{l^*}(S) \left\lceil \frac{B(S)}{C^{l^*}} \right\rceil Y^l(S) \geq R^{l^*}(S) \left\lceil \frac{B(S)}{C^{l^*}} \right\rceil \quad (23)$$

is valid for P and P'.

Inequality (23) is obtained from the inequality $Y^{l^*}(S) \geq \lceil B(S)/C^{l^*} \rceil$ using sequence-independent lifting in Yaman (2007). The same inequality can be obtained as MIR inequality.

Yaman (2007) proves that if $C^1 = 1$, then the cutset inequality (23) for $l^* \in L$ such that $R^{l^*}(S) > 0$ is facet defining for $P(S)$. Using Theorem 3.2, we can state the following proposition.

Proposition 3.5. *Let $S \subset V$ be such that the subgraphs induced by S and $V\backslash S$ are both connected, and last $l^* \in L$ be such that $R^{l^*}(S) > 0$. If $C^1 = 1$, then the cutset inequality* (23) *is facet defining for P.*

Proof. See the Online Supplement. □

Notice that if $C^2, \ldots, C^{|L|}$ are divisible by C^1, then we can scale the b_s values and the C^l values by dividing with C^1 so that $C^1 = 1$. Moreover, if $|L| = 1$ and $R^1(S) > 0$, then the cutset inequality (23) is facet defining for P for $S \subset V$ such that the subgraphs induced by S and $V\backslash S$ are both connected.

Next, we generate residual capacity inequalities as MIR inequalities.

Proposition 3.6. *Let $e \in E$, $l^* \in L$, and $S \subseteq W$ be such that $r^{l^*}(S) > 0$. The residual capacity inequality*

$$\sum_{l\in L} \left(r^{l^*}(S) \left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor + \min\{g(l, l^*), r^{l^*}(S)\} \right) y_e^l + \sum_{i\in S} b_i(1 - \lambda_i^e) \geq r^{l^*}(S) \left\lceil \frac{b(S)}{C^{l^*}} \right\rceil \quad (24)$$

is valid for P'.

Proof. See the Online Supplement. □

If $|L| = 1$, the residual capacity inequality becomes

$$r^1(S) y_e^1 + \sum_{i\in S} b_i(1 - \lambda_i^e) \geq r^1(S) \left\lceil \frac{b(S)}{C^1} \right\rceil. \quad (25)$$

Magnanti et al. (1993) prove the following: if $\lceil b(S)/C^1 \rceil \geq 2$, then this inequality defines a facet of P'_e. If $\lceil b(S)/C^1 \rceil = 1$, then the inequality defines a facet of P'_e if $|S| = 1$. Using Theorem 3.1, we can prove the following.

Corollary 3.1. *Let $e \in E$ be such that $\delta(S')\backslash\{e\} \neq \varnothing$ for every $S' \subset V$ such that there exists $q \in Q$ with $s(q) \in S'$ and $t(q) \in V\backslash S'$. Suppose that $|L| = 1$ and let $S \subseteq W$ be such that $r^1(S) > 0$. The residual capacity inequality* (25) *defines a facet of P' if $\lceil b(S)/C^1 \rceil \geq 2$ or if $\lceil b(S)/C^1 \rceil = 1$ and $|S| = 1$.*

4. Branch-and-Cut Algorithm

Because we have an exponential number of constraints (21) in NLP_{hose}, we use a branch-and-cut (B&C) algorithm, which starts with a larger feasible set $\{(\lambda, y) \in \mathbb{R}_+^{|W||E|} \times \mathbb{Z}_+^{|E||L|}\colon (20)\}$ and adds the violated inequalities iteratively. In this section, we first explain our separation algorithms for the feasibility cuts (21), as well as the demand cutset (23) and residual capacity (24) inequalities. Then, we briefly describe our upper bounding procedure.

4.1. Separation of Feasibility Cuts

Inequalities (21) can be separated by solving minimum cut problems. Given a pair $(\bar{\lambda}, \bar{y})$, we construct an auxiliary graph $\bar{G}_q = (V, E)$ for each commodity $q \in Q$ such that the capacity of each edge $e \in E$ is set to be $\bar{\lambda}^e_{s(q)} + \bar{\lambda}^e_{t(q)}$. If the capacity of the minimum cut $C(q)$ separating $s(q)$ and $t(q)$ is less than one, then we have a violated inequality (21) for commodity q. Otherwise, no inequality (21) is violated for q by the pair $(\bar{\lambda}, \bar{y})$. Hence, we add at most $|Q|$ feasibility cuts at each iteration.

4.2. Separation of Demand Cutset Inequalities

We have a heuristic separation algorithm for (23). For each commodity $q \in Q$, we use the cut $C(q)$ for which a feasibility cut (21) is violated. If the pair $(\bar{\lambda}, \bar{y})$ also violates a demand cutset inequality for $C(q)$ and the facility type $l^* \in L$, then we add the corresponding cut to the problem. Thus, we add at most $|Q||L|$ such inequalities at each iteration.

4.3. Separation of Residual Capacity Inequalities

We do not know any polynomial-time algorithm to separate inequalities (24), but we can separate a relaxed version of these inequalities in polynomial time. Let $e \in E$, $l^* \in L$, and $S \subseteq W$. Define the relaxed residual capacity inequality as

$$\sum_{l \in L}\left(C^l - (C^{l^*} - r^{l^*}(S))\left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor\right) y^l_e + \sum_{i \in S} b_i (1 - \lambda^e_i) \geq r^{l^*}(S)\left\lceil \frac{b(S)}{C^{l^*}} \right\rceil, \tag{26}$$

which is valid for P' as it is implied by inequality (24). Moreover, it is a MIR inequality.

