Provisioning Virtual Private Networks under Traffic Uncertainty

A. Altın
Department of Industrial Engineering, Bilkent University, Ankara, Turkey

E. Amaldi
Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy

P. Belotti
Dipartimento di Elettronica e Informazione, Politecnico di Milano, Italy

M. Ç. Pınar
Department of Industrial Engineering, Bilkent University, Ankara, Turkey

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.
NETWORKS, Vol. 49(1), 100–115 2007

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

Received September 2004; accepted May 2006
Correspondence to: E. Amaldi; e-mail: amaldi@elet.polimi.it
Contract grant sponsor: TUBITAK, The Scientific and Technological Research Institute of Turkey; contract grant number: MISAG-CNR-1
Contract grant sponsor: CNR, Consiglio Nazionale delle Ricerche, Italy; contract grant number: MISAG-CNR-1
DOI 10.1002/net.20145
Published online in Wiley InterScience (www.interscience.wiley.com).

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 $d_{ss} = 0$ 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_{AsymG} defined by the following inequalities:

$$\sum_{t:t\neq s} d_{st} \leq b_s^+, \ \sum_{t:t\neq s} d_{ts} \leq b_s^- \quad \forall\, s \in Q \tag{1}$$

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

Throughout this article we make the realistic assumption that traffic is *unsplittable,* 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:

$$\sum_{t:(s,t)\in S} d_{st} + \sum_{t:(t,s)\in S} d_{ts} \leq b_s \quad \forall\, s \in Q \tag{3}$$

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

with upper bounds on the cumulative entering and outgoing traffic. Figure 1 illustrates an instance of *Sym-G,* the corresponding optimal solution, and the routing of two traffic matrices within U_{SymG} with the related routing.

To the best of our knowledge, the state of the art on the computational complexity of these variants can be summarized as follows:

- *Sym-T* can be solved in polynomial time by shortest path computations [10];
- *Asym-T* is strongly NP-hard, and does not admit a polynomial time approximation scheme, unless P = NP [10], and the best known approximation algorithm is guaranteed to yield a VPN whose total reservation cost is at most 9.002 times larger than that of a minimum cost VPN [17];
- Any optimal solution for *Sym-T* provides a 2-approximation to *Sym-G* [10] but it is still open whether *Sym-G* is actually NP-hard;
- The best approximation algorithm for *Asym-G* has a 5.55 factor [11] but it is still open whether *Asym-G* is actually NP-hard.

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,

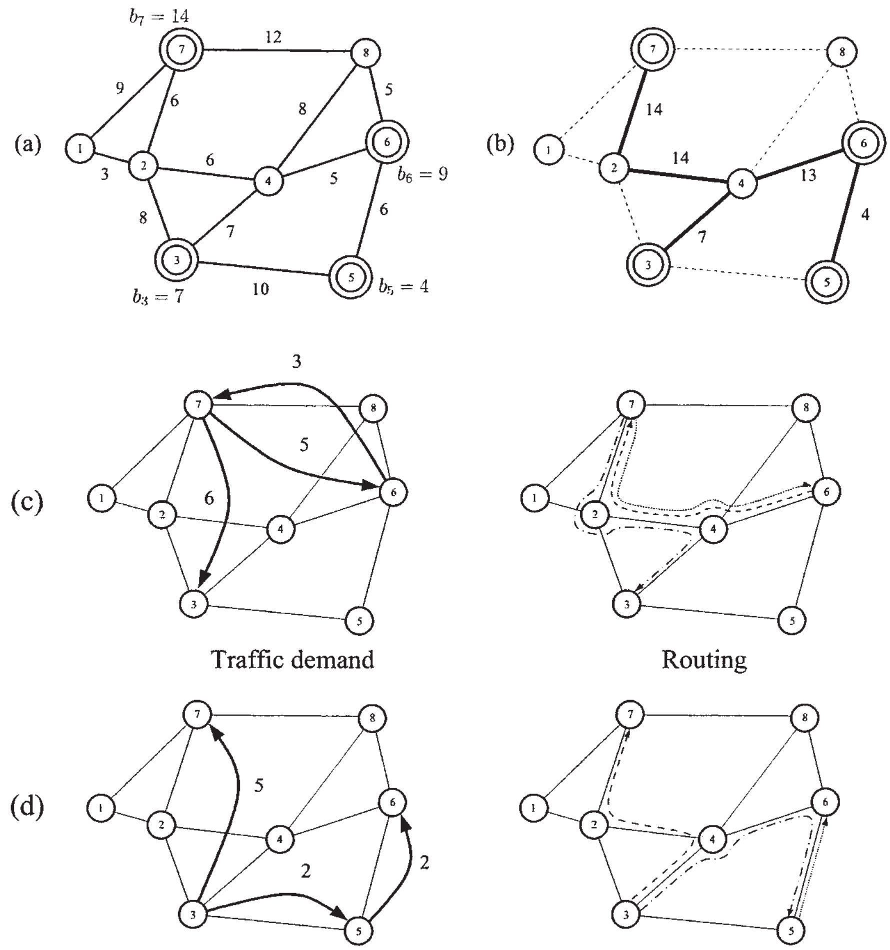

FIG. 1. An instance of *Sym-G*. The network topology is given in (a) with edge capacity costs, terminal nodes 3, 5, 6, and 7 are emphasized and the upper bounds b_s are indicated. An optimal solution is shown in (b) with edge capacities. Two sets of traffic demands within U_{SymG} are given in (c) and (d), with the related routing. For example, in (c), the traffic demand is such that $d_{76} = 5$, $d_{67} = 3$, $d_{73} = 6$, and the remaining d_{st}s are equal to zero.

