We investigate a network design problem under traffic uncertainty that arises when provisioning Virtual Private Networks (VPNs): given a set of terminals that must communicate with one another, and a set of possible traffic matrices, sufficient capacity has to be reserved on the links of the large underlying public network to support all possible traffic matrices while minimizing the total reservation cost. The problem admits several versions depending on the desired topology of the reserved links, and the nature of the traffic data uncertainty. We present compact linear mixed-integer programming formulations for the problem with the classical hose traffic model and for a less conservative robust variant relying on the traffic statistics that are often available. These flow-based formulations allow us to solve optimally medium-to-large instances with commercial MIP solvers. We also propose a combined branch-and-price and cutting-plane algorithm to tackle larger instances. Computational results obtained for several classes of instances are reported and discussed. © 2006 Wiley Periodicals, Inc.

Keywords: virtual private networks; network design; traffic uncertainty; robust optimization; mixed-integer linear programs; branch and price; cutting planes

1. INTRODUCTION

Virtual Private Networks (VPNs) are network services built over an existing public network to provide quality of service, flexibility, and security while saving costs through link multiplexing. They use encryption and tunneling to link branch offices to an enterprise network, or to extend organizations’ existing computing infrastructure to include partners, suppliers, and customers [14]. The use of public networks reduces operational costs due to economies of scale while ensuring a wider area accessibility and communication security through encryption. Flexibility and cost effectiveness have turned VPN solutions into a billion dollar industry.

In this article we address a network design problem that arises when provisioning VPNs. Given a set of terminals that must communicate with one another and a set of possible traffic patterns, sufficient capacity has to be reserved on the links of the large underlying public network to support all possible traffic patterns while minimizing cost. The solution of this resource management problem clearly depends on the possible traffic patterns and the constraints on the topology of the reserved links.

In traditional network design problems, it is assumed that a traffic matrix, that is, a set of demands for all origin–destination pairs, can be reliably estimated. Because a VPN service customer has in general no precise information about the expected traffic between the terminals to be connected, a collection of possible traffic matrices have to be simultaneously considered. In [7] a flexible model referred to as the hose model was proposed to specify the bandwidth requirements of a single VPN. In this hose model, the set of valid traffic matrices is defined by imposing, for each terminal \( t \), an upper bound on the total outgoing traffic from \( t \) (toward the other terminals) and an upper bound on the total entering traffic in \( t \) (from the other terminals).

The underlying network over which a VPN has to be provisioned is represented by an undirected graph \( G = (V, E) \). Each edge \( e \) in \( E \), also denoted by \( \{i, j\} \) to emphasize the two end-nodes \( i, j \) in \( V \), has a per-unit nonnegative reservation cost \( c_{ij} \). Let \( Q \subseteq V \) denote the set of terminals
that need to communicate with one another. For each ordered pair of terminals \((s, t)\), with \(s, t \in Q\), the nonnegative \(d_{st}\) represents the amount of traffic that has to be routed from \(s\) to \(t\). We assume that \(d_{st} = 0\) for every terminal \(s\) and denote by \(S\) the set of all ordered pairs of distinct terminals, namely \(S = \{(s, t) \in Q \times Q : s \neq t\}\). In the hose model, two nonnegative bounds \(b_s^+\) and \(b_s^-\) are specified for each terminal \(s \in Q\), and the traffic demands \(d_{st}\) are assumed to be nonnegative values that belong to the uncertainty set \(U_{SymG}\) defined by the following inequalities:

\[
\sum_{t \in Q} d_{st} \leq b_s^+ \quad \sum_{t \in Q} d_{ts} \leq b_s^- \quad \forall s \in Q \tag{1}
\]

\[
d_{st} \geq 0 \quad \forall (s, t) \in S. \tag{2}
\]

Throughout this article we make the realistic assumption that traffic is unsplitable, that is, for each demand pair \((s, t) \in S\) the traffic demand \(d_{st}\) is routed along a single path from \(s\) to \(t\). (Although the case where each traffic demand can be arbitrarily split and routed along several paths is considered among others in [8], multipath routing is currently hardly implementable in VPNs because packets related to a single flow may arrive out of sequence and thus cause critical problems at the Transfer Control Protocol, TCP, level.) Provisioning a VPN then consists of reserving capacity on the edges and selecting a single routing path for each demand pair to support all valid traffic matrices while minimizing the total reservation cost.

In the VPN provisioning problems considered in the present article, no assumption is made on the integrality of demand values even if the bounds \(b_s^+\) and \(b_s^-\) are integral. In other words, the resulting design should support integral as well as nonintegral traffic demand matrices from the uncertainty polyhedron \((1)-(2)\). Therefore, the network design problem of the present article, which has a multicommodity flow structure, is both continuous and discrete in nature because fractional amounts of capacity can be reserved on the links while demands must be routed along single paths.

If no particular constraint is imposed on the topology of the reserved network (union of all edges with a strictly positive capacity reservation), the VPN provisioning problem with the hose traffic model is called Asym-G. The version in which the reserved network is required to be a tree is referred to as Asym-T.

The symmetric versions, where a threshold \(b_s\) is given for the sum of incoming and outgoing traffic for all terminals \(s \in Q\), are referred to as Sym-G and Sym-T, respectively. These assumptions imply that the traffic demands are elements of the set \(U_{SymG}\) defined as follows:

\[
\sum_{t \in Q} d_{st} + \sum_{t \in Q} d_{ts} \leq b_s \quad \forall s \in Q \tag{3}
\]

\[
d_{st} \geq 0 \quad \forall (s, t) \in S, \tag{4}
\]

In the present article we develop new, compact, linear mixed-integer programming (MIP) formulations for the Asym-G and Sym-G variants of the VPN provisioning problem under the hose model of uncertainty (cf. Proposition 1). Although the computational complexity status of Asym-G is still open, medium-to-large instances of our models turn out to be solvable by the off-the-shelf mixed-integer optimizer Cplex 8.1 within short computing time.

Traffic uncertainty is a crucial feature in VPN provisioning. Because the hose model makes very weak assumptions (only imposes upper bounds on the inflow and outflow of each terminal), it may lead to excessive capacity reservation. This traffic model is a special case of the so-called polyhedral model in which the set of valid matrices is defined by an arbitrary polyhedron [4]. In principle, the polyhedral model allows us to focus on smaller subsets of traffic matrices than the hose model, but it is unclear how to actually define realistic polyhedra that would lead to less conservative VPN reservations.

From the application point of view, a service provider simultaneously provisions a number of VPNs for different customers over a certain time period and the service level agreements are renegotiated on a regular basis. Although the hose traffic model is adequate for new VPNs in the absence of precise traffic predictions, it is clearly overly conservative for VPNs that are already provisioned. Because service providers collect detailed terminal-to-terminal traffic statistics for each VPN, a less conservative traffic uncertainty model that exploits the available statistics is needed. Therefore, we investigate a less conservative robust variant of the problem, which exploits the traffic statistics that are available for existing VPNs. In particular, a corresponding compact linear MIP formulation is presented (cf. Proposition 2) and the robust VPN provisioning problem is shown to be NP-hard (cf. Proposition 3).

In practice, service providers face an incremental reservation problem: while provisioning a given set of VPNs,
they receive a request for a new VPN or for changes to the set of terminals of an existing one. Because the additional capacity requirements for a single VPN are in general very small compared with the overall network capacity, it is reasonable from an application point of view to focus on the uncapacitated versions of the VPN provisioning problem (A. Capone, personal communication, 2003). Due to the absence of capacity constraints and the fractional capacity that can be reserved on each link of the backbone network, the incremental problem can then be decomposed into a sequence of single VPN provisioning problems with appropriate traffic uncertainty models.

The rest of the article is organized as follows. In Section 2.1 we present compact flow-based linear MIP formulations for Asym-\(G\) and Sym-\(G\) under the hose uncertainty model. In Section 2.2 we investigate the less conservative robust variant of the VPN provisioning problem. For both hose uncertainty and robust uncertainty models we also discuss cutting planes for obtaining lower bounds. Finally, to close the discussion of the models, in Section 2.3 we give the path-based formulations more suitable for the exact solution method detailed in Section 3 where we describe a combined branch-and-price and cutting-plane algorithm to tackle larger instances of the three problem versions. Computational results obtained for several classes of instances are reported and discussed in Section 4. Section 5 contains some concluding remarks.

2. VPN PROVISIONING MODELS

2.1. VPN Provisioning Problem with the Hose Traffic Model

For new VPNs or existing ones in which at least a new terminal is added, demand statistics are not available or not reliable due to the lack of information about the traffic generated by the new terminals. The hose uncertainty model is then well suited for the demands involving at least a new terminal.

Consider the general variant Asym-\(G\) with asymmetric traffic matrices. The network cost, which depends on the capacity to be reserved so as to route all demands, has to be minimized. Because traffic is assumed to be unsplitable,
each demand has to be routed along a single path from origin to destination.

2.1.1. From Constant Traffic Matrices to the Hose Model. Let us first consider the capacity reservation problem aiming at minimizing cost while supporting a single traffic matrix \( D = (d_{st})_{s,t \in \Omega} \) and note that a straightforward minimum cost flow formulation can be solved in polynomial time. We define two classes of variables. For each edge \( \{i, j\} \), the continuous variable \( x_{ij} \) represents the capacity to be reserved on \( \{i, j\} \). Let \( A = \{\{i, j\} : \{i, j\} \in E\} \) denote the underlying set of oriented arcs. For each oriented arc \( (i, j) \in A \) and oriented pair of terminals \( (s, t) \in S \), the binary variable \( y_{ij}^{st} \) corresponds to the flow on the oriented arc \( (i, j) \) for the terminal pair \( (s, t) \)

\[
y_{ij}^{st} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is used to route demand } d_{st} \\ 0 & \text{otherwise} \end{cases}
\]

This leads to the flow formulation:

\[
\begin{align*}
\min & \sum_{\{i, j\} \in E} c_{ij} x_{ij} \\
\text{s.t.} & \sum_{\{i, j\} \in E} (y_{ij}^{st} - y_{ji}^{st}) \\
& = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V, \forall (s, t) \in S \quad (6) \\
& \sum_{\{i, j\} \in S} d_{st} (y_{ij}^{st} + y_{ji}^{st}) \leq x_{ij} \quad \forall \{i, j\} \in E \quad (7) \\
y_{ij}^{st} \in \{0, 1\} \quad \forall (i, j) \in A, \forall (s, t) \in S \quad (8) \\
x_{ij} \geq 0 \quad \forall \{i, j\} \in E, \quad (9)
\end{align*}
\]

where the objective function corresponds to the VPN reservation cost. Constraints (6) ensure demand satisfaction by imposing a unit flow between each oriented pair of terminals. For each edge \( \{i, j\} \) the corresponding constraint (7) guarantees that the capacity \( x_{ij} \) reserved on \( \{i, j\} \) is sufficient to carry the total traffic flowing through it.