For a given edge $e \in E$, a facility type $l^* \in L$, and a pair $(\bar{\lambda}^e, \bar{y}_e)$, finding a violated relaxed residual capacity inequality or showing that there is no such inequality is equivalent to solving the problem

$$\phi(e, l^*) = \min_{S \subseteq W}\left\{\sum_{i \in S} b_i (1 - \bar{\lambda}^e_i) - r^{l^*}(S) \cdot \left(\left\lceil \frac{b(S)}{C^{l^*}} \right\rceil - \sum_{l \in L}\left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \bar{y}^l_e\right)\right\}.$$

If $\sum_{l \in L}(C^l - C^{l^*}\lfloor C^l/C^{l^*} \rfloor)\bar{y}^l_e + \phi(e, l^*) \geq 0$, then $(\bar{\lambda}^e, \bar{y}_e)$ satisfies all (26) for $e \in E$ and $l^* \in L$. Otherwise, we have a violated relaxed residual capacity inequality defined by a minimizing set S. Since (26) is a MIR inequality, if $\sum_{l \in L}\lfloor C^l/C^{l^*} \rfloor \bar{y}^l_e \geq \lceil b(S)/C^{l^*} \rceil$ or $\sum_{l \in L}\lfloor C^l/C^{l^*} \rfloor \bar{y}^l_e \leq \lceil b(S)/C^{l^*} \rceil - 1$, it cannot be violated. This is because it would be dominated by $\sum_{i \in S} b_i (1 - \lambda^e_i) \geq 0$ and $\sum_{l \in L} C^l y^l_e + \sum_{i \in S}(1 - \lambda^e_i) b_i \geq b(S)$ otherwise. Then, using the arguments in Atamtürk and Rajan (2002), we can show that the relaxed residual capacity inequalities can be separated in the following way. For each $e \in E$ and $l^* \in L$, we construct the minimizing set

$$S(e, l^*) = \left\{ i \in W\colon \bar{\lambda}^e_i > \sum_{l \in L}\left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \bar{y}^l_e - \left\lfloor \sum_{l \in L}\left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \bar{y}^l_e \right\rfloor \right\},$$

and let

$$\Psi(S(e, l^*)) = \sum_{i \in S(e, l^*)} b_i (1 - \bar{\lambda}^e_i) - r^{l^*}(S(e, l^*)) \cdot \left(\left\lceil \frac{b(S(e, l^*))}{C^{l^*}} \right\rceil - \sum_{l \in L}\left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \bar{y}^l_e\right).$$

Note that $S(e, l^*)$ includes nodes with negative objective function coefficients in the separation problem (Atamtürk and Rajan 2002). Consequently, (26) for edge $e \in E$, facility type $l^* \in L$, and the set $S(e, l^*)$ is violated if $\lfloor \sum_{l \in L}\lfloor C^l/C^{l^*} \rfloor \bar{y}^l_e \rfloor < b(S(e, l^*))/C^{l^*} < \lceil \sum_{l \in L}\lfloor C^l/C^{l^*} \rfloor \bar{y}^l_e \rceil$ and $\sum_{l \in L}(C^l - C^{l^*}\lfloor C^l/C^{l^*} \rfloor)\bar{y}^l_e + \Psi(S(e, l^*)) < 0$, where the former condition ensures that $S(e, l^*)$ characterizes a feasible solution to the separation problem. Otherwise, no inequality (26) for this $e \in E$ and $l^* \in L$ is violated. Hence, for a given edge $e \in E$ and facility type $l^* \in L$, the separation of the relaxed residual capacity inequalities can be done in $O(|W|)$ time. This means that the complexity of the overall algorithm is $O(|W||E||L|)$.

We use Algorithm 1 to separate the relaxed residual capacity inequalities. Note that we solve the separation problem for the relaxed inequalities but add the stronger ones in case of a violation. Another alternative is to use a hybrid separation method, where for each edge e and facility type l^*, we check if any strong residual capacity inequality is violated for the set $S(e, l^*)$. We have implemented both methods and observed that the former method is as efficient as the latter one. Hence, we use the former method displayed in Algorithm 1 for the relaxed inequalities.

Algorithm 1 (Residual capacity inequality separation)

for all edge $e \in E$ **do**
  for all facility type $l^* \in L$ **do**
    $\bar{Y}^{l^*}_e := \sum_{l \in L}\left\lfloor \frac{C^l}{C^{l^*}} \right\rfloor \bar{y}^l_e$
    $S(e, l^*) := \{i \in W\colon \bar{\lambda}^e_i > \bar{Y}^{l^*}_e - \lfloor \bar{Y}^{l^*}_e \rfloor\}$

$$\Psi(S(e,l^*)) = \sum_{i\in S(e,l^*)} b_i(1-\bar\lambda_i^e) - r^{l^*}(S(e,l^*)) \cdot\left(\left\lceil \frac{b(S(e,l^*))}{C^{l^*}}\right\rceil - \sum_{l\in L}\left\lfloor\frac{C^l}{C^{l^*}}\right\rfloor \bar y_e^l\right)$$

if $\lfloor \bar Y_e^{l^*}\rfloor < \dfrac{b(S(e,l^*))}{C^{l^*}} < \lceil \bar Y_e^{l^*}\rceil$ and

$\sum_{l\in L}\left(C^l - C^{l^*}\left\lfloor\frac{C^l}{C^{l^*}}\right\rfloor\right)\bar y_e^l + \Psi(S(e,l^*)) < 0$ **then**

Add the violated residual capacity inequality

$$\sum_{l\in L}\left(r^{l^*}(S(e,l^*))\left\lfloor\frac{C^l}{C^{l^*}}\right\rfloor + \min\{g(l,l^*), r^{l^*}(S(e,l^*))\}\right)y_e^l + \sum_{i\in S(e,l^*)} b_i(1-\lambda_i^e) \ge r^{l^*}(S(e,l^*))\left\lceil\frac{b(S(e,l^*))}{C^{l^*}}\right\rceil.$$

4.4. Heuristics

Given the difficulty of the problem, we expect it to be useful to incorporate approximation heuristics into our B&C algorithm. These algorithms yield easy-to-compute upper bounds, useful especially for the large instances that are relatively more difficult to solve.

We apply a simple rounding heuristic to get upper bounds on the optimal solution. Thus, at each node of the B&C tree, if we cannot find any violated inequality, then we have a feasible solution for the LP relaxation of the NLP_{hose} problem. Let $(\bar\lambda, \bar y)$ be the current fractional solution. We simply generate a feasible solution $(\bar\lambda, \hat y)$ such that $\hat y_e^l = \lceil \bar y_e^l\rceil$ for all $e\in E$ and $l\in L$. Bienstock et al. (1998) also use a similar method and mention that it is efficient.