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 *unsplittable,* 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 $D = (d_{st})_{s,t \in Q}$ 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:

$$\min \sum_{\{i,j\} \in E} c_{ij} x_{ij} \tag{5}$$

$$\text{s.t.} \sum_{j:\{i,j\} \in E} (y_{ij}^{st} - y_{ji}^{st}) = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \in V,\ \forall\, (s,t) \in S \tag{6}$$

$$\sum_{(s,t) \in S} d_{st}(y_{ij}^{st} + y_{ji}^{st}) \le x_{ij} \quad \forall\, \{i, j\} \in E \tag{7}$$

$$y_{ij}^{st} \in \{0, 1\} \quad \forall\, (i, j) \in A,\ \forall\, (s, t) \in S \tag{8}$$

$$x_{ij} \ge 0 \quad \forall\, \{i, j\} \in E, \tag{9}$$

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:

$$\min \sum_{\{i,j\} \in E} c_{ij} \sum_{(s,t) \in S} d_{st}(y_{ij}^{st} + y_{ji}^{st}) \tag{10}$$

$$\text{s.t.} \sum_{j:\{i,j\} \in E} (y_{ij}^{st} - y_{ji}^{st}) = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \in V,\ \forall\, (s,t) \in S \tag{11}$$

$$y_{ij}^{st} \in \{0, 1\} \quad \forall\, (i, j) \in A,\ \forall\, (s, t) \in S \tag{12}$$

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

$$\min \sum_{\{i,j\} \in E} c_{ij} x_{ij} \tag{13}$$

$$\text{s.t.} \sum_{j:\{i,j\} \in E} (y_{ij}^{st} - y_{ji}^{st}) = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \in V,\ \forall\, (s,t) \in S \tag{14}$$

$$\sum_{(s,t) \in S} d_{st}(y_{ij}^{st} + y_{ji}^{st}) \le x_{ij} \quad \forall\, D \in U_{\text{AsymG}}(U_{\text{SymG}}),\ \forall\, \{i, j\} \in E \tag{15}$$

$$y_{ij}^{st} \in \{0, 1\} \quad \forall\, (i, j) \in A,\ \forall\, (s, t) \in S \tag{16}$$

$$x_{ij} \ge 0 \quad \forall\, \{i, j\} \in E. \tag{17}$$

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 $\tilde{y}_{ij}^{st}$ and the capacity variables $\tilde{x}_{ij}$. Because the resulting routing vector $\tilde{\mathbf{y}}$ 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 $\tilde{x}$s are considered as coefficients and the edge overload $\sum_{(s,t) \in S} d_{st}(\tilde{y}_{ij}^{st} + \tilde{y}_{ji}^{st}) - \tilde{x}_{ij}$ is maximized over this polyhedron. If there exists a valid traffic matrix $(\check{d}_{st})_{s,t \in Q}$ that is not supported by the capacity $\tilde{x}_{ij}$ reserved on edge $\{i, j\}$, that is, if such overload is positive, a new constraint (7) with coefficients $\check{d}_{st}$ 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 special 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*:

$$\min \sum_{\{i,j\}\in E} c_{ij}x_{ij} \tag{18}$$

$$\text{s.t.} \sum_{j:\{i,j\}\in E} (y_{ij}^{st} - y_{ji}^{st}) = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \in V,\ \forall\,(s,t)\in S \tag{19}$$

$$\sum_{s\in Q} (b_s^+\omega_{ij}^{s+} + b_s^-\omega_{ij}^{s-}) \le x_{ij} \quad \forall\,\{i,j\}\in E \tag{20}$$

$$\omega_{ij}^{s+} + \omega_{ij}^{t-} \ge (y_{ij}^{st} + y_{ji}^{st}) \quad \forall\,\{i,j\}\in E,\ \forall\,(s,t)\in S \tag{21}$$

$$y_{ij}^{st} \in \{0, 1\} \quad \forall\,(i,j)\in A,\ \forall\,(s,t)\in S \tag{22}$$

$$\omega_{ij}^{s+},\ \omega_{ij}^{s-},\ x_{ij} \ge 0 \quad \forall\,\{i,j\}\in E,\ \forall\, s\in Q. \tag{23}$$

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:

$$x_{ij} \ge \max \sum_{(s,t)\in S} d_{st}(y_{ij}^{st} + y_{ji}^{st}) \tag{24}$$

$$(\omega_{ij}^{s+}) \quad \sum_{t\neq s} d_{st} \le b_s^+ \quad \forall\, s\in Q \tag{25}$$

$$(\omega_{ij}^{s-}) \quad \sum_{t\neq s} d_{ts} \le b_s^- \quad \forall\, s\in Q \tag{26}$$

$$d_{st} \ge 0 \quad \forall\,(s,t)\in S, \tag{27}$$

where the ωs 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:

$$x_{ij} \ge \min \sum_{s\in Q} (b_s^+\omega_{ij}^{s+} + b_s^-\omega_{ij}^{s-}) \tag{28}$$

$$\omega_{ij}^{s+} + \omega_{ij}^{t-} \ge y_{ij}^{st} + y_{ji}^{st} \quad \forall\,(s,t)\in S \tag{29}$$

$$\omega_{ij}^{s+},\ \omega_{ij}^{s-} \ge 0 \quad \forall\, s\in Q. \tag{30}$$

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 the 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

$$\sum_{s\in Q} b_s\omega_{ij}^{s} \le x_{ij} \quad \forall\,\{i,j\}\in E \tag{31}$$

$$\omega_{ij}^{s} + \omega_{ij}^{t} \ge y_{ij}^{st} + y_{ji}^{st} \quad \forall\,\{i,j\}\in E,\ \forall\,(s,t)\in S \tag{32}$$

$$\omega_{ij}^{s},\ x_{ij} \ge 0 \quad \forall\,\{i,j\}\in E,\ \forall\, s\in Q. \tag{33}$$