If all the demands \( d_{st} \) are precisely known, (5)–(9) is a linear mixed-integer program. Due to the nonnegativity of the costs \( c_{ij} \) and of all variables, constraints (7) are satisfied as equalities in any optimal solution to (5)–(9). Therefore, the continuous variables \( x_{ij} \) can be omitted and we have the equivalent formulation:

\[
\begin{align*}
\min & \sum_{\{i, j\} \in E} c_{ij} \sum_{\{i, j\} \in S} d_{st} (y_{ij}^{st} + y_{ji}^{st}) \\
\text{s.t.} & \sum_{\{i, j\} \in E} (y_{ij}^{st} - y_{ji}^{st}) \\
& = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V, \forall (s, t) \in S \quad (11) \\
y_{ij}^{st} \in \{0, 1\} \quad \forall (i, j) \in A, \forall (s, t) \in S \quad (12)
\end{align*}
\]

with only binary variables. Because there are no capacity constraints and fractional capacities can be reserved on the edges, the constraint matrix is totally unimodular and the problem is solvable in polynomial time [15].

If the traffic matrix \( D \) is subject to the hose uncertainty model, a first formulation is obtained by requiring that capacity constraints (7) hold for all traffic matrices satisfying constraints (1)–(2), or respectively (3)–(4). Thus, we obtain the following semi-infinite MIP formulation

\[
\begin{align*}
\min & \sum_{\{i, j\} \in E} c_{ij} x_{ij} \\
\text{s.t.} & \sum_{\{i, j\} \in E} (y_{ij}^{st} - y_{ji}^{st}) \\
& = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V, \forall (s, t) \in S \quad (13) \\
& \sum_{\{i, j\} \in S} d_{st} (y_{ij}^{st} + y_{ji}^{st}) \\
& \leq x_{ij} \quad \forall D \in U_{\text{AsymG}}(U_{\text{SymG}}), \forall \{i, j\} \in E \quad (14) \\
y_{ij}^{st} \in \{0, 1\} \quad \forall (i, j) \in A, \forall (s, t) \in S \quad (15) \\
x_{ij} \geq 0 \quad \forall \{i, j\} \in E. \quad (16)
\end{align*}
\]

Two previous works [4, 8] deal with the semi-infinite nature of the problem by dynamically generating a set of traffic matrices corresponding to vertices of the demand polyhedron. Solving the restricted problem yields values for the flow variables \( \hat{y}_{ij}^{st} \) and the capacity variables \( \hat{x}_{ij} \). Because the resulting routing vector \( \tilde{y} \) may be infeasible for the hose polyhedron defined by constraints (1)–(2), or by (3)–(4), a new vertex (traffic matrix) is sought by solving another linear program, where the \( \tilde{y}s \) and \( \hat{x}s \) are considered as coefficients and the edge overload \( \sum_{\{i, j\} \in S} d_{st} (\hat{y}_{ij}^{st} + \hat{y}_{ji}^{st}) - \hat{x}_{ij} \) is maximized over this polyhedron. If there exists a valid traffic matrix \( (\hat{d}_{st})_{s,t \in \Omega} \) that is not supported by the capacity \( \hat{x}_{ij} \) reserved on edge \( \{i, j\} \), that is, if such overload is positive, a new constraint (7) with coefficients \( \hat{d}_{st} \) is added to the formulation. If no such traffic matrix is found for any edge \( \{i, j\} \), the reserved capacities support all valid traffic matrices. Thus, these row-generation algorithms repeatedly improve a dual bound until primal feasibility is reached. Although the focus is on critical demand values, \( |E| \) linear programs must be solved at each iteration. In the
eral case of the hose model, these linear programs reduce to min-cost-flow problems, which yield an integral maximum flow on each edge [4]. In [4], computational results are reported for a special class of instances with unbounded entering traffic in each terminal. In [8], the authors consider multiple-path routing and check the validity of a reservation vector over a demand polyhedron with integral vertices. To the best of our knowledge, this does not actually imply that such a reservation vector supports all valid traffic matrices with fractional demands when single-path routing is considered. Examples where a reservation vector that is feasible for all valid integral traffic matrices does not support some valid fractional ones, can indeed be easily found.

2.1.2. A Compact Linear MIP Formulation for the Problem with Hose Model. Unlike in the above-mentioned works, we simultaneously consider all demand constraints and derive a compact linear MIP formulation that avoids the semi-infinite MIP formulation.

Proposition 1. The Asym-G problem (13)–(17) is equivalent to the following linear mixed-integer program:

$$\min \sum_{(i,j) \in E} c_{ij}x_{ij}$$  \hspace{1cm} (18)

s.t. \hspace{1cm} \sum_{j \in [i,j] \in E} (y_{ij}^u - y_{ij}^d) = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \hspace{1cm} \forall i \in V, \forall (s, t) \in S \hspace{1cm} (19)

$$\sum_{i \in Q} \left( b_{st}^+ \omega_{ij}^+ + b_{ti}^- \omega_{ij}^- \right) \leq x_{ij} \hspace{1cm} \forall \{i, j\} \in E \hspace{1cm} (20)$$

$$\omega_{ij}^+ + \omega_{ij}^- \geq (y_{ij}^u + y_{ij}^d) \hspace{1cm} \forall \{i, j\} \in E, \forall (s, t) \in S$$  \hspace{1cm} (21)

$$y_{ij}^u \in \{0, 1\} \hspace{1cm} \forall (i, j) \in A, \forall (s, t) \in S$$  \hspace{1cm} (22)

$$b_{st}^+ \omega_{ij}^+ + b_{ti}^- \omega_{ij}^- = 0 \hspace{1cm} \forall \{i, j\} \in E, \forall s \in Q \hspace{1cm} (23)$$

Proof. Start from the semi-infinite flow formulation (13)–(17). Consider a single edge \{i, j\} and treat the \(y\) variables as parameters. The worst-case value for the capacity \(x_{ij}\) to be reserved on edge \{i, j\} is obtained by solving the following optimization problem:

$$x_{ij} = \max \sum_{(i,j) \in S} d_{ij}(y_{ij}^u + y_{ij}^d)$$  \hspace{1cm} (24)

$$(\omega_{ij}^-) \sum_{r \in S} d_{rs} \leq b_{st}^+ \hspace{1cm} \forall s \in Q \hspace{1cm} (25)$$

$$d_{st} = 0 \hspace{1cm} \forall (s, t) \in S,$$  \hspace{1cm} (27)

where the \(\omega\)s are the corresponding dual variables. Because this linear program is feasible and bounded, we can apply a duality transformation similarly to [16] to obtain the equivalent formulation:

$$x_{ij} = \min \sum_{s \in Q} \left( b_{st}^+ \omega_{ij}^+ + b_{ti}^- \omega_{ij}^- \right)$$  \hspace{1cm} (28)

$$\omega_{ij}^+ + \omega_{ij}^- \geq y_{ij}^u + y_{ij}^d \hspace{1cm} \forall (s, t) \in S \hspace{1cm} (29)$$

$$\omega_{ij}^+, \omega_{ij}^- \geq 0 \hspace{1cm} \forall s \in Q.$$  \hspace{1cm} (30)

By replacing constraints (24)–(27) with (28)–(30), we obtain a lower bound on the capacity that is required on edge \{i, j\}. This substitution immediately leads from (13)–(17) to the linear MIP formulation (18)–(23) after observing that the min in (28) can be omitted due to the nature of the objective function, which is a nonnegatively weighted sum of variables \(x_{ij}\), and the continuous nature of the \(x_{ij}\) variables.

Analogously, for the Sym-G case we obtain a compact formulation with the same objective function (18), the flow conservation constraint (19), constraints (22), and

$$\sum_{i \in Q} b_{ji} \omega_{ij}^+ = x_{ij} \hspace{1cm} \forall \{i, j\} \in E \hspace{1cm} (31)$$

$$\omega_{ij}^+ + \omega_{ij}^- \geq y_{ij}^u + y_{ij}^d \hspace{1cm} \forall \{i, j\} \in E, \forall (s, t) \in S \hspace{1cm} (32)$$

$$\omega_{ij}^+, \omega_{ij}^- \geq 0 \hspace{1cm} \forall \{i, j\} \in E, \forall s \in Q.$$  \hspace{1cm} (33)

As we shall see in Section 4, tackling this compact linear MIP formulation with commercial solvers (e.g., Cplex 8.1) yields optimal solutions in reasonable time even for large-size instances.

2.1.3. Cutting Planes and Lower Bounds. Some valid inequalities can be easily derived for the formulations given in the preceding paragraphs. These inequalities are useful in cutting plane procedures for numerical solution of large instances as we shall see in Section 4.