We have also adapted the approximation algorithm of Gupta et al. (2001) for designing VPNs with continuous capacity reservation to our problem. However, based on some preliminary tests we chose to use the rounding heuristic.

5. Experimental Results

In this section we report the results of a computational study for NLP_{hose} with a single facility and with two facilities. Let $NLP_{\text{GD}}^{\text{hose}}$ be the NLP_{GD} model for the hose uncertainty definition, which we solve using ILOG CPLEX. Then, we compare our B&C algorithm with CPLEX on instances from the network design literature. The instances *polska, dfn, newyork, france, janos, atlanta, tai, nobel-eu, pioro, gui39, cost266, norway,* and *sun* are from the SND website (Zuse-Institute Berlin), whereas the remaining seven instances are the ones used in Altın et al. (2007) for a VPN design problem. For the SND instances the average pairwise demand estimates d_q are available. Hence, to generate an initial hose polyhedron, we let the bandwidth of each terminal node be the total demand incident to it; i.e., $b_i = \sum_{q\in Q:\, s(q)=i \text{ or } t(q)=i} d_q$ for all $i\in W$. Naturally, this is an assumption we make to construct an initial hose polyhedron. The choice of most effective bandwidth values is beyond the scope of the current study. However, we discuss the sensitivity of the routing performance to the choice of bandwidth values in §5.3. Moreover, we compare the hose model and the BS model in §5.1. For the latter model, we consider the interval $[d_q/1.2, 1.2d_q]$ for each commodity $q\in Q$.

We have used AMPL to model $NLP_{\text{GD}}^{\text{hose}}$ as well as CPLEX 9.1 MIP solver to solve it. The B&C algorithm is implemented in C using MINTO (Mixed INTeger Optimizer; see Nemhauser et al. 1994) and CPLEX 9.1 as LP solver. We have set a two-hour time limit both for AMPL and MINTO. The branching rule for the B&C algorithm is to choose the integer variable with fractional part closest to 0.5. Node selection is done using best-bound search. We discuss our results for single- and two-facility cases in §§5.1 and 5.2, respectively. See also the Online Supplement for detailed test results.

5.1. Single-Facility NLP_{hose}

Here, we assume that there is only one type of facility available with a capacity of C units. Then the demand cutset inequalities (23) reduce to

$$Y^1(S) \ge \left\lceil\frac{B(S)}{C}\right\rceil \quad \forall S\subset V, \tag{27}$$

which ensure that the total capacity across a cut is sufficient to support the total demand between all terminal pairs whose endpoints are on different shores of the cut. Moreover, the residual capacity inequalities are

$$\sum_{i\in S\cap W}\frac{b_i}{C}(1-\lambda_i^e) \ge \left(\frac{b(S)}{C} - \left\lfloor\frac{b(S)}{C}\right\rfloor\right)\left(\left\lceil\frac{b(S)}{C}\right\rceil - y_e\right) \quad \forall S\subset V,\ e\in E. \tag{28}$$

Notice that the inequalities (24) and (26) are identical for the single-facility case. Thus, we implement an exact separation algorithm for the residual capacity inequalities (24).

First, we compare our B&C algorithm with solving the single facility $NLP_{\text{GD}}^{\text{hose}}$ using CPLEX. We use the demand cutset inequalities (27) and the arc residual capacity inequalities (28) together with the feasibility cuts (21) in our B&C algorithm.

We could solve 7 out of 18 instances to optimality in two hours using both CPLEX and B&C. Figure 3(a) shows the change in solution time as a result of using our B&C algorithm rather than CPLEX to solve these seven instances. We see that B&C yields significantly shorter solution times in all these instances, which grows as large as 99.7% for *bhvdc*. Moreover, we provide a comparison of termination gaps with CPLEX and our B&C algorithm for the remaining 11 instances in Figure 3(b).

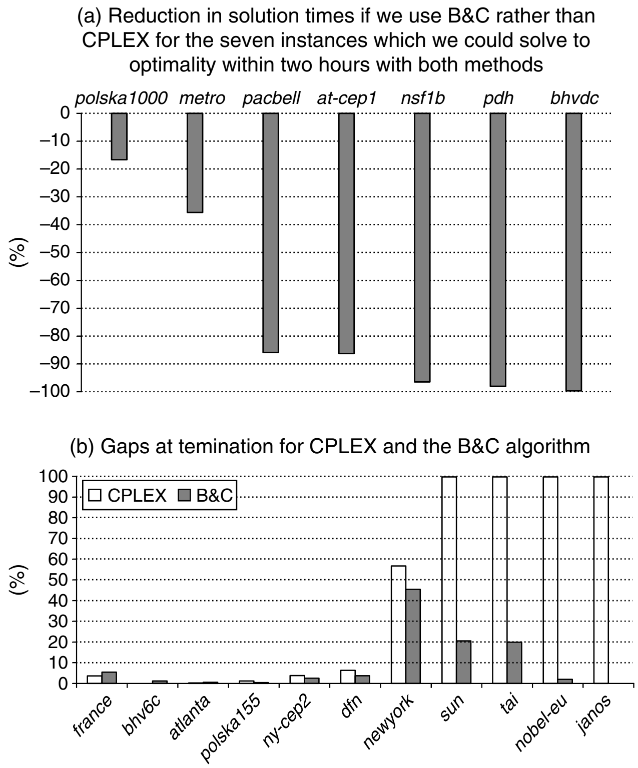

Figure 3 **Comparison of Solution Times and Termination Gaps for the Single-Facility Case**

Even though CPLEX gives better upper bounds than B&C in *dfn, ny-cep2*, and *atlanta*, the gaps at termination are better for the B&C algorithm in the first two of these instances. On the other hand, B&C is clearly superior for *newyork, tai, janos, nobel-eu,* and *sun*. The most important observation here is the significant degradation in the performance of CPLEX relative to the B&C algorithm as the network size increases. The instances *tai, janos, nobel-eu,* and *sun* are very good examples of this behavior. Except *tai*, all of the nodes are demand nodes in these instances, and we observe that among such cases, only in *dfn* and *atlanta* has CPLEX performed slightly better than B&C. The upper bound of CPLEX is just 0.07% and 0.2% tighter than the one of B&C in *dfn* and *atlanta*, respectively. On the other hand, the upper bounds we obtain with B&C are 100% better than the bounds with CPLEX in *tai, janos, nobel-eu,* and *sun*. Finally, a comparison of the gaps at termination shows that the B&C algorithm is clearly superior in 8 of the 11 instances with much lower gaps for *tai*, *nobel-eu*, and *sun*, in addition to the zero gap for *janos*. In sum, B&C is superior in terms of solution times or termination gaps in 15 of the 18 instances.