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 $y^{st}(F') = \sum_{(i,j)\in F'} y_{ij}^{st}$. For any subset of nodes $W \subset V$, 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 $W \subset V$, the *directed cut* $\delta'(W)$ is the set of arcs with the tail in W and the head in

$V \backslash W$.

Given any subset $W \subset V$, if there exists a nonempty set of terminal pairs $S' = \{(s, t) \in S : s \in W, t \in V \backslash W\}$, 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,

$$y^{st}(\delta'(W)) \geq 1. \tag{34}$$

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

$$\sum_{(i,j)\in\delta'(W)} (\omega_{ij}^{s+} + \omega_{ij}^{t-}) \geq 1 \quad \forall\,(s, t) \in S : |\{s, t\} \cap W| = 1 \tag{35}$$

that only involve ω variables and that are satisfied by any feasible solution of *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:

$$x(\delta(W)) \geq \sum_{(s,t)\in S:|\{s,t\}\cap W|=1} d_{st}. \tag{36}$$

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

$$\max\left\{ \sum_{(s,t)\in S:|\{s,t\}\cap W|=1} d_{st} : \sum_{t:t\neq s} d_{ts} \leq b_s^- \ \ \forall\, s \in Q, \sum_{t:t\neq s} d_{st} \leq b_s^+ \ \ \forall\, s \in Q, d_{st} \geq 0 \ \ \forall\,(s, t) \in S \right\},$$

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

$$\sum_{\{i,j\}\in\delta(W)} \sum_{s\in Q} (b_s^+\omega_{ij}^{s+} + b_s^-\omega_{ij}^{t-}) \geq d[\delta(W)]. \tag{37}$$

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 ω variables and a subset of these inequalities provides a lower bound on *Asym-G*. As for (37), we may get rid of the x variables and express the objective function in terms of the ω variables only:

$$\sum_{\{i,j\}\in E} c_{ij} \sum_{(s,t)\in S} (b_s^+\omega_{ij}^{s+} + b_s^-\omega_{ij}^{t-}). \tag{38}$$

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 *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_s^+, b_s^-), $\forall s \in Q$. Let Δ 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:

$$(P_{\text{dual}}) \min \sum_{\{i,j\}\in E} c_{ij} \sum_{(s,t)\in S} (b_s^+\omega_{ij}^{s+} + b_s^-\omega_{ij}^{t-})$$

$$\text{s.t.} \sum_{(i,j)\in\delta'(W)} (\omega_{ij}^{s+} + \omega_{ij}^{t-}) \geq 1 \quad \forall\,(s, t, W) \in \Delta.$$

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 *Sym-G*. From (32) and (34) we obtain the following two families of valid inequalities:

$$\sum_{(i,j)\in\delta'(W)} (\omega_{ij}^{s} + \omega_{ij}^{t}) \geq 1 \quad \forall\, W \subset V,\ \forall\,(s, t) \in S : |\{s, t\} \cap W| = 1$$

$$\sum_{\{i,j\}\in\delta(W)} \sum_{s\in Q} b_s\omega_{ij}^{s} \geq d(\delta(W)) \quad \forall\, W \subset V.$$

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 $[d'_{st} - \hat{d}_{st}, d'_{st} + \hat{d}_{st}]$, where $\hat{d}_{st} \in [0, d'_{st}]$ 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 corresponding intervals and concerning the way different demands d_{st} may be interrelated.

For nonnegative deviations $\hat{d}_{st}$, 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:

$$\min \sum_{\{i,j\}\in E} c_{ij} \sum_{(s,t)\in S} (d'_{st} + \hat{d}_{st})(y^{st}_{ij} + y^{st}_{ji}) \tag{39}$$

$$\text{s.t.} \sum_{j:\{i,j\}\in E} (y^{st}_{ij} - y^{st}_{ji}) = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \in V,\ \forall\, (s,t) \in S \tag{40}$$

$$y^{st}_{ij} \in \{0, 1\} \quad \forall\, (i,j) \in A,\ \forall\, (s,t) \in S. \tag{41}$$