Let us consider the flow formulation (18)–(23) of Asym-G. For any subset of edges \(F \subseteq E\), we use the notation \(x(F) = \sum_{(i,j) \in F} x_{ij}\) and, similarly, for any subset of arcs \(F' \subseteq A\) the notation \(y(F') = \sum_{(i,j) \in F'} y_{ij}^u\). For any subset of nodes \(W \subseteq V\), the cut \(\delta(W)\) is the set of edges with only one end-node in \(W\), that is, \(\delta(W) = \{ (i,j) \in E : i \in W, j \notin W \}\). For any subset \(W \subseteq V\), the directed cut \(\delta'(W)\) is the set of arcs with the tail in \(W\) and the head in
Given any subset $W \subset V$, if there exists a nonempty set of terminal pairs $S' = \{(s, t) \in S : s \in W, t \in \mathbb{V}\setminus W\}$, then for each one of these terminal pairs $(s, t)$ at least one flow variable associated with an arc of the directed cut $\delta'(W)$ must be nonzero, that is,

$$y^c(\delta'(W)) \geq 1. \quad (34)$$

From (21) we obtain the following family of inequalities:

$$\sum_{(i,j) \in E} (\omega^+_i + \omega^-_j) \geq 1 \quad \forall (s, t) \in S : \{s, t\} \cap W = 1 \quad (35)$$

that only involve $\omega$ variables and that are satisfied by any feasible solution of $\text{Asym-G}$. Consider now the capacity $x[\delta(W)]$ needed across a cut $\delta(W)$ to support the demands between all terminal pairs whose endpoints are on different shores. A lower bound is as follows:

$$x(\delta(W)) \geq \sum_{(i,j) \in S, |i| \cap W = 1} d_{ij}. \quad (36)$$

Given the traffic uncertainty, the right-hand side of (36) is not known a priori and $x[\delta(W)]$ has to be large enough in the worst case scenario. Because the maximum traffic across a cut $\delta(W)$, denoted by $d[\delta(W)]$, amounts to

$$\max\left\{ \sum_{(i,j) \in E, |i| \cap W = 1} d_{ij}, \sum_{i \in S} d_{is} \leq b^+_s \quad \forall s \in Q, \sum_{i \in S} d_{is} \leq b^-_s \quad \forall s \in Q, d_{st} = 0 \quad \forall (s, t) \in S \right\},$$

any feasible solution of $\text{Asym-G}$ must satisfy $x[\delta(W)] \geq d[\delta(W)]$. As $x_{ij}$ is defined in constraints (20), we can write equivalently:

$$\sum_{(i,j) \in E, |i| \cap W = 1} (b^+_i \omega^+_j + b^-_i \omega^-_j) \geq d[\delta(W)]. \quad (37)$$

The demand cut-set inequalities (35) and the capacity cut-set inequalities (37) express two necessary conditions for supporting all valid traffic matrices. Inequalities (35) and (37) are not violated by a solution of the linear relaxation of the formulation (18)–(23), as they are implied by the flow conservation constraints (19). However, a cut formulation including only the $\omega$ variables and a subset of these inequalities provides a lower bound on $\text{Asym-G}$. As for (37), we may get rid of the $x$ variables and express the objective function in terms of the $\omega$ variables only:

$$\sum_{(i,j) \in E} c_{ij} \sum_{t \in Q} (b^+_i \omega^+_j + b^-_i \omega^-_j). \quad (38)$$

The goal is then to minimize (38) subject to inequalities (35) and (37). A cutting-plane procedure for obtaining a lower bound on the optimal value of $\text{Asym-G}$ takes a small initial set of such inequalities and repeatedly adds violated cuts identified through efficient separation procedures. Although separating inequalities (35) requires solving a maximum-flow problem, and hence is easy, separating inequalities (37) can be shown to be NP-hard.

In the present article we derive dual (lower) bounds on the optimal value by using inequalities (35). Consider $G = (V, E)$ and the set of values $(b^+_i, b^-_i), \forall s \in Q$. Let $\Delta$ denote a subset of all possible triplets $\{s, t, \delta(W)\}$, where $W \subset V$ and $\{s, t\} \cap W = 1$. A lower bound is given by the following linear program:

$$\begin{align*}
(P_{\text{dual}}) \min & \sum_{(i,j) \in E} c_{ij} \sum_{(s,t) \in S} (b^+_i \omega^+_j + b^-_i \omega^-_j) \\
\text{s.t.} & \sum_{(i,j) \in E} (\omega^+_i + \omega^-_j) \geq 1 \quad \forall (s, t, W) \in \Delta.
\end{align*}$$

Because this problem does not involve the $y$ variables, it is quickly solved for a reasonable number of inequalities.

Inequalities (35) and (37) are easily adapted to $\text{Sym-G}$. From (32) and (34) we obtain the following two families of valid inequalities:

$$\sum_{(i,j) \in E} (\omega^+_i + \omega^-_j) \geq 1 \quad \forall W \subset V, (s, t) \in S : \{s, t\} \cap W = 1 \quad (39)$$

$$\sum_{(i,j) \in E, s \in Q} b_i \omega_i \geq d(\delta(W)) \quad \forall W \subset V. \quad (40)$$

2.2. Robust VPN Provisioning Problem

Because detailed traffic statistics (reliable average and standard deviation estimates of terminal-to-terminal traffic) are available for existing VPNs, we consider a variant of the VPN provisioning problem that exploits this additional information. Unlike the hose model, where bounds are imposed on the aggregated traffic through each terminal, we adopt a less conservative traffic uncertainty model and consider each demand $d_{st}$ individually. However, our assumptions regarding the valid traffic matrices are very mild: we just assume that we know the range of values that each demand $d_{st}$ can take. Specifically, for each pair of terminals $s, t \in Q$, the demand $d_{st}$ is assumed to take values in an interval $[d_{st} - \delta_{st}, d_{st} + \delta_{st}]$, where $\delta_{st} \in [0, d_{st}]$ denotes the positive deviation from the nominal value $d'_{st}$. It is worth emphasizing that we do not make any assumptions concerning the distribution of the traffic demands within the corre-
sponding intervals and concerning the way different demands \( d_i \), may be interrelated.

For nonnegative deviations \( \tilde{d}_{it} \), the robust VPN provisioning problem in which the demands are subject to the above interval uncertainty can be formulated as the following binary integer program:

\[
\min \sum_{(i,j) \in E} c_{ij} \sum_{(i,j) \in S} (d_{it} + \tilde{d}_{it})(y^u_{ij} + y^l_{ij})
\]

s.t.

\[
\sum_{(i,j) \in E} (y^u_{ij} - y^l_{ij}) = \begin{cases} 
1 & i = s \\
-1 & i = t \\
0 & \text{otherwise}
\end{cases} \quad \forall \, i \in V, \, \forall \, (s, t) \in S 
\]

\[
y^u_{ij} \in \{0, 1\} \quad \forall \, (i, j) \in A, \, \forall \, (s, t) \in S.
\]

As for the nominal problem (10)–(12) corresponding to the nominal traffic matrix \( D' = (d'_{it})_{i,t \in Q} \), this robust version can be solved in polynomial time.

Because it may be overly pessimistic to assume that all demands simultaneously vary within the corresponding uncertainty intervals so as to adversely affect the solution, we consider a positive integer parameter \( \Gamma, 1 \leq \Gamma \leq |S| \), which allows us to adjust the tradeoff between the VPN robustness and its degree of conservatism as in the general approach for robust discrete optimization problems under data uncertainty proposed in [5, 6]. Whereas minimum and maximum values for point-to-point demand can be obtained through repeated measurements over time, it is difficult to predict when a demand is at its maximum. Although in principle we cannot exclude the possibility that all demands take their peak values simultaneously, traffic statistics show that, in general, peaks in the demand values are reached at different times. It is then reasonable to limit the conservativeness of the model by adopting a robust approach à la Bertsimas and Sim [5, 6] and by assuming that at any point in time at most \( \Gamma \) of the demand values simultaneously take their worst-case values. In the VPN provisioning setting, the parameter \( \Gamma \) can be interpreted as the margin on the risk that the operator is willing to take for not satisfying the customer requirements over a given period of time.

To avoid overdimensioned VPNs, we minimize the maximum reservation cost while protecting us against the extreme behavior of at most \( \Gamma \) of the demands. More precisely, we consider the following robust VPN provisioning problem:

\[
\min \sum_{(i,j) \in E} c_{ij} x_{ij}
\]

s.t.

\[
\sum_{(i,j) \in E} (y^u_{ij} - y^l_{ij}) = \begin{cases} 
1 & i = s \\
-1 & i = t \\
0 & \text{otherwise}
\end{cases} \quad \forall \, i \in V, \, \forall \, (s, t) \in S
\]

\[
\sum_{(i,j) \in E} d_{it}(y^u_{ij} + y^l_{ij}) + \max_{\{S : \delta \subseteq S, \Gamma \leq |S|\}} \sum_{(i,j) \in S} \tilde{d}_{it}(y^u_{ij} + y^l_{ij}) \leq x_{ij} \quad \forall \, (i, j) \in E 
\]

\[
y^u_{ij} \in \{0, 1\} \quad \forall \, (i, j) \in A, \, \forall \, (s, t) \in S
\]

where \( \tilde{S} \) denotes a subset of \( S \) of cardinality at most \( \Gamma \). We refer to this robust variant of the problem as \( \text{Rob-G} \). It is worth pointing out that the general probabilistic guarantees derived for the extreme behavior of more than \( \Gamma \) parameters in [5] are also valid in our case. We also note that choosing \( \Gamma \) equal to the cardinality of \( S \) is equivalent to setting all demand values at their upper bounds, which yields the problem (39)–(41).

As for \( \text{Sym-G} \) and \( \text{Asym-G} \), this robust variant of the VPN provisioning problem admits a compact linear MIP formulation. The following result can be viewed as a special case of the first theorem in [5] for general discrete optimization problems.