We have also investigated the individual and joint influence of the two types of cuts on the root relaxation solution qualities and the total solution times. We consider the four cases F, $F\&D$, $F\&R$, and *all*, where each capital letter shows which of the feasibility cuts (F), demand cutset inequalities (D), and residual capacity inequalities (R) are used throughout the B&C algorithm.

We have considered six instances that were solved to optimality in relatively shorter times. In Figure 4, we display the percentage of improvement for solution times and the relative change for root gaps when we use each setting rather than F, e.g., the change in solution time for $F\&D$ is $((\text{time}(F\&D) - \text{time}(F))/\text{time}(F)) * 100$ and the change in root gap is $\text{gap}(F\&D) - \text{gap}(F)$. Figure 4 shows that the impact of demand cutset inequalities both on root gaps and solution times is significant. The residual capacity inequalities also yield reasonable improvements in root gaps. Although adding residual capacity and demand cutset inequalities together does not improve the root gaps, it improves the solution times. Average improvements in root gaps and solution times are 86% and 84.72%, respectively, for the setting *all*.

Next, we compare the design cost for the hose model with the BS model for $\Gamma = \lceil 0.1|Q|\rceil$, $\Gamma = \lceil 0.15|Q|\rceil$, and $\Gamma = \lceil 0.25|Q|\rceil$. We show the percentage increase in design costs for the BS model, which is measured as $((\text{cost}_{\text{BS}} - \text{cost}_{\text{hose}})/\text{cost}_{\text{hose}}) \times 100$, in Figure 5. We see that the BS model leads to more-costly designs with respect to the hose model as Γ, i.e., the level of conservatism, increases. Average differences are 0.36%, 4.83%, and 10.01%, respectively.

Finally, we consider the set of instances for which we could solve both the deterministic and robust problems in less than two hours, and we show the

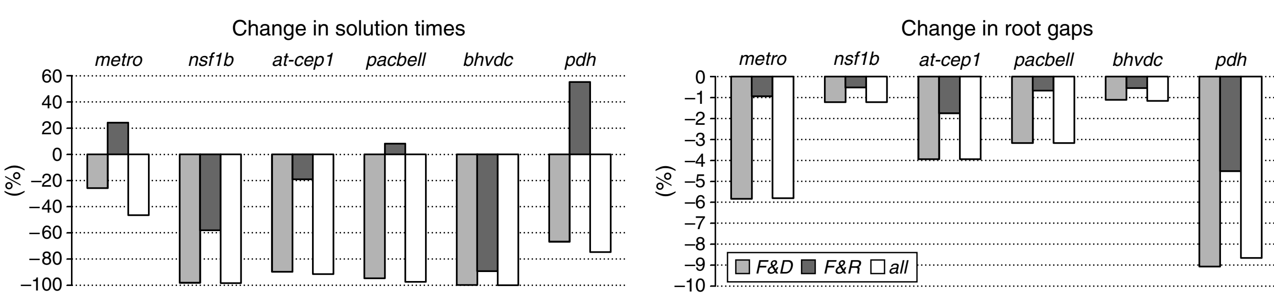

Figure 4 **Impact of Different Cuts**

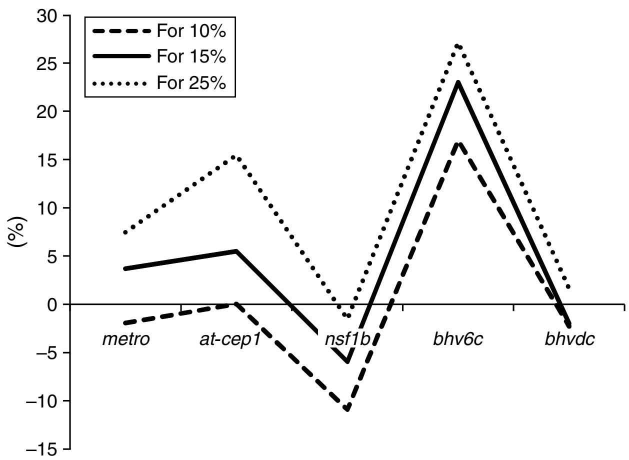

Figure 5 **Increase in Cost if We Use the BS Model with Different Γ Rather than the Hose Model**

change in the optimal capacity installation costs in Figure 6(a). The average increase in the total reservation cost as we shift to the robust counterpart from the deterministic NLP is 17.62%. Although we have to pay for the additional flexibility that the hose model provides, we avoid overconservative designs by exploiting the hose model. Suppose that we have the bandwidth capacities for all nodes and we look for a design that can support the worst case that can happen based on the given information. Clearly, the safest approach would be to fix the demand to its worst-case value as $d_q=\min\{b_{s(q)},b_{t(q)}\}$ for each $q\in Q$ and

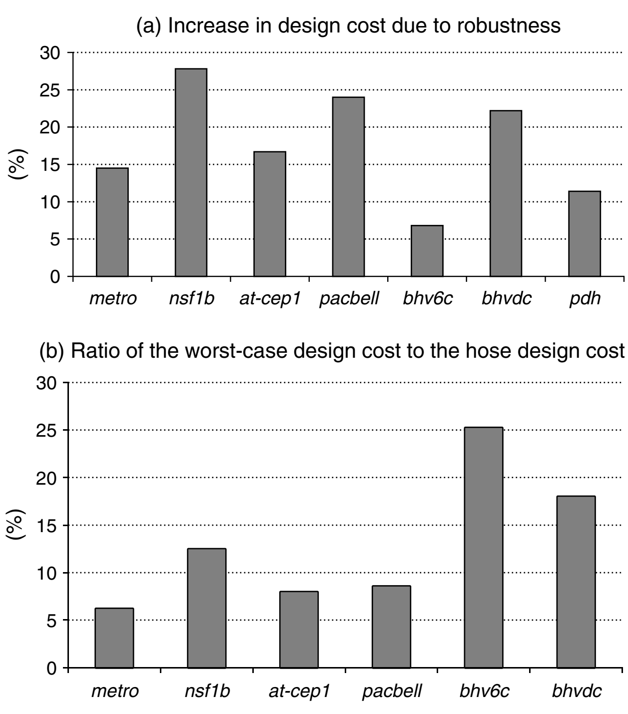

Figure 6 **Hose Design Cost vs. the Deterministic and the Worst-Case Design Costs**

then solve (1)–(5). For the above instances, the cost of this worst-case deterministic model is 6 to 25 times larger than the cost with the hose model. We show the magnitudes of increase in terms of the ratio of the worst-case cost to the hose design cost in Figure 6(b).