As for the nominal problem (10)–(12) corresponding to the nominal traffic matrix $D' = (d'_{st})_{s,t\in Q}$, 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 Γ, $1 \le \Gamma \le |S|$, 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 Γ of the demand values simultaneously take their worst-case values. In the VPN provisioning setting, the parameter Γ 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 Γ of the demands. More precisely, we consider the following robust VPN provisioning problem:

$$\min \sum_{\{i,j\}\in E} c_{ij} x_{ij} \tag{42}$$

$$\text{s.t.} \sum_{j:\{i,j\}\in E} (y^{st}_{ij} - y^{st}_{ji}) = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \in V,\ \forall\, (s,t) \in S \tag{43}$$

$$\sum_{(s,t)\in S} d'_{st}(y^{st}_{ij} + y^{st}_{ji}) + \max_{\{\tilde{S}\,:\,\tilde{S}\subseteq S, |\tilde{S}|\le\Gamma\}} \sum_{(s,t)\in\tilde{S}} \hat{d}_{st}(y^{st}_{ij} + y^{st}_{ji}) \le x_{ij} \quad \forall\, \{i,j\} \in E \tag{44}$$

$$y^{st}_{ij} \in \{0, 1\} \quad \forall\, (i,j) \in A,\ \forall\, (s,t) \in S \tag{45}$$

$$x_{ij} \ge 0 \quad \forall\, \{i,j\} \in E, \tag{46}$$

where $\tilde{S}$ denotes a subset of S of cardinality at most Γ. We refer to this robust variant of the problem as *Rob-G*. It is worth pointing out that the general probabilistic guarantees derived for the extreme behavior of more than Γ parameters in [5] are also valid in our case. We also note that choosing Γ 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 *Sym-G* and *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*:

$$\min \sum_{\{i,j\}\in E} c_{ij} x_{ij} \tag{47}$$

$$\text{s.t.} \sum_{j:\{i,j\}\in E} (y^{st}_{ij} - y^{st}_{ji}) = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall\, i \in V,\ \forall\, (s,t) \in S \tag{48}$$

$$\sum_{(s,t)\in S} d'_{st}(y^{st}_{ij} + y^{st}_{ij}) + \Gamma\pi^0_{ij} + \sum_{(s,t)\in S} \pi^{st}_{ij} \le x_{ij} \quad \forall\, \{i,j\} \in E \tag{49}$$

$$\pi^0_{ij} + \pi^{st}_{ij} \ge \hat{d}_{st}(y^{st}_{ij} + y^{st}_{ji}) \quad \forall\, \{i,j\} \in E,\ \forall\, (s,t) \in S \tag{50}$$

$$y^{st}_{ij} \in \{0, 1\} \quad \forall\, (i,j) \in A,\ \forall\, (s,t) \in S \tag{51}$$

$$\pi^0_{ij},\ \pi^{st}_{ij},\ x_{ij} \ge 0 \quad \forall\, \{i,j\} \in E,\ \forall\, (s,t) \in S. \tag{52}$$

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

$$\max_{\{\tilde{S}\,:\,\tilde{S}\subseteq S, |\tilde{S}|\le\Gamma\}} \sum_{(s,t)\in\tilde{S}} \hat{d}_{st}(\bar{y}^{st}_{ij} + \bar{y}^{st}_{ji}) \tag{53}$$

is equivalent to

$$\max \sum_{(s,t)\in S} \alpha_{st}\hat{d}_{st}(\bar{y}_{ij}^{st} + \bar{y}_{ji}^{st}) \tag{54}$$

$$\text{s.t.} \sum_{(s,t)\in S} \alpha_{st} \leq \Gamma \tag{55}$$

$$\alpha_{st} \in \{0, 1\} \quad \forall (s, t) \in S. \tag{56}$$

The key observation here is that the above problem, viewed as a linear program by relaxing the integrality of α_{st} variables, always has a binary optimal solution as it is a simple knapsack. Therefore, it can be replaced with

$$\max \sum_{(s,t)\in S} \alpha_{st}\hat{d}_{st}(\bar{y}_{ij}^{st} + \bar{y}_{ji}^{st}) \tag{57}$$

$$\text{s.t.} \sum_{(s,t)\in S} \alpha_{st} \leq \Gamma \tag{58}$$

$$0 \leq \alpha_{st} \leq 1 \quad \forall (s, t) \in S, \tag{59}$$

which admits the following dual:

$$\min \Gamma\pi_{ij}^0 + \sum_{(s,t)\in S} \pi_{ij}^{st} \tag{60}$$

$$\text{s.t. } \pi_{ij}^0 + \pi_{ij}^{st} \geq \hat{d}_{st}(\bar{y}_{ij}^{st} + \bar{y}_{ji}^{st}) \quad \forall \{i, j\} \in E,\ \forall (s, t) \in S \tag{61}$$

$$\pi_{ij}^0,\ \pi_{ij}^{st} \geq 0 \quad \forall \{i, j\} \in E,\ \forall (s, t) \in S. \tag{62}$$

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 $\hat{d}_{st} = 0$ 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 and 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_i, we consider two other terminals v_i and w_i in Q connected by three separate parallel paths that correspond to the three literals in the clause. Each one of these (v_i, w_i)-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, w_i), $1 \leq i \leq m$, must intersect in the appropriate way. For example, if the 3-SAT instance contains a first clause C_{i_1} with the literals x_{j_1}, $\bar{x}_{j_2}$ and x_{j_3}, and a second clause C_{i_2} with the literals $\bar{x}_{j_1}$, x_{j_2} and x_{j_3}, 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}, w_{i_1})-path shares exactly one edge (the second one) with the path connecting terminals s_{j_1} and t_{j_1} that corresponds to variable x_{j_1}; the second (v_{i_1}, w_{i_1})-path shares exactly one edge (the second one) with the path connecting s_{j_2} and t_{j_2} that corresponds to $\bar{x}_{j_2}$; the third (v_{i_1}, w_{i_1})-path shares exactly one edge (the second one) with the path connecting s_{j_3} and t_{j_3} that corresponds to x_{j_3}. The three (v_{i_2}, w_{i_2})-paths corresponding to the three literals of the second clause are constructed accordingly. Note that, because the variable x_{j_3} occurs in both clauses, the (s_{j_3}, t_{j_3})-path associated with x_{j_3} (the first one) shares its second edge with the third (v_{i_1}, w_{i_1})-path and its fourth edge with the third (v_{i_2}, w_{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, w_i)-path corresponding to a literal of a clause. Note that all paths in G, which connect any (s_j, t_j) (respectively (v_i, w_i)) terminal pair but are not (s_j, t_j)-paths (respectively (v_i, w_i)-paths), contain more than 11 (respectively 3) edges.