**Proposition 2.** The robust VPN provisioning problem (42)–(46) has the following equivalent linear mixed-integer programming formulation:

\[
\min \sum_{(i,j) \in E} c_{ij} x_{ij}
\]

s.t.

\[
\sum_{(i,j) \in E} (y^u_{ij} - y^l_{ij}) = \begin{cases} 
1 & i = s \\
-1 & i = t \\
0 & \text{otherwise}
\end{cases} \quad \forall \, i \in V, \, \forall \, (s, t) \in S
\]

\[
\sum_{(i,j) \in E} d_{it}(y^u_{ij} + y^l_{ij}) + \Gamma \pi^0_{ij} + \sum_{(i,j) \in \tilde{S}} \pi^u_{ij} \leq x_{ij} \quad \forall \, (i, j) \in E
\]

\[
\pi^0_{ij} + \pi^u_{ij} \geq \tilde{d}_{it}(y^u_{ij} + y^l_{ij}) \quad \forall \, (i, j) \in E, \, \forall \, (s, t) \in S
\]

\[
y^u_{ij} \in \{0, 1\} \quad \forall \, (i, j) \in A, \, \forall \, (s, t) \in S
\]

**Proof.** As in [5, 6], we first consider the constraints whose parameters are subject to uncertainty, namely constraints (44). Given a vector \( \bar{y} \in \{0, 1\}^{|A| \times |S|} \), for each \( \{i, j\} \in E \) the problem

\[
\max \sum_{(i,j) \in E} \tilde{d}_{it}(\bar{y}^u_{ij} + \bar{y}^l_{ij})
\]
is equivalent to
\[
\max \sum_{(i,j) \in E} \alpha_{ij} \bar{d}_{ij} (\bar{y}_{ij} + \bar{y}_{ji}) \quad (54)
\]
\[\text{s.t.} \ \sum_{(i,j) \in E} \alpha_{ij} \leq \Gamma \quad (55)\]
\[\alpha_{ij} \in \{0, 1\} \ \forall (s, t) \in S. \quad (56)\]

The key observation here is that the above problem, viewed as a linear program by relaxing the integrality of \(\alpha_{ij}\) variables, always has a binary optimal solution as it is a simple knapsack. Therefore, it can be replaced with
\[
\max \sum_{(i,j) \in E} \alpha_{ij} \bar{d}_{ij} (\bar{y}_{ij} + \bar{y}_{ji}) \quad (57)
\]
\[\text{s.t.} \ \sum_{(i,j) \in E} \alpha_{ij} \leq \Gamma \quad (58)\]
\[0 \leq \alpha_{ij} \leq 1 \ \forall (s, t) \in S, \quad (59)\]

which admits the following dual:
\[
\min \Gamma s_t + \sum_{(i,j) \in E} \pi_{ij} \quad (60)
\]
\[\text{s.t.} \ \pi^0_{ij} + \pi_{ij} \geq \bar{d}_{ij} (\bar{y}_{ij} + \bar{y}_{ji}) \ \forall \{i, j\} \in E, \ \forall (s, t) \in S \quad (61)\]
\[\pi^0_{ij}, \pi_{ij} \geq 0 \ \forall \{i, j\} \in E, \ \forall (s, t) \in S. \quad (62)\]

By strong duality this dual is feasible and bounded, and has the same optimal objective function value as the primal problem (54)–(56). Substituting (60)–(62) in (42)–(46) yields (47)–(52).

Unlike the robust discrete optimization problems with 0–1 variables considered in [5], there is unfortunately strong evidence that the robust VPN provisioning problem is substantially harder to solve than the associated nominal problem, that is (39)–(41) with \(\bar{d}_{ij} = 0\) for all \((s, t) \in S\), which can be solved in polynomial time.

**Proposition 3.** The robust VPN provisioning problem (42)–(46) is NP-hard.

**Proof.** We proceed by polynomial time reduction from the following version of the Satisfiability problem.

3-SAT: Given a set of \(m\) clauses \(C_1, \ldots, C_m\) in \(n\) Boolean variables \(x_1, \ldots, x_n\) and their complements \(\bar{x}_1, \ldots, \bar{x}_n\) with exactly 3 literals per clause, does there exist a truth assignment for the variables that satisfies all the clauses?

Because each clause is a disjunction of its literals, such a truth assignment must make true at least one literal per clause. The 3-SAT problem is known to be NP-complete even when restricted to instances in which each Boolean variable \(x_j\) occurs (as \(x_j\) or \(\bar{x}_j\)) in at most five clauses [9]. The construction is similar to the one used to establish that it is NP-complete to decide whether in a given graph \(k\) distinct pairs of nodes can be connected with \(k\) edge-disjoint paths.

For any given instance of 3-SAT where each variable occurs in at most five clauses, we will construct a special instance of the robust VPN provisioning problem, defined by a graph \(G = (V, E)\) with \(c_{ij} = 1\) for all \(\{i, j\} \in E\), a set of terminals \(Q \subseteq V\), appropriate nominal demands and deviations between each pair of terminals and \(\Gamma = 1\). We will then verify that the former instance of 3-SAT is satisfiable if and only if the latter instance admits a robust VPN of total cost at most \(11n + 2m\).

Consider an arbitrary instance of 3-SAT in which each Boolean variable occurs in at most five clauses. For each variable \(x_j\) in this instance, we consider two terminals \(s_j, t_j\) in \(Q\) connected by two separate parallel paths corresponding, respectively, to the variable \(x_j\) and its complement \(\bar{x}_j\). Each one of these \((s_j, t_j)\)-paths consists of 11 edges and has 10 intermediate nodes. For each clause \(C_v\), we consider two other terminals \(v_i, v_j\) in \(Q\) connected by three separate parallel paths that correspond to the three literals in the clause. Each one of these \((v_i, v_j)\)-paths consists of three edges and has two intermediate nodes. The paths associated with the pairs of terminals \((s_j, t_j), 1 \leq j \leq n\), and \((v_i, v_j), 1 \leq i \leq m\), must intersect in the appropriate way. For example, if the 3-SAT instance contains a first clause \(C_{i_1}\) with the literals \(x_{i_1}, \bar{x}_{i_1}\) and \(x_{\bar{i}_1}\), and a second clause \(C_{i_2}\) with the literals \(\bar{x}_{i_2}, x_{i_2}\) and \(x_{\bar{i}_2}\), the three separate parallel paths associated with the three literals of each clause are constructed as shown in Figure 2. The first \((v_{i_1}, v_{i_2})\)-path shares exactly one edge (the second one) with the path connecting terminals \(s_{i_1}\) and \(t_{i_1}\) that corresponds to variable \(x_{i_1}\); the second \((v_{i_1}, v_{\bar{i}_2})\)-path shares exactly one edge (the second one) with the path connecting \(s_{i_1}\) and \(t_{\bar{i}_2}\) that corresponds to \(\bar{x}_{i_1}\); the third \((v_{i_1}, v_{\bar{i}_2})\)-path shares exactly one edge (the second one) with the path connecting \(s_{i_1}\) and \(t_{\bar{i}_2}\) that corresponds to \(x_{\bar{i}_1}\). The three \((v_{i_1}, v_{i_2})\)-paths corresponding to the three literals of the second clause are constructed accordingly. Note that, because the variable \(x_{i_1}\) occurs in both clauses, the \((s_{i_1}, t_{\bar{i}_1})\)-path associated with \(x_{i_1}\) (the first one) shares its second edge with the third \((v_{i_1}, v_{i_2})\)-path and its fourth edge with the third \((v_{i_1}, v_{i_2})\)-path.

Because each Boolean variable occurs in at most five clauses and each \((s_j, t_j)\)-path consists of 11 edges, we can make sure that each edge of any \((s_j, t_j)\)-path (associated with \(x_j\) or \(\bar{x}_j\)) belongs to at most one \((v_i, v_j)\)-path corresponding to a literal of a clause. Note that all paths in \(G\), which connect any \((s_j, t_j)\) (respectively \((v_i, v_j)\)) terminal pair but are not \((s_j, t_j)\)-paths (respectively \((v_i, v_j)\)-paths), contain more than 11 (respectively 3) edges.

The special instance of the robust VPN provisioning
problem corresponding to the given 3-SAT instance is defined, together with the above graph \( G \) and set of terminals \( Q \subseteq V \), by the nominal demands \( d''_{sxij} = 0 \) for all \( j \) and \( d''_{wvwi} = 0 \) for all \( i \), and the deviations \( \hat{d}_{sxij} = 1 \) for all \( j \) and \( \hat{d}_{wvwi} = 1 \) for all \( i \). The nominal demands and deviations for all other pairs of terminals in \( Q \) are set to zero.

Because \( \Gamma = 1 \) and by construction each \((v_j, w_i)\)-path shares exactly one edge with one of the \((s_j, t_j)\)-paths, asking whether there exists a robust VPN with total reservation cost at most \( 11n \) means deciding whether it is possible to route a unit of flow from \( s_j \) to \( t_j \) (i.e., to select one of the two alternative \( 11 \)-edge \((s_j, t_j)\)-paths) for every \( j \), \( 1 \leq j \leq n \), and a unit of flow from \( v_i \) to \( w_j \) (i.e., to select one of the three alternative \( 3 \)-edge \((v_i, w_j)\)-paths) for every \( i \), \( 1 \leq i \leq m \), so that for each \( i \) the selected \((v_i, w_j)\)-path shares exactly one edge with a selected \((s_j, t_j)\). Note that, due to unit costs, the total reservation cost cannot be smaller than \( 11n + 3m - m = 11n + 2m \). Indeed, a path with at least \( 11 \) (respectively \( 3 \)) edges is needed to connect each one of the \( n \) (respectively \( m \)) terminal pairs \((s_j, t_j)\) [respectively \((v_i, w_j)\)] and each selected \((v_i, w_j)\)-path shares at most one edge with one of the selected \((s_j, t_j)\)-paths.

Given any truth assignment for the 3-SAT instance, it is straightforward to select appropriate routing paths for the \( n \) pairs of terminals \((s_j, t_j)\) and \( m \) pairs of terminals \((v_i, w_j)\) based on the truth values of the variables and literals (at least one literal in each clause is true). Conversely, it is easy to derive from any such \( n + m \) paths [where each selected \((v_i, w_j)\)-path shares exactly one edge with one of the selected \((s_j, t_j)\)-paths] a truth assignment for the original instance of 3-SAT.

Because demands are assumed to be routed along single paths, the reduction still holds when the VPN provisioning problem is restricted to instances with nonzero nominal demands \( d'' \) for all \((s_j, t_j)\) and \((v_i, w_j)\) terminal pairs. If we consider instances with \( d''_{sxij} = 1 \) and \( d''_{wvwi} = 1 \) for all \( j \) and \( i \), then the question is whether there exists a robust VPN of total cost at most \( 22n + 4m \).

Note that an important difference with respect to the 0–1 robust discrete optimization problems considered in [5] lies in the fact that here each uncertain parameter \( \hat{d}_{sxij} \) multiplies several binary variables \( y_{ij} \), see (39) and (44), instead of a single one.

In Section 4 we report computational results obtained by tackling this compact linear MIP formulation for medium-to-large-size instances with the Cplex 8.1 MIP solver.