5.2. Two-Facility NLP_{hose}

In the two-facility case, we consider two types of facilities, namely, low-capacity (*LCF*) and high-capacity (*HCF*) facilities with transmission capacities of C^1 and C^2 units, respectively. Naturally, the cost of installing each facility is different and economies of scale prevail; i.e., the cost of $\lceil C^2/C^1\rceil$ *LCF*s is more than the cost of one *HCF*. For $S\subset V$, the demand cutset inequalities (23) reduce to the following inequalities:

- *The LCF case*, i.e., $l^*=1$, where the resulting inequalities can be as follows:

$$\begin{cases} R^1(S)Y^1(S)+\left(R^1(S)\left\lfloor\dfrac{C^2}{C^1}\right\rfloor+\min\{g(2,1),R^1(S)\}\right) & \\ \quad\cdot Y^2(S)\ge R^1(S)\left\lceil\dfrac{B(S)}{C^1}\right\rceil & \text{if } C^1,C^2<B(S),\\ Y^1(S)+Y^2(S)\ge 1 & \text{if } C^1,C^2\ge B(S),\\ Y^1(S)+\left\lceil\dfrac{B(S)}{C^1}\right\rceil Y^2(S)\ge\left\lceil\dfrac{B(S)}{C^1}\right\rceil & \\ & \text{if } C^1<B(S) \text{ and } C^2\ge B(S).\end{cases}$$

- *The HCF case*, i.e., $l^*=2$ where we can have

$$\begin{cases} \min\{C^1,R^2(S)\}Y^1(S)+R^2(S)Y^2(S) & \\ \quad\ge R^2(S)\left\lceil\dfrac{B(S)}{C^2}\right\rceil & \text{if } C^1,C^2<B(S),\\ Y^1(S)+Y^2(S)\ge 1 & \text{if } C^1,C^2\ge B(S),\\ C^1Y^1(S)+\lceil B(S)\rceil Y^2(S)\ge\lceil B(S)\rceil & \\ & \text{if } C^1<B(S) \text{ and}\\ & \quad C^2\ge B(S).\end{cases}$$

The two types of residual capacity inequalities (24) for each edge $e\in E$ and set $S\subset V$ are

$$\begin{cases} r^1(S)y_e^1+\left(r^1(S)\left\lfloor\dfrac{C^2}{C^1}\right\rfloor+\min\{g(2,1),r^1(S)\}\right)y_e^2 & \\ \quad-\displaystyle\sum_{i\in S\cap W}b_i\lambda_i^e\ge r^1(S)\left\lceil\dfrac{b(S)}{C^1}\right\rceil-b(S) & \text{for } l^*=1,\\ \min\{C^1,r^2(S)\}y_e^1+r^2(S)y_e^2-\displaystyle\sum_{i\in S\cap W}b_i\lambda_i^e & \\ \quad\ge r^2(S)\left\lceil\dfrac{b(S)}{C^2}\right\rceil-b(S) & \text{for } l^*=2.\end{cases}$$

The number of residual capacity and demand cutset inequalities are doubled as we move from the single-facility case to the two-facility case. As a result, the

LP models we solve at each iteration of the B&C algorithm can rapidly get large. Therefore, we have tried the following five different schemes for adding violated cuts:

- *HA*: add only HCF-type inequalities in all nodes of the B&C tree;
- *HR*: add only HCF-type inequalities only at the root node;
- *GHA*: add HCF-type inequalities gradually—i.e., add a violated HCF residual capacity inequality only if no HCF demand cutset inequality is violated, in all nodes of the B&C tree;
- *GHR*: gradually add HCF-type inequalities—i.e., add a violated HCF residual capacity inequality only if no HCF demand cutset inequality is violated, only at the root node of the B&C tree; and
- *GAR*: gradually add all valid inequalities—i.e., add violated LCF and HCF residual inequalities only if no LCF or HCF demand cutset inequality is violated, at the root node.

We compared the performances of the five settings in terms of the gaps at termination as shown in Figure 7. The instances for which the B&C algorithm could not find a feasible solution within the two-hour time limit are assigned a 105% gap. Furthermore, we leave the *bhv6c* instance out of this analysis because all schemes stopped with the same gap. Consequently, we see that the average gaps at termination for these 11 instances are 32.6%, 38.5%, 31.1%, 31.2%, and 56.9% for *HA, HR, GHA, GHR,* and *GAR*, respectively. The average number of nodes in the B&C tree for these five settings are 13,968, 11,769, 7,869, 8,903, and 8,629. An important point to note here is that the number of nodes is one for those instances terminated with no feasible solution. Thus, even though the highest number of such cases are observed for *GAR*, the size of the B&C tree is smaller for *GHA* on average. In what follows, we provide the results with *GHA*.

Initially, we consider the six instances, which we could solve to optimality both with CPLEX and the B&C algorithm. Figure 8(a) shows the change in solution times defined as $((\text{time}(B\&C) - \text{time}(CPLEX))/\text{time}(CPLEX)) * 100$. We see that B&C is faster than CPLEX for all of these instances. CPLEX was faster in only *pacbell*, which we do not show in Figure 8(a) in order not to blur the figure. Although the percentage change seems quite significant for this instance, the difference is actually in seconds, and we could solve it in less than one minute in both cases.

Next, we provide the test results for the remaining 11 instances in Figure 8(b). The termination gap for the instances, for which we could not solve the LP relaxation in two hours, is taken to be 105%.