The special instance of the robust VPN provisioning

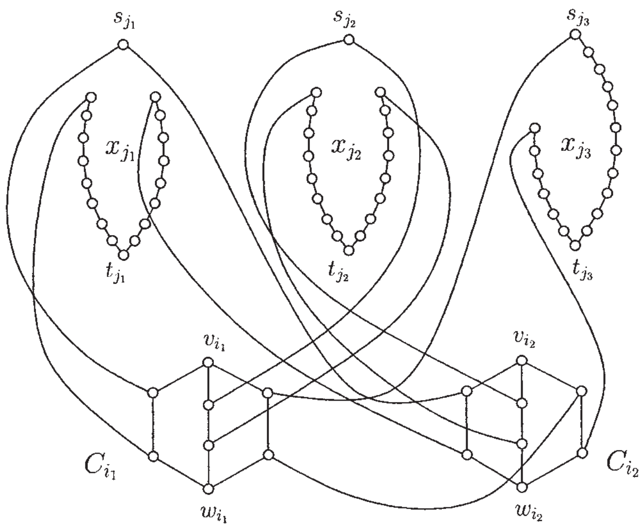

FIG. 2. Graph of the special instance of the VPN provisioning problem associated with a 3-SAT instance with the clauses $C_{i_1} = (x_{j_1} \vee \bar{x}_{j_2} \vee x_{j_3})$ and $C_{i_2} = (\bar{x}_{j_1} \vee x_{j_2} \vee x_{j_3})$.

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 $d'_{s_j t_j} = 0$ for all j and $d'_{v_i w_i} = 0$ for all i, and the deviations $\hat{d}_{s_j t_j} = 1$ for all j and $\hat{d}_{v_i w_i} = 1$ 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_i, 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 $11n + 2m$ amounts to 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_i (i.e., to select one of the three alternative 3-edge (v_i, w_i)-paths) for every i, $1 \leq i \leq m$, so that for each i the selected (v_i, w_i)-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_i)] and each selected (v_i, w_i)-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_i) 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_i)-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_i) terminal pairs. If we consider instances with $d'_{s_j t_j} = 1$ and $d'_{v_i w_i} = 1$ 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 $\hat{d}_{st}$ multiplies several binary variables y^{st}_{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):

$$\max \sum_{(s,t)\in S} d_{st}(\bar{y}^{st}_{ij} + \bar{y}^{st}_{ji})$$

$$\text{s.t.}$$

$$(\pi^0_{ij}) \quad \sum_{(s,t)\in S} \frac{d_{st} - d'_{st}}{\hat{d}_{st}} \leq \Gamma$$

$$(\sigma^{st}_{ij}) \quad -d_{st} \leq -d'_{st} \quad \forall\, (s, t) \in S$$

$$(\rho^{st}_{ij}) \quad d_{st} \leq d'_{st} + \hat{d}_{st} \quad \forall\, (s, t) \in S,$$

where π^0_{ij}, σ^{st}_{ij} and ρ^{st}_{ij} are the corresponding dual variables. It is easy to show that constraints (49) and (50) can be replaced by

$$(\Gamma + \sum_{(s,t)\in S} d'_{st}/\hat{d}_{st})\pi^0_{ij} + \sum_{(s,t)\in S} ((d'_{st} + \hat{d}_{st})\rho^{st}_{ij} - d'_{st}\sigma^{st}_{ij}) \leq x_{ij} \quad \forall\, \{i, j\} \in E$$

$$\pi^0_{ij}/\hat{d}_{st} - \sigma^{st}_{ij} + \rho^{st}_{ij} \geq y^{st}_{ij} + y^{st}_{ji} \quad \forall\, \{i, j\} \in E,\ \forall\, (s, t) \in S$$

$$\pi^0_{ij},\ \sigma^{st}_{ij},\ \rho^{st}_{ij} \geq 0 \quad \forall\, \{i, j\} \in E,\ \forall\, (s, t) \in S$$

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

$$\sum_{(i,j)\in\delta^+(W)} (\pi^0_{ij}/\hat{d}_{st} - \sigma^{st}_{ij} + \rho^{st}_{ij}) \geq 1 \quad \forall\, W \subset V,\ \forall\, (s, t) \in S : |\{s, t\} \cap W| = 1$$

$$\sum_{\{i,j\}\in\delta(W)} ((\Gamma + \sum_{(s,t)\in S} d'_{st}/\hat{d}_{st})\pi^0_{ij} + \sum_{(s,t)\in S} ((d'_{st} + \hat{d}_{st})\rho^{st}_{ij} - d'_{st}\sigma^{st}_{ij})) \geq d(\delta(W)) \quad \forall\, W \subset V.$$