**2.2.1. Cutting Planes.** Valid inequalities similar to those in Section 2.1.3 can also be derived for Rob-G. Consider the relaxation of (53) equivalent to (57)–(59):

\[
\max \sum_{(s, t) \in S} d_{st}(\bar{\gamma}_{st} + \bar{\gamma}_{st})
\]

\[
\text{s.t.}
\]

\[
(\pi_0^0) \sum_{(i, j) \in S} \frac{d_{st} - d_{st}'}{\bar{d}_{st}} \leq \Gamma
\]

\[
(\sigma_{ij}) \quad -d_{ij} \leq -d_{ij}^{\bar{d}} \quad \forall (s, t) \in S
\]

\[
(\rho_{ij}) \quad d_{ij} \leq d_{ij}^{\bar{d}} + \hat{d}_{ij} \quad \forall (s, t) \in S,
\]

where \( \pi_0^0, \sigma_{ij}^0 \) and \( \rho_{ij}^0 \) are the corresponding dual variables. It is easy to show that constraints (49) and (50) can be replaced by

\[
\left( \Gamma + \sum_{(i, j) \in S} \frac{d_{ij}'}{\hat{d}_{ij}} \right) \pi_{ij}^0 + \sum_{(i, j) \in S} \left( (d_{ij}^{\bar{d}} + \hat{d}_{ij}) \rho_{ij}^0 - d_{ij}^{\bar{d}} \sigma_{ij}^0 \right) \leq \bar{y}_{ij} \quad \forall \{i, j\} \in E
\]

\[
\pi_{ij}^0 \hat{d}_{ij} - \sigma_{ij}^0 + \rho_{ij}^0 = \bar{y}_{ij} + \bar{y}_{ij}^{\bar{d}} \quad \forall \{i, j\} \in E, \quad \forall (s, t) \in S
\]

\[
\pi_{ij}^0 \sigma_{ij}^0 + \rho_{ij}^0 = 0 \quad \forall \{i, j\} \in E, \quad \forall (s, t) \in S
\]

and that the two types of valid inequalities for Rob-G are:

\[
\sum_{(i, j) \in E(W)} (\pi_{ij}^0 \hat{d}_{ij} - \sigma_{ij}^0 + \rho_{ij}^0) \geq 1 \quad \forall W \subseteq V,
\]

\[
\sum_{(i, j) \in E(W)} \left( (\Gamma + \sum_{(i, j) \in S} \frac{d_{ij}'}{\hat{d}_{ij}}) \pi_{ij}^0 + \sum_{(i, j) \in S} \left( (d_{ij}^{\bar{d}} + \hat{d}_{ij}) \rho_{ij}^0 - d_{ij}^{\bar{d}} \sigma_{ij}^0 \right) \right) \quad \forall W \subseteq V.
\]

**2.3. Path Formulations.**

The flow-based models of the preceding sections quickly become too large for medium-to-large network design in-
stances. The resulting test problems are usually too time-
consuming for numerical solution by off-the-shelf MIP
solvers. An alternative approach is to adopt path-based
formulations suitable for column generation algorithms.

Let us consider path variables instead of flow variables.
As traffic is unsplittable, these variables are binary. Further
notation is needed: \( P \) denotes the set of all possible oriented
paths between any two given nodes in \( Q \), \( P_{ij} \) the set of all
paths in \( P \) containing the edge \( \{i, j\} \in E \), and \( P^{st} \) the set
of paths between the two demand nodes \( s \) and \( t \).

For each path \( p \in P \), we consider a binary variable \( z_{tp} \)
that is equal to 1 if and only if the path \( p \), implicitly defined
between two nodes \( s \) and \( t \), is used to satisfy the demand
\( d_{st} \). The initial semi-infinite path formulation for Asym-G
can then be expressed as follows:

\[
\begin{align*}
\min & \sum_{(i,j) \in E} c_{ij} x_{ij} \\
\text{s.t.} & \sum_{p \in P^{st}} z_{tp} \geq 1 \quad \forall (s, t) \in S \\
& \sum_{(i,j) \in E} d_{st} \sum_{p \in P^{st}} z_{tp} \leq x_{ij} \quad \forall D \in U_{AsymG}, \forall \{i,j\} \in E \\
& z_{tp} \in \{0, 1\} \quad \forall p \in P
\end{align*}
\]

(63)

Constraints (64) ensure demand satisfaction by imposing
that at least one path is used between each terminal pair,
while constraints (65) define sufficient capacity on each
edge \( \{i,j\} \) to support all traffic matrices in the uncertainty
set \( U_{AsymG} \). As in Section 2.1.2, we obtain the linear mixed-
integer path-based formulation for Asym-G:

\[
\begin{align*}
\min & \sum_{(i,j) \in E} c_{ij} x_{ij} \\
\text{s.t.} & \sum_{p \in P^{st}} z_{tp} \geq 1 \quad \forall (s, t) \in S \\
& \sum_{(i,j) \in E} (b_i^+ \omega_{ij}^+ + b_j^- \omega_{ij}^-) \leq x_{ij} \quad \forall \{i,j\} \in E \\
& \omega_{ij}^+ + \omega_{ij}^- \geq \sum_{p \in P^{st}} z_{tp} \quad \forall (s, t) \in S, \forall \{i,j\} \in E \\
& z_{tp} \in \{0, 1\} \quad \forall (s, t) \in S, \forall p \in P^{st} \\
& \omega_{ij}^+, \omega_{ij}^- \geq 0 \quad \forall \{i,j\} \in E, \forall s \in Q.
\end{align*}
\]

(68)

(70)

\( A \) similar linear path-based formulation is obtained for
Rob-G:

\[
\begin{align*}
\min & \sum_{(i,j) \in E} c_{ij} x_{ij} \\
\text{s.t.} & \sum_{p \in P^{st}} z_{tp} \geq 1 \quad \forall (s, t) \in S \\
& \sum_{(i,j) \in E} d_{st} \sum_{p \in P^{st}} z_{tp} + \Gamma \pi_{ij}^0 + \sum_{(i,j) \in E} \pi_{ij}^s \leq x_{ij} \quad \forall \{i,j\} \in E \\
& \pi_{ij}^s \geq \sum_{p \in P^{st}} z_{tp} - \pi_{ij}^s \quad \forall \{i,j\} \in E, \forall (s, t) \in S \\
& z_{tp} \in \{0, 1\} \quad \forall (s, t) \in S, \forall p \in P^{st} \\
& \pi_{ij}^s, \pi_{ij}^s \geq 0 \quad \forall \{i,j\} \in E, \forall (s, t) \in S.
\end{align*}
\]

The above path-based formulations are equivalent to
their flow-based counterparts in Proposition 1 and Proposition
2.

3. AN EXACT SOLUTION METHOD

Because practical VPNs can contain hundreds of termi-
nals in networks with thousands of nodes, and our flow
formulations of Asym-G, Sym-G, and Rob-G are not viable
for networks of that size, we use the more flexible path
formulations, described in Section 2.3, and adopt a column
generation approach. As we are facing linear mixed-integer
programs, the path formulations are tackled within a branch-
and-price framework discussed in Section 3.1. To evaluate
the quality of the upper bounds found by the branch-and-
price algorithm, we devise a cutting-plane procedure that
provides lower bounds. This procedure, based on the dis-
cussion in Sections 2.1.3 and 2.2.1, and described below,
also helps to deal with the well-known tailing off effect (i.e.,
the generation of spurious columns—paths—that do not
improve the objective function value of the linear relax-
which can substantially affect the performance of column generation. Cutting planes are not used throughout the branch-and-price algorithm, but only in the initial column generation phase performed at the root node: after each pricing iteration, we run one iteration of the cutting-plane procedure.

### 3.1. A Branch-and-Price and Cutting-Plane Algorithm

To take advantage of the path formulations and solve large practical instances of Asym-G, Sym-G, and Rob-G, we combine column generation and branch and bound. This joint approach, known as branch and price, has been introduced by Barnhart et al. [3] to solve large integer programs; an application to a multicommodity flow problem with integer flow variables can be found in [2]. Although in the following we only describe this approach for Asym-G, the versions for Sym-G and Rob-G are easily derived.

We start from a path formulation (68)–(73) with a small set $P_0^*$ of paths for each terminal pair $(s, t)$. At least one path per terminal pair is needed to ensure feasibility, but we take the $K$ shortest paths from $s$ to $t$ with respect to the edge costs $c_{ij}$. Relaxing integrality yields the Restricted Linear Problem (RLP) that we solve through the primal simplex method, retrieving the values of the dual variables $\tilde{\sigma}_{st}$, $\tilde{\xi}_{ij}$, $\tilde{\eta}_{ij}$ of constraints (69), (70), and (71), respectively.

Consider a terminal pair $(s, t)$ and a variable $z_{p}$ corresponding to a path $p \in P^*$ that has reduced cost $-\tilde{\sigma}_{st} + \sum_{(i,j) \in p} \tilde{\eta}_{ij}$. All variables included in RLP have zero or positive reduced cost after the primal simplex algorithm is called, but there may exist a $z_{p}$, not included in the current formulation, with a negative reduced cost. Seeking one such variable for each terminal pair $(s, t)$ means solving the so-called pricing problem. We do not enumerate all paths in the set of remaining paths $P^{\text{new}}(P_0^*)$, as its cardinality may be exponential in the number of vertices. Instead, we solve a shortest path problem on a directed graph $G_{ij}(s, t)$ whose arcs $(i, j)$ correspond to edges in $G$ and have cost $\tilde{\eta}_{ij}$. If the length of such a shortest path $p$ is smaller than $\tilde{\sigma}_{st}$, then $z_{p}$ has negative reduced cost and can be included in RLP.

The column generation procedure can be summarized as follows:

1. Build a formulation (68)–(73) with the set $P_0^*$ of $K$ shortest paths for every $(s, t) \in S$.
2. Obtain RLP by relaxing the integrality constraints.
3. Solve RLP, obtaining a primal solution $(\bar{x}_{ij}, \bar{\omega}_{ij}^+, \bar{\omega}_{ij}^-, \bar{z}_{p})$ and the associated dual solution $(\bar{\sigma}_{st}, \bar{\xi}_{ij}, \bar{\eta}_{ij})$.
4. For each terminal pair $(s, t) \in S$:
   - Build an auxiliary undirected graph $G_{ij}(s, t)$ such that the length of each edge is $\bar{\eta}_{ij}$;
   - Find the shortest path $\bar{p}$ in $G_{ij}(s, t)$ and let the length of $\bar{p}$ be $\bar{l}_{p}(s, t)$;
   - If $\bar{l}_{p}(s, t) < \bar{\sigma}_{st}$, then add the related variable $z_{p}$ to RLP;
5. If at least one variable has been added, go to Step 3.