We see that our B&C algorithm is superior to CPLEX, especially for the large instances where all nodes are demand nodes just like the single-facility case. This is quite obvious especially for *tai*, *nobel-eu*, *pioro*, and *cost266* because the MIP solver could not find even a feasible solution in two hours, whereas the B&C algorithm successfully produced some upper bounds. Specifically, the upper bounds for *nobel-eu* and *pioro* are quite promising. Moreover, the $NLP_{\text{GD}}^{\text{hose}}$ problem could not be solved for *newyork* because of insufficient memory. In two cases, i.e., *norway* and *gui*, we could not find any upper bound with either of the methods. On the other hand, the B&C algorithm is better in six of the remaining nine instances with much lower gaps for *dfn*, *tai*, *nobel-eu*, *pioro*, and *cost266*.

A final analysis in Figure 9 is about the price of robustness measured in terms of the percent change in the final design cost for the two-facility case. The average increase in the optimal reservation costs of

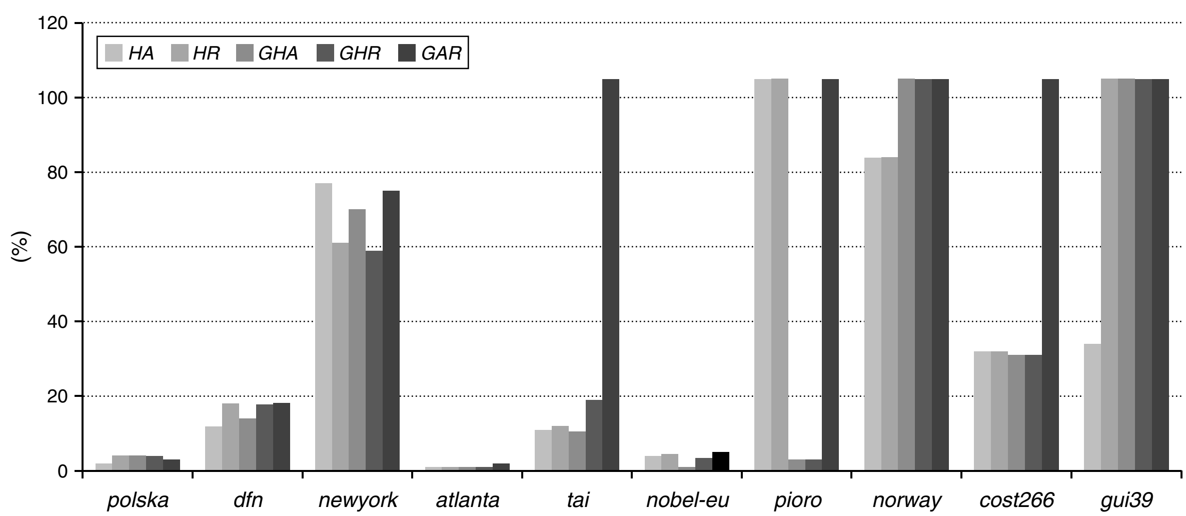

Figure 7 Percent Gaps at Termination for Each Scheme

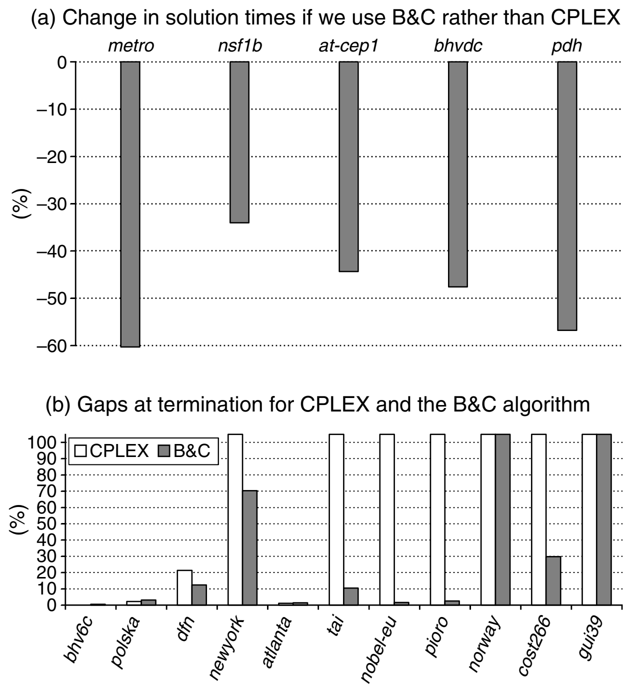

Figure 8 **Comparison of Solution Times and Termination Gaps for the Two-Facility Case**

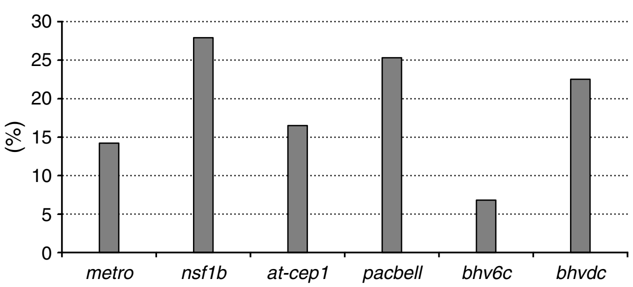

Figure 9 **Increase in Design Cost as a Result of Robustness**

the six instances for which gaps could be calculated is 18.86%.

5.3. Parametric Hose Case

In this section, we consider the *metro* instance and analyze the sensitivity of the robust design to the choice of bandwidth capacities. First, we generate 20 demand matrices $\tilde{d}^1, \ldots, \tilde{d}^{20}$, where the demand $\tilde{d}_q^j$ for each $q \in Q$ is normally distributed with mean $\bar{d}_q$ and standard deviation $K\bar{d}_q$ for $K \in (0, 1]$. Next, for $\tau \in \Re_{++}$, we let $b_i = \tau \sum_{q \in Q:\, s(q)=i \text{ or } t(q)=i} \bar{d}_q$ for all $i \in V$ and solve the corresponding NLP_{GD} to get the optimal capacity configuration $y(\tau)$. Then, for each $j = 1, \ldots, 20$, we determine the maximum total flow $F^j(\tau)$ we can route given demand matrix $\tilde{d}^j$ and link capacities $y(\tau)$ by solving a linear programming problem. We calculate the fraction of demand routed as $F^j(\tau)/\sum_{q \in Q} \tilde{d}_q^j$ and take the average over 20 demand