2.3. *Path Formulations*

The flow-based models of the preceding sections quickly become too large for medium-to-large network design instances. 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_p 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:

$$\min \sum_{\{i,j\}\in E} c_{ij}x_{ij} \tag{63}$$

$$\text{s.t.} \sum_{p\in P^{st}} z_p \geq 1 \quad \forall\,(s,t)\in S \tag{64}$$

$$\sum_{(s,t)\in S} d_{st} \sum_{p\in P^{st}\cap P_{ij}} z_p \leq x_{ij} \quad \forall\, D \in U_{\text{AsymG}},\ \forall\,\{i,j\}\in E \tag{65}$$

$$z_p \in \{0,1\} \quad \forall\, p\in P \tag{66}$$

$$x_{ij} \geq 0 \quad \forall\,\{i,j\}\in E. \tag{67}$$

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:*

$$\min \sum_{\{i,j\}\in E} c_{ij}x_{ij} \tag{68}$$

$$\text{s.t.} \sum_{p\in P^{st}} z_p \geq 1 \quad \forall\,(s,t)\in S \tag{69}$$

$$\sum_{s\in Q} (b_s^+\omega_{ij}^{s+} + b_s^-\omega_{ij}^{s-}) \leq x_{ij} \quad \forall\,\{i,j\}\in E \tag{70}$$

$$\omega_{ij}^{s+} + \omega_{ij}^{t-} \geq \sum_{p\in P^{st}\cap P_{ij}} z_p \quad \forall\,(s,t)\in S,\ \forall\,\{i,j\}\in E \tag{71}$$

$$z_p \in \{0,1\} \quad \forall\,(s,t)\in S,\ \forall\, p\in P^{st} \tag{72}$$

$$\omega_{ij}^{s+},\ \omega_{ij}^{s-},\ x_{ij} \geq 0 \quad \forall\,\{i,j\}\in E,\ \forall\, s\in Q. \tag{73}$$

For *Sym-G* we have the following path-based robust formulation:

$$\min \sum_{\{i,j\}\in E} c_{ij}x_{ij}$$

$$\text{s.t.} \sum_{p\in P^{st}} z_p \geq 1 \quad \forall\,(s,t)\in S$$

$$\sum_{s\in Q} b_s\omega_{ij}^s \leq x_{ij} \quad \forall\,\{i,j\}\in E$$

$$\omega_{ij}^s + \omega_{ij}^t \geq \sum_{p\in P^{st}\cap P_{ij}} z_p \quad \forall\,(s,t)\in S,\ \forall\,\{i,j\}\in E$$

$$z_p \in \{0,1\} \quad \forall\,(s,t)\in S,\ \forall\, p\in P^{st}$$

$$\omega_{ij}^s,\ x_{ij} \geq 0 \quad \forall\,\{i,j\}\in E,\ \forall\, s\in Q.$$

A similar linear path-based formulation is obtained for *Rob-G:*

$$\min \sum_{\{i,j\}\in E} c_{ij}x_{ij}$$

$$\text{s.t.} \sum_{p\in P^{st}} z_p \geq 1 \quad \forall\,(s,t)\in S$$

$$\sum_{(s,t)\in S} d'_{s,t} \sum_{p\in P^{st}\cap P_{ij}} z_p + \Gamma\pi_{ij}^0 + \sum_{(s,t)\in S} \pi_{ij}^{st} \leq x_{ij} \quad \forall\,\{i,j\}\in E$$

$$\pi_{ij}^{st} \geq \hat{d}_{st} \sum_{p\in P^{st}\cap P_{ij}} z_p - \pi_{ij}^0 \quad \forall\,\{i,j\}\in E,\ \forall\,(s,t)\in S$$

$$z_p \in \{0,1\} \quad \forall\,(s,t)\in S,\ \forall\, p\in P^{st}$$

$$\pi_{ij}^0,\ \pi_{ij}^{st},\ x_{ij} \geq 0 \quad \forall\,\{i,j\}\in E,\ \forall\,(s,t)\in S.$$

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 terminals 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 discussion 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 relaxation), 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^{st} 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 $\bar{\sigma}_{st}$, $\bar{\xi}_{ij}$, $\bar{\eta}_{ij}^{st}$ of constraints (69), (70), and (71), respectively.

Consider a terminal pair (s, t). A variable z_p corresponding to a path $p \in P^{st}$ has reduced cost $-\bar{\sigma}_{st} + \sum_{\{i,j\}\in p} \bar{\eta}_{ij}^{st}$. 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 $P^{st}\backslash P_0^{st}$, as its cardinality may be exponential in the number of vertices. Instead, we solve a shortest path problem on a directed graph $G_\eta(s, t)$ whose arcs (i, j) correspond to edges in G and have cost $\bar{\eta}_{ij}^{st}$. If the length of such a shortest path p is smaller than $\bar{\sigma}_{st}$, 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^{st} 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 ($\bar{x}_{ij}$, $\bar{\omega}_{ij}^{s+}$, $\bar{\omega}_{ij}^{s-}$, $\bar{z}_p$) and the associated dual solution ($\bar{\sigma}_{st}$, $\bar{\xi}_{ij}$, $\bar{\eta}_{ij}^{st}$);
4. For each terminal pair $(s, t) \in S$:
   - Build an auxiliary undirected graph $G_{\bar{\eta}}(s, t)$ such that the length of each edge is $\bar{\eta}_{ij}^{st}$;
   - Find the shortest path $\tilde{p}$ in $G_{\bar{\eta}}(s, t)$ and let the length of $\tilde{p}$ be $l_{\bar{\eta}}^*(s, t)$;
   - If $l_{\bar{\eta}}^*(s, t) < \bar{\sigma}_{st}$, then add the related variable $z_{\tilde{p}}$ to RLP;
5. If at least one variable has been added, go to Step 3. Otherwise, dual feasibility is obtained and RLP is solved.