The procedure is applied until no new path with negative reduced cost is found. Upon termination the current solution is an optimal solution of RLP. If one or more $z_{p}$ variables are fractional, we divide the RLP by applying a branching step. The resulting subproblems are relaxed and solved with the technique described above, and so are all subproblems corresponding to nodes in the branch-and-bound tree.

It is worth pointing out that branching on one variable $z_{p}$ may unbalance the branch-and-bound tree, or even make the algorithm loop: if a variable $z_{p}$ is fixed to zero but the pricing procedure finds a path $p' = p$ with negative reduced cost, a variable $z_{p'}$ is added even if path $p$ is forbidden. Therefore, the following branching rule is used at a branch-and-bound node $N_k$ with at least one fractional $z_{p}$ variable related to a terminal pair $(s, t)$. We choose an edge $e$ contained in $p$, then create two new nodes $N_{k+1}$ and $N_{k+2}$, such that all path variables in $N_{k+1}$ for terminal pair $(s, t)$ associated with paths containing $e$ are set to zero, while in $N_{k+2}$ their sum must be equal to one. Once the constraints

$$\sum_{p \in P^{\text{new}}(s, t)} z_{p} = 0 \quad \text{and} \quad \sum_{p \in P^{\text{add}}(s, t)} z_{p} = 1$$

are added to the subproblems associated to $N_{k+1}$ and respectively $N_{k+2}$, the pricing problem is solved by a shortest path computation that takes into account the branching constraints at upper levels of the branch-and-bound tree.

As a result of a major shortcoming of column generation techniques, the tailing off effect, several columns with negative reduced cost are generated even when an optimal solution is reached, requiring unnecessary extra running time. We have used the cutting planes (35) described in Section 2 to obtain a lower bound and close the gap w.r.t. the upper bound given by column generation.

We consider a combined branch-and-price and cutting-plane algorithm, referred to as BPC, where an iteration of the cutting-plane procedure is run after each pricing iteration of the column generation. For each terminal pair $(s, t)$, a directed cut $\delta'(W)$ is sought such that $\sum_{(i,j) \in \delta'(W)} (\omega_{ij}^+ + \omega_{ij}^-)$ is minimum, which amounts to solving the maximum $(s, t)$-flow problem on a graph whose edges $\{i, j\}$ have capacity $(\omega_{ij}^+ + \omega_{ij}^-)$. If the max-flow obtained is less than one, the cutting plane is inserted.

We have observed a dramatic improvement by adding a limited number of cutting planes, whose separation is equivalent to the max-flow problem. Using the notation of Section 2.1.3 the cutting plane procedure can be outlined as follows:

1. Let $\Delta := \emptyset$; $k := 0$;
2. Solve $(P_{\text{dual}})$, obtaining values $\bar{\omega}_{ij}^+$ and $\bar{\omega}_{ij}^-$ for all $(i, j) \in E$ and $s \in Q$;
3. feasible := TRUE;
4. For each $(s, t) \in S$:
   - Let $G_{st} := (V, A)$ be an auxiliary graph where $A = \{(i, j) : (i, j) \in E\}$ and let $u_{ij} = u_{ji} = \omega_{ij}^+ + \omega_{ij}^-$ for all $(i, j) \in A$;
• Solve the maximum-flow/minimum-cut problem over \( G_{st} = (V, A) \) with capacities \( u_{ij} \) and let \( \delta(W) \) be a minimum capacity cut;
• If the capacity of \( \delta(W) \) is smaller than 1, set \( \Delta := \Delta \cup (s, t, W) \) and \( \text{feasible} := \text{FALSE} \);
5. If \( \text{feasible} = \text{TRUE} \), then stop, otherwise go to Step 2.

4. COMPUTATIONAL RESULTS

We have tested the compact linear MIP formulations and our BPC algorithm for Sym-G, Asym-G and Rob-G on medium to large-size network topologies. The following instances have been considered:

- **bhv3/6/13**: instances of a multicommodity flow problem studied in [2];
- **arpanet, eon, latadl, nsf, pachell, toronto, usld**: topologies of well-known backbone networks found in the IEEE literature;
- **res15/67/8/9, at-cep, ny-cep, nor-sun**: instances of different multicommodity flow problems found at http://www.di.unipi.it/di/groups/optimize/Data/MMCF.html#Rsrv;
- **stein1/2/3/4**: a set of Steiner tree problem instances with 50 and 75 nodes, available at the Web page http://www.brunel.ac.uk/depts/ma/research/jeb/orlib/steininfo.html;
- **n45, n49, n147**: general multicommodity flow instances;
- **g200**: a network topology with 200 nodes and 914 links (instance originating from Dan Bienstock and provided by Pasquale Avella, 2004);
- **t3-X, t4-X**: a set of backbone networks with 250 and 304 nodes randomly generated with the gt-itm software (http://www.cc.gatech.edu/projects/gtitm/gt-itm/).

All instances are available from ftp://ftp.elet.polimi.it/users/Pietro.Belotti/mcf/vpn in a pseudo-DIMACS format and in AMPL data format.

Because single VPNs are considered, the set \( Q \) of terminals has been randomly selected as a small subset of \( V \) with a density \( |Q|/|V| \) of approximately 15%. The values of \( b_s, b^+_t, b^-_t, d^-_s, \) and \( d^+_s \) have been generated based on realistic random traffic matrices.

All tests have been carried out with a Pentium Xeon 2.8 GHz processor, running under Linux with 2 GB of memory available. The BPC algorithm has been implemented in C++ using the open-source COIN/Bcp software framework (http://www.coin-or.org/documentation.html#BCP) and Cplex 8.1 as LP solver, while Cplex 8.1 MIP solver has been used for our compact formulations. Initially we generate, for each terminal pair \((s,t)\), \( K \) shortest paths from \( s \) to \( t \) through the HREA algorithm described in [13].

The computational results are summarized in five tables that include the following information:

- the instance name and the network parameters (the cardinalities of \( V, E \), and of the set of terminals \( Q \subseteq V \)),
- the time \( (t_{root}) \) spent by column generation in the root node, that is, the time required to solve the relaxed problem, starting with \( K = 5 \) paths for each terminal pair,
- the time \( (t_{tot}) \) spent in total by the BPC algorithm,
- the time \( (t_{cplex}) \) spent by the Cplex 8.1 MIP solver on the linear flow-based compact formulation,
- \#paths and \#cuts: the number of paths and cuts generated by the pricing and the cutting-plane iterations.

All times are expressed in seconds.

A time limit of 2 hours has been given both for the BPC algorithm and for tackling the compact flow-based formulations with the Cplex 8.1 MIP solver. If the time limit is reached before obtaining an optimal solution, we report in brackets the gap between the best feasible solution and the best lower bound that have been found. A “—” indicates that no feasible solution was found (in the \( t_{tot} \) column it means that the branch-and-bound part was not performed). The label “mem” indicates that the problem could not be solved because of excessive memory requirements, namely more than 2 GB of RAM. In the BPC algorithm, if the time limit is reached while solving the linear relaxation, the gap is given in brackets in the \( t_{root} \) column.

According to Table 1, the small-size Sym-G instances (with up to 40 nodes) are solved very rapidly and the performance of Cplex 8.1 on the compact linear MIP formulation and of the BPC algorithm are comparable in most cases.

The computational results obtained for larger Sym-G instances are reported in Table 2. Our compact flow-based formulation turns out to be very tight so as to yield the optimal solution within less than a minute even for the large problem “n147.” For larger instances, however, it pays to develop a specialized combined branch-and-price and cutting-plane method because the compact linear MIP formulation leads to excessive memory requirements. Our BPC algorithm, which performs better on “n147” and on a few smaller instances such as “n45” and “stein2,” allows us to solve “g200” and “t3-3” optimally within the 2-hour time limit and to tackle larger instances. Unlike other experiments, for “t3-0” we have set \( K = 20 \) (instead of \( K = 5 \)) and let the BPC method run for 252286.68 sec. The gap is in fact reduced to 3.24% after 79293.58 seconds and then progressively to 0.95%. Tuning the value of \( K \) and allowing for more than 2 hours of computing time, we can also obtain gaps below a few percentage points for the other large instances. Note that, for all instances that have been solved to optimality, an optimal solution is already found at the root node of the branch-and-price tree. The BPC algorithm stops after the initial column generation phase because the solution found has the same value as the optimal integer solution. In very few cases branch-and-bound steps are performed (and the branch-and-price time taken after the RLP is solved, \( t_{tot} - t_{root} \), is negligible in most cases) and they do not improve the lower bound. Because the number of paths that are generated during the column generation phase is moderate with respect to the number of terminal
pairs, the approach is viable even for larger instances. The insertion of cutting planes has a dramatic effect in limiting the tailing off of the objective function value when additional paths are added, thus helping to close the gap within a very short computing time. Notice that the number of cuts separated at the root node is the same order of magnitude as the number of paths.