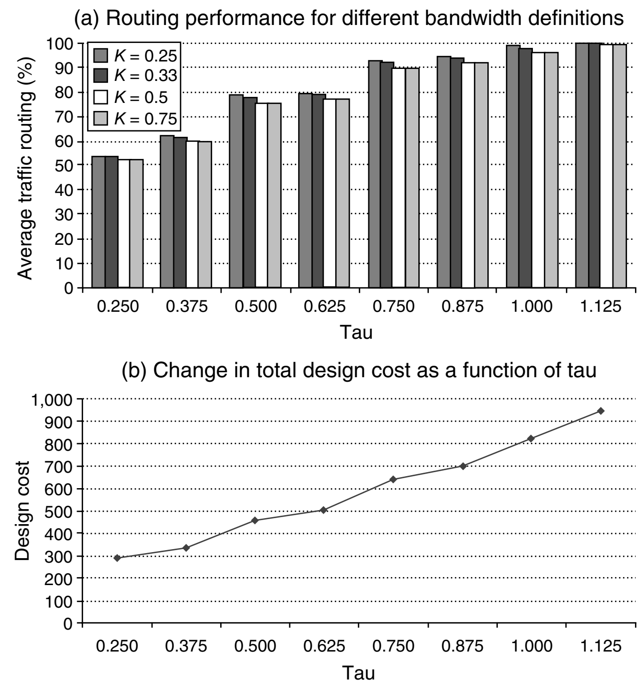

Figure 10 **Implications of the Bandwidth Definition on Routing Performance and Design Cost**

matrices to evaluate the performance of the optimal hose design $y(\tau)$ for a given τ.

We have performed several tests with $K \in \{0.25, 0.33, 0.5, 0.75\}$ and eight different values of $\tau \in [0.25, 1.125]$. Figure 10(a) shows the average percentage of traffic we could route under different hose definitions and K values.

As expected, Figure 10(a) shows that independent of what K is, the traffic routing rate improves as we consider a larger hose polyhedron, i.e., as τ grows. This is natural because a larger hose polyhedron implies a more conservative design. On the other hand, for a given τ, the demand satisfaction rate is negatively affected by demand deviations. However, higher protection comes at a cost, and Figure 10(b) shows how the total cost changes with τ.

The proper choice of τ is related to the accuracy of the demand information as well as the trade-off between the design cost and the service level. We study the hose polyhedron for $\tau = 1$ for our tests in §§5.1 and 5.2. The results above show that the average routing rates of the corresponding robust design for $K \in \{0.25, 0.33, 0.5, 0.75\}$ are 98.62%, 97.92%, 96.10%, and 92.81%, respectively.

6. Conclusion

In this paper we studied the network loading problem where the pairwise traffic demands are not assumed to be known in advance. We used a polyhedral definition of traffic demands and sought to design a network that is capable of supporting infinitely many nonsimultaneous demand realizations. Based on a compact formulation and a decomposition property, we gave a detailed polyhedral analysis for a specific demand uncertainty description, the hose model. The polyhedral analysis formed the basis of an efficient B&C algorithm. Our computational results reveal that projecting out the flow variables and the use of a B&C algorithm is quite effective for both single- and two-facility problem types. An important question is whether similar developments can be expected for uncertainty polyhedron descriptions other than the hose model. We will answer this question in subsequent papers.

Acknowledgments

Research partially supported by TUBITAK-CNRS (TUBITAK project 105M322 and CNRS project BOSPHORE 10843 TD). The authors thank A. Ridha Mahjoub and Hervé Kerivin for helpful discussions and three anonymous referees for detailed suggestions.

References

Altın, A., P. Belotti, M. Ç. Pınar. 2010. OSPF routing with optimal oblivious performance ratio under polyhedral demand uncertainty. *Optim. Engrg.* **11**(3) 395–422.

Altın, A., E. Amaldi, P. Belotti, M.Ç. Pınar. 2007. Provisioning virtual private networks under traffic uncertainty. *Networks* **49**(1) 100–115.

Atamtürk, A. 2002. On capacitated network design cut-set polyhedra. *Math. Programming B* **92**(3) 425–437.

Atamtürk, A. 2006. Strong reformulations of robust mixed 0–1 programming. *Math. Programming* **108**(2–3) 235–250.

Atamtürk, A., O. Günlük. 2007. Network design arc set with variable upper bounds. *Networks* **50**(1) 17–28.

Atamtürk, A., D. Rajan. 2002. On splittable and unsplittable capacitated network design arc-set polyhedra. *Math. Programming* **92**(2) 315–333.

Atamtürk, A., M. Zhang. 2007. Two-stage robust network flow and design under demand uncertainty. *Oper. Res.* **55**(4) 662–673.

Avella, P., S. Mattia, A. Sassano. 2007. Metric inequalities and the network loading problem. *Discrete Optim.* **4**(1) 103–114.

Belotti, P., M. Ç. Pınar. 2008. Optimal oblivious routing under linear and ellipsoidal uncertainty. *Optim. Engrg.* **9**(3) 257–271.

Ben-Ameur, W., H. Kerivin. 2005. Routing of uncertain traffic demands. *Optim. Engrg.* **6**(3) 283–313.

Ben-Tal, A., A. Nemirovski. 1998. Robust convex optimization. *Math. Oper. Res.* **23**(4) 769–805.

Ben-Tal, A., A. Nemirovski. 1999. Robust solutions of uncertain linear programs. *Oper. Res. Lett.* **25**(1) 1–13.

Ben-Tal, A., A. Nemirovski. 2008. Selected topics in robust convex optimization. *Math. Programming* **112**(1) 125–158.

Ben-Tal, A., A. Goryashko, E. Guslitzer, A. Nemirovski. 2004. Adjustable robust solutions of uncertain linear programs. *Math. Programming* **99**(2) 351–376.

Berger, D., B. Gendron, J.-Y. Potvin, S. Raghavan, P. Soriano. 2000. Tabu search for a network loading problem with multiple facilities. *J. Heuristics* **6**(2) 253–267.