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 $p' \equiv p$ 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 $z_{\check{p}}$ variable related to a terminal pair (s, t). We choose an edge $\check{e}$ contained in $\check{p}$, 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 $\check{e}$ are set to zero, while in N_{k+2} their sum must be equal to one. Once the constraints

$$\sum_{p\in P^{st}:\check{e}\in p} z_p = 0 \quad \text{and} \quad \sum_{p\in P^{st}:\check{e}\in p} z_p = 1$$

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 $\sum_{(i,j)\in\delta'(W)} (\omega_{ij}^{s+} + \omega_{ij}^{t-})$ is minimum, which amounts to solving the maximum (s, t)-flow problem on a graph whose edges $\{i, j\}$ have capacity $(\omega_{ij}^{s+} + \omega_{ij}^{t-})$. 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_{dual}), obtaining values $\tilde{\omega}_{ij}^{s+}$ and $\tilde{\omega}_{ij}^{s-}$ for all $\{i, j\} \in E$ and $s \in Q$;
3. *feasible* := TRUE;
4. For each $(s, t) \in S$:
   - Let $\tilde{G}_{st} = (V, A)$ be an auxiliary graph where $A = \{(i, j) : \{i, j\} \in E\}$ and let $u_{ij} = u_{ji} = \omega_{ij}^{s+} + \omega_{ij}^{t-}$ for all $(i, j) \in A$;

- Solve the maximum-flow/minimum-cut problem over $\tilde{G}_{st} = (V, A)$ 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 *feasible* := FALSE;

5. If *feasible* = 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, pacbell, toronto, usld: topologies of well-known backbone networks found in the IEEE literature;

res1/5/6/7/8/9, 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_s^+, b_s^-, d'_{st}, and $\hat{d}_{st}$ 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_{cf}) 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

TABLE 1. Results for small *Sym-G* instances: BPC algorithm and the compact linear MIP formulation solved with Cplex.

Name	$\|V\|$	$\|E\|$	$\|Q\|$	t_{root}	t_{tot}	#paths	#cuts	t_{cf}
arpanet	24	50	10	11.03	11.03	1219	1055	1.97
at-cep	15	22	6	0.02	0.02	211	100	0.03
bhv3	29	62	15	2.65	—	1695	1498	8.34
bhv6	27	39	15	0.14	0.03	1123	648	0.13
bhv13	29	36	13	0.48	0.04	1027	1212	0.86
cost239	11	22	5	0.02	0.02	129	46	0.07
eon	19	37	15	0.49	0.49	1272	714	1.28
latadl	39	86	17	397.27	397.27	3917	4259	219.55
metro	11	42	5	0.01	0.01	126	44	0.26
njlata	11	23	8	0.14	0.14	405	212	0.33
nor-sun	27	51	13	70.53	70.53	2019	1604	64.24
nsf	14	21	10	0.45	0.45	672	422	0.86
ny-cep	16	49	9	0.71	0.71	573	354	0.75
pacbell	15	21	7	0.09	0.09	294	176	0.16
toronto	25	55	11	2.53	2.53	1026	887	6.27
usld	28	45	15	7.55	7.55	1753	1496	4.17

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 2. Results for medium-to-large-size *Sym-G* instances: BPC algorithm and the compact linear MIP formulation solved with Cplex.

Name	$\|V\|$	$\|E\|$	$\|Q\|$	t_{root}	t_{tot}	#paths	#cuts	t_{cf}
res1	45	63	15	36.62	43.39	2843	3533	32.41
res5	44	67	18	174.21	174.21	3895	4641	58.78
res6	44	60	14	93.64	93.64	2820	2641	6.63
res7	44	62	18	2920.21	2920.21	11,341	10,019	37.68
res8	50	79	19	184.27	184.27	4860	6406	92.71
res9	50	76	23	1048.50	74.24	11,467	12,620	107.16
stein1	50	100	17	40.99	40.99	2778	2657	43.65
stein2	50	63	20	1.46	1.46	2488	2586	1.48
stein3	75	94	32	97.07	97.07	8192	16,084	61.88
stein4	75	150	33	3654.09	3654.09	11,124	16,411	3443.33
n45	45	63	20	5.89	6.51	3187	4512	38.62
n49	49	57	14	1.03	1.03	1357	1378	0.98
n147	147	265	37	18.80	18.80	7668	8211	52.72
g200	200	914	31	58.01	5986	6546	63.42	*mem*
t3-0	250	444	52	(0.95%)	—	97,025	246,679	*mem*
t3-1	250	456	53	(17.24%)	—	33,164	141,516	*mem*
t3-2	250	469	42	(1.30%)	—	44,395	220,123	*mem*
t3-3	250	446	50	6959.42	6959.42	48,190	363,584	*mem*
t3-4	250	465	39	(0.71%)	—	40,939	122,825	*mem*
t4-0	304	453	55	(0.02%)	—	42,524	293,763	*mem*
t4-1	304	457	63	(7.69%)	—	28,710	114,345	*mem*
t4-2	304	454	58	(19.30%)	—	28,335	96,122	*mem*
t4-3	304	458	67	(26.88%)	—	33,589	133,557	*mem*
t4-4	304	468	63	(0.82%)	—	43,349	275,168	*mem*

TABLE 3. Results for *Asym-G* instances: Compact linear MIP formulation solved with Cplex.