Another interesting result emerged from preliminary experiments performed on small, randomly generated Sym-G instances. We have observed that for most instances even the linear relaxation has an integer solution, that is, all flow variables $y$ are either 0 or 1; in the few cases where at least one $y$ variable is fractional, restoring the integrality constraints gives an integer solution with the same cost. Moreover, for most Sym-G instances the edges $(i, j)$ with positive capacity $x_{ij}$ formed a tree; in the remaining cases, imposing a tree structure on the solution, that is, solving the related Sym-T problem, yielded a solution with the same cost. Thus, our experimental results support the following conjecture:

**Sym-G always admits an optimal tree solution.**

The same conjecture was also made by Erlebach and Rüegg

<table>
<thead>
<tr>
<th>Name</th>
<th></th>
<th></th>
<th></th>
<th>$t_{root}$</th>
<th>$t_{tot}$</th>
<th>#paths</th>
<th>#cuts</th>
<th>$t_{cf}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>arpanet</td>
<td>24</td>
<td>50</td>
<td>10</td>
<td>11.03</td>
<td>11.03</td>
<td>1219</td>
<td>1055</td>
<td>1.97</td>
</tr>
<tr>
<td>at-cep</td>
<td>15</td>
<td>22</td>
<td>6</td>
<td>0.02</td>
<td>0.02</td>
<td>211</td>
<td>100</td>
<td>0.03</td>
</tr>
<tr>
<td>bhv3</td>
<td>29</td>
<td>62</td>
<td>15</td>
<td>2.65</td>
<td>—</td>
<td>1695</td>
<td>1498</td>
<td>8.34</td>
</tr>
<tr>
<td>bhv6</td>
<td>27</td>
<td>39</td>
<td>15</td>
<td>0.14</td>
<td>0.03</td>
<td>1123</td>
<td>648</td>
<td>0.13</td>
</tr>
<tr>
<td>bhv13</td>
<td>29</td>
<td>36</td>
<td>13</td>
<td>0.48</td>
<td>0.04</td>
<td>1027</td>
<td>1212</td>
<td>0.86</td>
</tr>
<tr>
<td>cost239</td>
<td>11</td>
<td>22</td>
<td>5</td>
<td>0.02</td>
<td>0.02</td>
<td>129</td>
<td>46</td>
<td>0.07</td>
</tr>
<tr>
<td>eon</td>
<td>19</td>
<td>37</td>
<td>15</td>
<td>0.49</td>
<td>0.49</td>
<td>1272</td>
<td>714</td>
<td>1.28</td>
</tr>
<tr>
<td>latadl</td>
<td>39</td>
<td>86</td>
<td>17</td>
<td>397.27</td>
<td>397.27</td>
<td>3917</td>
<td>4259</td>
<td>219.55</td>
</tr>
<tr>
<td>metro</td>
<td>11</td>
<td>42</td>
<td>5</td>
<td>0.01</td>
<td>0.01</td>
<td>126</td>
<td>44</td>
<td>0.26</td>
</tr>
<tr>
<td>njlata</td>
<td>11</td>
<td>23</td>
<td>8</td>
<td>0.14</td>
<td>0.14</td>
<td>405</td>
<td>212</td>
<td>0.33</td>
</tr>
<tr>
<td>nosun</td>
<td>27</td>
<td>51</td>
<td>13</td>
<td>70.53</td>
<td>70.53</td>
<td>2019</td>
<td>1604</td>
<td>64.24</td>
</tr>
<tr>
<td>nsf</td>
<td>14</td>
<td>21</td>
<td>10</td>
<td>0.45</td>
<td>0.45</td>
<td>672</td>
<td>422</td>
<td>0.86</td>
</tr>
<tr>
<td>ny-cep</td>
<td>16</td>
<td>49</td>
<td>9</td>
<td>0.71</td>
<td>0.71</td>
<td>573</td>
<td>354</td>
<td>0.75</td>
</tr>
<tr>
<td>pacbell</td>
<td>15</td>
<td>21</td>
<td>7</td>
<td>0.09</td>
<td>0.09</td>
<td>294</td>
<td>176</td>
<td>0.16</td>
</tr>
<tr>
<td>toronto</td>
<td>25</td>
<td>55</td>
<td>11</td>
<td>2.53</td>
<td>2.53</td>
<td>1026</td>
<td>887</td>
<td>6.27</td>
</tr>
<tr>
<td>usld</td>
<td>28</td>
<td>45</td>
<td>15</td>
<td>7.55</td>
<td>7.55</td>
<td>1753</td>
<td>1496</td>
<td>4.17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th></th>
<th></th>
<th></th>
<th>$t_{root}$</th>
<th>$t_{tot}$</th>
<th>#paths</th>
<th>#cuts</th>
<th>$t_{cf}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>res1</td>
<td>45</td>
<td>63</td>
<td>15</td>
<td>36.62</td>
<td>43.39</td>
<td>2843</td>
<td>3533</td>
<td>32.41</td>
</tr>
<tr>
<td>res5</td>
<td>44</td>
<td>67</td>
<td>18</td>
<td>174.21</td>
<td>174.21</td>
<td>3895</td>
<td>4641</td>
<td>58.78</td>
</tr>
<tr>
<td>res6</td>
<td>44</td>
<td>60</td>
<td>14</td>
<td>93.64</td>
<td>93.64</td>
<td>2820</td>
<td>2641</td>
<td>6.63</td>
</tr>
<tr>
<td>res7</td>
<td>44</td>
<td>62</td>
<td>18</td>
<td>2920.21</td>
<td>2920.21</td>
<td>11,341</td>
<td>10,019</td>
<td>37.68</td>
</tr>
<tr>
<td>res8</td>
<td>50</td>
<td>79</td>
<td>19</td>
<td>184.27</td>
<td>184.27</td>
<td>4860</td>
<td>6406</td>
<td>92.71</td>
</tr>
<tr>
<td>res9</td>
<td>50</td>
<td>76</td>
<td>23</td>
<td>1048.50</td>
<td>74.24</td>
<td>11,467</td>
<td>12,620</td>
<td>107.16</td>
</tr>
<tr>
<td>stein1</td>
<td>50</td>
<td>100</td>
<td>17</td>
<td>40.99</td>
<td>40.99</td>
<td>2778</td>
<td>2657</td>
<td>43.65</td>
</tr>
<tr>
<td>stein2</td>
<td>50</td>
<td>63</td>
<td>20</td>
<td>1.46</td>
<td>1.46</td>
<td>2488</td>
<td>2586</td>
<td>1.48</td>
</tr>
<tr>
<td>stein3</td>
<td>75</td>
<td>94</td>
<td>32</td>
<td>97.07</td>
<td>97.07</td>
<td>8192</td>
<td>16,084</td>
<td>61.88</td>
</tr>
<tr>
<td>stein4</td>
<td>75</td>
<td>150</td>
<td>33</td>
<td>3654.09</td>
<td>3654.09</td>
<td>11,124</td>
<td>16,411</td>
<td>3443.33</td>
</tr>
<tr>
<td>n45</td>
<td>45</td>
<td>63</td>
<td>20</td>
<td>5.89</td>
<td>5.89</td>
<td>3187</td>
<td>4512</td>
<td>38.62</td>
</tr>
<tr>
<td>n49</td>
<td>49</td>
<td>57</td>
<td>14</td>
<td>1.03</td>
<td>1.03</td>
<td>1357</td>
<td>1378</td>
<td>0.98</td>
</tr>
<tr>
<td>n147</td>
<td>147</td>
<td>265</td>
<td>37</td>
<td>18.80</td>
<td>18.80</td>
<td>7668</td>
<td>8211</td>
<td>52.72</td>
</tr>
<tr>
<td>g200</td>
<td>200</td>
<td>914</td>
<td>31</td>
<td>58.01</td>
<td>5786</td>
<td>6546</td>
<td>63.42</td>
<td>mem</td>
</tr>
<tr>
<td>t3-0</td>
<td>250</td>
<td>444</td>
<td>52</td>
<td>(0.95%)</td>
<td>—</td>
<td>97,025</td>
<td>246,679</td>
<td>mem</td>
</tr>
<tr>
<td>t3-1</td>
<td>250</td>
<td>456</td>
<td>53</td>
<td>(17.24%)</td>
<td>—</td>
<td>33,164</td>
<td>141,516</td>
<td>mem</td>
</tr>
<tr>
<td>t3-2</td>
<td>250</td>
<td>469</td>
<td>42</td>
<td>(1.30%)</td>
<td>—</td>
<td>44,395</td>
<td>220,123</td>
<td>mem</td>
</tr>
<tr>
<td>t3-3</td>
<td>250</td>
<td>466</td>
<td>50</td>
<td>6959.42</td>
<td>6959.42</td>
<td>48,190</td>
<td>363,584</td>
<td>mem</td>
</tr>
<tr>
<td>t3-4</td>
<td>250</td>
<td>465</td>
<td>39</td>
<td>(0.71%)</td>
<td>—</td>
<td>40,939</td>
<td>122,825</td>
<td>mem</td>
</tr>
<tr>
<td>t4-0</td>
<td>304</td>
<td>453</td>
<td>55</td>
<td>(0.02%)</td>
<td>—</td>
<td>42,524</td>
<td>293,763</td>
<td>mem</td>
</tr>
<tr>
<td>t4-1</td>
<td>304</td>
<td>457</td>
<td>63</td>
<td>(7.69%)</td>
<td>—</td>
<td>28,710</td>
<td>114,345</td>
<td>mem</td>
</tr>
<tr>
<td>t4-2</td>
<td>304</td>
<td>454</td>
<td>58</td>
<td>(19.30%)</td>
<td>—</td>
<td>28,335</td>
<td>96,122</td>
<td>mem</td>
</tr>
<tr>
<td>t4-3</td>
<td>304</td>
<td>458</td>
<td>67</td>
<td>(26.88%)</td>
<td>—</td>
<td>33,589</td>
<td>133,557</td>
<td>mem</td>
</tr>
<tr>
<td>t4-4</td>
<td>304</td>
<td>468</td>
<td>63</td>
<td>(0.82%)</td>
<td>—</td>
<td>43,349</td>
<td>275,168</td>
<td>mem</td>
</tr>
</tbody>
</table>
TABLE 3. Results for Asym-G instances: Compact linear MIP formulation solved with Cplex.