Bertsimas, D., M. Sim. 2003. Robust discrete optimization and network flows. *Math. Programming* **98**(1–3) 49–71.

Bertsimas, D., M. Sim. 2004. The price of robustness. *Oper. Res.* **52**(1) 35–53.

Bienstock, D., O. Günlük. 1996. Capacitated network design—Polyhedral structure and computation. *INFORMS J. Comput.* **8**(3) 243–259.

Bienstock, D., S. Chopra, O. Günlük, C.-Y. Tsai. 1998. Minimum cost capacity installation for multicommodity network flows. *Math. Programming* **81**(2) 177–199.

Brockmüller, B., O. Günlük, L. A. Wolsey. 2004. Designing private line networks—Polyhedral analysis and computation. *Trans. Oper. Res.* **16**(1, 2) 7–24.

Cornuéjols, G. 2008. Valid inequalities for mixed integer linear programs. *Math. Programming* **112**(1) 3–44.

Duffield, N., P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, J. E. van der Merive. 1999. A flexible model for resource management in virtual private networks. *Proc. ACM SIGCOMM,* ACM, New York, 95–108.

Fingerhut, J. A., S. Suri, J. S. Turner. 1997. Designing least-cost nonblocking broadband networks. *J. Algorithms* **24**(2) 287–309.

Goyal, N., N. Olver, F. B. Shepherd. 2008. The VPN conjecture is true. *Proc. ACM STOC,* ACM, New York, 443–450.

Grandoni, F., V. Kaibel, G. Oriolo, M. Skutella. 2008. A short proof of the VPN tree routing conjecture on ring networks. *Oper. Res. Lett.* **36**(3) 361–365.

Günlük, O. 1999. A branch-and-cut algorithm for capacitated network design problems. *Math. Programming* **86**(1) 17–39.

Gupta, A., A. Kumar, T. Roughgarden. 2003. Simpler and better approximation algorithms for network design. *Proc. ACM STOC,* ACM, New York, 365–372.

Gupta, A., J. Kleinberg, A. Kumar, R. Rastogi, B. Yener. 2001. Provisioning a virtual private network: A network design problem for multicommodity flow. *Proc. ACM STOC, Hersonissos, Greece*, ACM, New York, 389–398.

Hurkens, C. A. J., J. C. M. Keijsper, L. Stougie. 2007. Virtual private network design: A proof of the tree routing conjecture on ring networks. *SIDMA* **21**(2) 482–503.

Italiano, G. F., R. Rastogi, B. Yener. 2002. Restoration algorithms for virtual private networks in the hose model. *INFOCOM Proc 21st Annual Joint Conf. IEEE Comput. Comm. Soc.*, Vol. 1. IEEE, Washington, DC, 131–139.

Karaşan, O., M. Ç. Pınar, H. Yaman. 2005. Robust DWDM routing and provisioning under polyhedral demand uncertainty. Technical report, Bilkent University, Bilkent, Ankara, Turkey.

Kumar, A., R. Rastogi, A. Silberschatz, B. Yener. 2001. Algorithms for provisioning virtual private networks in the hose model. *ACM SIGCOMM Comput. Comm. Rev.* **31**(4) 135–146.

Labbé, M., H. Yaman. 2004. Projecting the flow variables for hub location problems. *Networks* **44**(2) 84–93.

Magnanti, T. L., P. Mirchandani. 1993. Shortest paths, single origin-destination network design, and associated polyhedra. *Networks* **23**(2) 103–121.

Magnanti, T. L., P. Mirchandani, R. Vachani. 1993. The convex hull of two core capacitated network design problems. *Math. Programming* **60**(1–3) 233–250.

Magnanti, T. L., P. Mirchandani, R. Vachani. 1995. Modeling and solving the two-facility capacitated network loading problem. *Oper. Res.* **43**(1) 142–157.

Mirchandani, P. 2000. Projections of the capacitated network loading problem. *Eur. J. Oper. Res.* **122**(3) 534–560.

Mudchanatongsuk, S., F. Ordoñez, J. Liu. 2008. Robust solutions for network design under transportation cost and demand uncertainty. *J. Oper. Res. Soc.* **59**(5) 652–662.

Nemhauser, G. L., M. W. P. Savelsbergh, G. C. Sigismondi. 1994. MINTO, a mixed INTeger optimizer. *Oper. Res. Lett.* **15**(1) 47–58.

Onaga, K., O. Kakusho. 1971. On feasibility conditions of multicommodity flows in networks. *IEEE Trans. Circuit Theory* **18**(4) 425–429.

Ordoñez, F., J. Zhao. 2007. Robust capacity expansion of network flows. *Networks* **50**(2) 136–145.

Raack, C., A. M. C. A. Koster, S. Orlowski, R. Wessäly. 2010. On cut-based inequalities for capacitated network design polyhedra. *Networks*, ePub ahead of print June 30, http://onlinelibrary.wiley.com/doi/10.1002/net.20395/abstract.

Rardin, R. L., L. A. Wolsey. 1993. Valid inequalities and projecting the multicommodity extended formulation for uncapacitated fixed charge network flow problems. *Eur. J. Oper. Res.* **71**(1) 95–109.

Soyster, A. L. 1973. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Oper. Res.* **21**(5) 1154–1157.

Swamy, C., A. Kumar. 2002. Primal-dual algorithms for connected facility location problems. *Proc. Internat. Workshop Approximation Algorithms Combinatorial Optim. (APPROX). Lecture Notes in Computer Science Series*, Vol. 2462. Springer, Berlin, 256–270.

van Hoesel, S. P. M., A. M. C. A. Koster, R. L. M. J. van de Leensel, M. W. P. Savelsbergh. 2002. Polyhedral results for the edge capacity polytope. *Math. Programming* **92**(2) 335–358.

Wolsey, L. A. 1998. *Integer Programming*. Wiley-Interscience, New York.

Yaman, H. 2007. The integer knapsack cover polyhedron. *SIAM J. Discrete Math.* **21**(3) 551–572.

Yaman, H., O. E. Karaşan, M. Ç. Pınar. 2007. Restricted robust uniform matroid maximization under interval uncertainty. *Math. Programming* **110**(2) 431–441.

Zuse-Institute Berlin. SNDlib. Accessed March 8, 2010, http://sndlib.zib.de/home.action.