Name	$\|V\|$	$\|E\|$	$\|Q\|$	t_{cf}
arpanet	24	50	10	6.34
at-cep	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-cep	16	49	9	1.37
pacbell	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

[8] and it has been proved by Hurkens et al. [12] for networks consisting of a single cycle. Note that the situation for *Asym-G* differs slightly: most instances have an integer optimum with the same cost as the linear relaxation optimal solution, but we have found some instances with an integer optimum with larger cost.

Computational results for *Asym-G* are reported in Table 3. The compact flow-based formulation is remarkably tight and effective also in this case. All mid-size instances can be solved to optimality in a few minutes with the Cplex 8.1 MIP solver, except for "stein4," for which no integer solution could be found within the time limit. Among the large networks, only "t3-4" could be solved to optimality, whereas the remaining "tX-X" and "g200" could not be solved due to excessive memory requirements. In general, asymmetric instances appear to be more challenging than symmetric ones for the BPC approach. Because for 9 out of the 40 instances considered the gap remains large after reaching the time limit, we do not report BPC computing times.

It has still to be determined whether the substantial difference observed for several instances is mainly due to the remarkable quality of the compact linear MIP formulation or to a relevant margin for improving the BPC algorithm, by including for instance efficient memory management procedures that purge unused paths/cuts and ω 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 ω variables in the *Asym-G* MIP formulation (18)–(23) is twice the number of ω 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, Γ 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

TABLE 4. Results for small-size instances of the *robust* VPN provisioning problem: BPC algorithm and the compact linear MIP formulation solved with Cplex.

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%)

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-

TABLE 5. Results for medium to large-size instances of the *robust* VPN provisioning problem: BPC algorithm and the compact linear MIP formulation solved with Cplex.

Name	$\|V\|$	$\|E\|$	$\|Q\|$	t_{root}	t_{tot}	#paths	#cuts	t_{cf}
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	11,868	11,687	*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 ω 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

[1] E. Amaldi, P. Belotti, and M.Ç. Pınar, Multicommodity network design under polyhedral demand uncertainty, manuscript 2006, presented at 19th International Symposium on Mathematical Programming (ISMP), Rio de Janeiro, Brasil, July 2006.

[2] C. Barnhart, C.A. Hane, and P.H. Vance, Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems, Oper Res 48 (2000), 318–326.

[3] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W.P. Savelsbergh, and P.H. Vance, Branch-and-price: Column generation for solving huge integer programs, Oper Res 46 (1998), 316–329.

[4] W. Ben-Ameur and H. Kerivin, Routing of uncertain demands, Optimizat Eng 3 (2005), 283–313.

[5] D. Bertsimas and M. Sim, Robust discrete optimization and network flows, Math Program Ser. B 98 (2003), 43–71.

[6] D. Bertsimas and M. Sim, The price of robustness, Oper Res 52 (2004), 35–53.

[7] N. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. Ramakrishnan, and J.E. van der Merive, "A flexible model for resource management in Virtual Private Networks," Proc ACM SIGCOMM, Cambridge, MA, 1999, pp. 95–108.

[8] T. Erlebach and M. Rüegg, "Optimal bandwidth reservation in hose-model VPNs with multi-path routing," Proc IEEE Infocom 2004, 7–11 March 2004, vol. 4, pp. 2275–2282.

[9] M.R. Garey and D.S. Johnson, Computers and intractability: A guide to the theory of NP-completeness, W.H. Freeman, New York, NY, 1979.

[10] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener, "Provisioning a virtual private network: A network design problem for multicommodity flow," Proc ACM Symposium on Theory of Computing (STOC), Crete, Greece, 2001, pp. 389–398.

[11] A. Gupta, A. Kumar, and T. Roughgarden, "Simpler and better approximation algorithms for network design," Proc ACM Symposium on Theory of Computing (STOC), San Diego, CA, 2003, pp. 365–372.

[12] C. Hurkens, J. Keijsper, and L. Stougie, "Virtual private network design: a proof of the tree routing conjecture on ring networks," Proc 11th International IPCO conference, M. Jünger and V. Kaibel (Editors), Springer-Verlag, Lecture Notes in Computer Science series, vol. 3509, 2005, pp. 407–421.

[13] V. Jiménez and A. Marzal, "Computing the *K* shortest paths: A new algorithm and an experimental comparison," Algorithm Engineering, J.S. Vitter and C.D. Zaroliagis (Editors), Springer-Verlag, Lecture Notes in Computer Science series, vol. 1668, 1999, pp. 15–29.

[14] C. McDonald, Virtual private networks: An overview. `http://www.intranetjournal.com/foundation/vpn1.shtml`, 2004.

[15] A. Schrijver, Theory of linear and integer programming, John Wiley and Sons, New York, 1986.

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

[17] C. Swamy and A. Kumar, "Primal-dual algorithms for connected facility location problems," Proc International Workshop on Approximation Algorithms for Combinatorial Optimization (APPROX), Lecture Notes in Computer Science series, vol. 2462, 2002, pp. 256–270.