| Name     | |V|  | |E|  | |Q|  | tcf  |
|----------|-----------------|---------|----------|---------|----------|-----------------|
| arpanet  | 24  | 50  | 10  | 6.34 |
| at-cp    | 15  | 22  | 6   | 0.05 |
| bhv3     | 29  | 62  | 15  | 7.60 |
| bhv6     | 27  | 39  | 15  | 0.29 |
| bhv13    | 29  | 36  | 13  | 1.37 |
| cost239  | 11  | 22  | 5   | 0.04 |
| eon      | 19  | 37  | 15  | 1.44 |
| latadl   | 39  | 86  | 17  | 471.06 |
| metro    | 11  | 42  | 5   | 0.40 |
| njlata   | 11  | 23  | 8   | 1.02 |
| nor-sun  | 27  | 51  | 13  | 38.00 |
| nsf      | 14  | 21  | 10  | 1.62 |
| ny-cp    | 16  | 49  | 9   | 1.37 |
| pachell  | 15  | 21  | 7   | 0.09 |
| toronto  | 25  | 55  | 11  | 29.14 |
| usld     | 28  | 45  | 15  | 29.02 |
| res1     | 45  | 63  | 15  | 37.53 |
| res5     | 44  | 67  | 18  | 64.14 |
| res6     | 44  | 60  | 14  | 37.87 |
| res7     | 44  | 62  | 18  | 39.79 |
| res8     | 50  | 79  | 19  | 215.50 |
| res9     | 50  | 76  | 23  | 138.01 |
| stein1   | 50  | 63  | 20  | 1.91 |
| stein2   | 50  | 100 | 17  | 96.40 |
| stein3   | 75  | 94  | 32  | 113.49 |
| stein4   | 75  | 150 | 33  | —    |
| n45      | 45  | 63  | 20  | 102.47 |
| n49      | 49  | 57  | 14  | 1.13 |
| n147     | 147 | 265 | 37  | 48.99 |
| t3-4     | 250 | 465 | 39  | 720.06 |

variables. However, we should point out that passing from the symmetric to the asymmetric Hose models renders the problem significantly harder. This is indeed the case for Sym-T and Asym-T. Although Sym-T can be solved by repeated application of shortest path algorithms, Asym-T is strongly NP-hard. Concerning the behavior of the branch-and-price algorithm, we should notice that the number of $\omega$ variables in the Asym-G MIP formulation (18)–(23) is twice the number of $\omega$ variables in the Sym-G formulation. We believe that this increase in the number of variables negatively affects the pricing and separation phases of the BPC algorithm. We have observed this in a few numerical examples that we did not report in the tables because the running times were worse than with the compact formulation. Let us consider the three examples res1, stein1, and n147. For the Sym-G case from Table 2, we see that res1 was solved to optimality by generating 2843 paths and 3533 cuts. The same statistics when we pass to Asym-G uncertainty problems are 4972 and 5952, respectively. For stein1, in the Sym-G case the number of paths and cuts were 2778 and 2657, respectively. Passing to Asym-G, these numbers become 5628 and 17225, respectively. Finally, in the n147 case, we have 7668 paths and 8211 cuts in the Sym-G case and 9823 paths and 35629 cuts for Asym-G. This seems to be a pattern for almost all test problems.

The results obtained for Rob-G are summarized in Tables 4 and 5. Here, $\Gamma$ is taken equal to 15% of the total number of terminal pairs, but other intermediate values yield similar results. Although the robust VPN provisioning problem leads to less conservative VPNs by exploiting the available traffic statistics, Tables 4 and 5 indicate that the corresponding compact linear MIP formulation is much harder to tackle with Cplex 8.1 than those for Asym-G and Sym-G. However, the BPC algorithm is remarkably effective in solving Rob-G instances, and it compares very favorably with Cplex 8.1 even for small-size instances. As happens with Sym-G, the compact flow-based formulation for Rob-G requires more memory than the BPC algorithm in which paths and cuts are dynamically generated starting from small initial sets. For all instances larger than “n147,” the compact formulations do not fit within 2 GB of RAM, whereas the path-and-cut formulations, which have a potentially exponential size, lead to optimal solutions in less than 2 hours. Note that only a limited number of paths and cuts are generated.

The improved performance of the BPC for this uncertainty model cannot be related to a lower dimensional demand polyhedron as might be between Asym-G and Sym-G (see discussion above), because there is a large increase in the number of variables in Rob-G with respect to Asym-G. Therefore, we cannot say that it is just the number of variables that affects the performance of the BPC algorithm. Our partial explanation for this is the favorable structure of the polyhedron that results from the Bertsimas and Sim constraints. More specifically, although $2|S| + 1 = |Q|(|Q| - 1) + 1$ constraints describe the latter and only $2|Q|$ are needed for Asym-G ($Q$ being the set of terminal
nodes and $S$ the set of pairs of terminals), most of the Rob-$G$
constraints are upper and lower bounding constraints, perhaps simpler to handle by the LP solver.

5. CONCLUDING REMARKS

We have addressed the network design problem under traffic uncertainty arising when provisioning Virtual Private Networks. Sufficient capacity must be reserved on the links of a large underlying public network to support all valid traffic matrices between a given set of terminals while minimizing total reservation costs.

The first contribution of this article consists of new compact linear MIP formulations that allow us to solve to optimality medium-to-large-size instances of Asym-$G$ and Sym-$G$ in less than 15 minutes of computing time. These flow-based formulations also provide significant experimental support to the conjecture about Sym-$G$ and Sym-$T$.

To exploit traffic statistics that are available for provisioned VPNs, we have investigated a robust VPN provi-

| Name | $|V|$ | $|E|$ | $|Q|$ | $t_{root}$ | $t_{tot}$ | #paths | #cuts | $t_{cf}$ (gap) |
|------|-----|-----|-----|-------|-------|-----|-----|-------------|
| arpanet | 24 | 50 | 10 | 0.14 | 0.21 | 463 | 178 | (6.88%) |
| at-cep | 15 | 22 | 6 | 0.03 | 0.04 | 172 | 79 | 0.06 |
| bhv3 | 29 | 62 | 15 | 0.25 | 0.35 | 1059 | 411 | 158.50 |
| bhv6 | 27 | 39 | 15 | 0.33 | 0.47 | 1139 | 414 | 3.52 |
| bhv13 | 29 | 36 | 13 | 0.21 | 0.30 | 787 | 306 | 0.82 |
| cost239 | 11 | 22 | 5 | 0.02 | 0.03 | 108 | 39 | 0.04 |
| eon | 19 | 37 | 15 | 0.44 | 0.57 | 1120 | 597 | 3337.24 |
| latadl | 39 | 86 | 17 | 1.74 | 2.14 | 1532 | 1028 | (9.24%) |
| metro | 11 | 42 | 5 | 0.14 | 0.15 | 278 | 152 | 0.45 |
| njlata | 11 | 23 | 8 | 0.05 | 0.08 | 292 | 156 | 56.72 |
| nor-sun | 27 | 51 | 13 | 0.62 | 0.74 | 974 | 715 | 686.27 |
| nsf | 14 | 21 | 10 | 0.06 | 0.09 | 479 | 172 | 7.45 |
| ny-cep | 16 | 49 | 9 | 0.24 | 0.29 | 453 | 290 | 7.15 |
| pacbell | 15 | 21 | 7 | 0.03 | 0.05 | 222 | 82 | 1.16 |
| toronto | 25 | 55 | 11 | 0.21 | 0.31 | 564 | 219 | (6.43%) |
| usld | 28 | 45 | 15 | 0.39 | 0.54 | 1076 | 413 | (2.75%) |

| Name | $|V|$ | $|E|$ | $|Q|$ | $t_{root}$ | $t_{tot}$ | #paths | #cuts | $t_{cf}$ (gap) |
|------|-----|-----|-----|-------|-------|-----|-----|-------------|
| res1 | 45 | 63 | 15 | 0.91 | 1.12 | 1240 | 810 | (1.53%) |
| res5 | 44 | 67 | 18 | 1.26 | 1.59 | 1615 | 895 | 1249.61 |
| res6 | 44 | 60 | 14 | 0.56 | 0.75 | 1001 | 533 | (3.08%) |
| res7 | 44 | 62 | 18 | 1.46 | 1.77 | 1619 | 1177 | (1.93%) |
| res8 | 50 | 79 | 19 | 2.68 | 3.11 | 1864 | 1642 | (11.00%) |
| res9 | 50 | 76 | 23 | 2.08 | 2.70 | 2613 | 997 | (0.04%) |
| stein1 | 50 | 63 | 20 | 1.56 | 1.94 | 1972 | 1114 | 29.94 |
| stein2 | 50 | 100 | 17 | 2.45 | 2.90 | 1801 | 1282 | (0.65%) |
| stein3 | 75 | 94 | 32 | 12.75 | 14.22 | 5555 | 3878 | (1.03%) |
| stein4 | 75 | 150 | 33 | 33.03 | 35.49 | 6090 | 6114 | — |
| n45 | 45 | 63 | 20 | 1.59 | 1.98 | 2072 | 1112 | (8.85%) |
| n49 | 45 | 57 | 14 | 0.87 | 1.04 | 1363 | 8560 | 7.21 |
| n147 | 147 | 265 | 37 | 235.09 | 240.75 | 13,640 | 19,114 | 442.36 |
| g200 | 200 | 914 | 31 | 2668.21 | 2679.64 | 37,116 | 50,180 | mem |
| t3-0 | 250 | 444 | 52 | 441.09 | 446.07 | 109,092 | 23,544 | mem |
| t3-1 | 250 | 456 | 53 | 676.98 | 681.31 | 21,100 | 43,133 | mem |
| t3-2 | 250 | 469 | 42 | 649.31 | 654.12 | 13,977 | 28,549 | mem |
| t3-3 | 250 | 446 | 50 | 562.91 | 564.54 | 16,687 | 16,974 | mem |
| t3-4 | 250 | 465 | 39 | 236.97 | 247.25 | 18,353 | 23,489 | mem |
| t4-0 | 304 | 453 | 55 | 2782.64 | 2822.95 | 18,353 | 23,489 | mem |
| t4-1 | 304 | 457 | 63 | 3578.18 | 3606.15 | 23,796 | 23,250 | mem |
| t4-2 | 304 | 454 | 58 | 3393.84 | 3437.39 | 19,986 | 26,110 | mem |
| t4-3 | 304 | 458 | 67 | 6824.91 | 6855.39 | 29,118 | 39,378 | mem |
| t4-4 | 304 | 468 | 63 | 5492.71 | 5520.48 | 24,549 | 34,754 | mem |
sioning problem, Rob-G, which leads to less conservative reservations. The compact linear MIP formulation for this NP-hard problem is more challenging to tackle with a commercial solver than the one for Asym-G.

The combined branch-and-price and cutting-plane algorithm we have devised, based on path and cut formulations, performs well on large Sym-G instances and remarkably well on all Rob-G instances. Further work is needed to establish whether it can also compare favourably with the compact formulations for large Asym-G instances.

In [1] we develop a more efficient branch-and-price approach that limits the memory requirements, due to the generation of paths and to unused \(\omega\) variables, and that involves a more balanced branching rule. The compact linear MIP formulations and algorithm are also extended to the integer multicommodity network design problem under demand uncertainty, where the demand matrix is assumed to lie in a union of polytopes.

Acknowledgments

The authors thank Francesco Maffioli for helpful discussions.

REFERENCES


