Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in DISCRETE MATHEMATICS

Electronic Notes in Discrete Mathematics 64 (2018) 315–324 www.elsevier.com/locate/endm

The asymmetric VPN tree problem: polyhedral results and Branch-and-Cut

Ibrahima Diarrassouba [a,1], Pedro Henrique P. V. Liguori [b,2,4] and A. Ridha Mahjoub [b,3]

[a] *Université Le Havre Normandie, LMAH, 76600, Le Havre, France*

[b] *Université Paris-Dauphine PSL Research University, CNRS, UMR [7243], LAMSADE, 75016 Paris, France*

Abstract

In this paper we consider a variant of the *virtual private network design problem* (VPNDP). Given an uncapacitated physical network, represented by a graph $G = (V \cup P, E)$, where V is the set of VPN routers and P is the set of clients for which it is given thresholds on the amount of traffic that each client can send (b_p^+) or receive (b_p^-), the VPNDP asks for (1) a connected sub-network $G' = (V' \cup P, E')$, (2) a client assignments (p, v), $p \in P$ and $v \in V'$, and (3) a bandwidth allocation u_e, $e \in E'$, in order to accommodate any traffic demand matrix that respects client thresholds. When G' is acyclic, we have a VPN tree (VPNT). Also, when client thresholds are asymmetric, *i.e.*, $\sum_{p \in P} b_p^+ \neq \sum_{b \in P} b_p^-$, the problem has been shown to be NP-hard. In this paper, we give MILP formulations for the asymmetric VPN tree problem. Also, we discuss the polytope associated with one of these formulations and describe several classes of valid inequalities. Moreover, we present necessary and sufficient conditions under which these inequalities define facets. We also devise separation routines. Using these routines, we propose a Branch-and-Cut algorithm and present a computational study.

Keywords: Asymmetric VPN tree, polyhedral approach, facet, Branch-and-Cut.

[1] Email: ibrahima.diarrassouba@univ-lehavre.fr

[2] Email: pedro-henrique.liguori@lamsade.dauphine.fr

[3] Email: mahjoub@lamsade.dauphine.fr

[4] Supported by CNPq

https://doi.org/10.1016/j.endm.2018.02.006

1571-0653/

1 Introduction

The general VPN Design Problem (VPNDP) can be presented in terms of a graph $G = (V \cup P, E)$, where P is the set of VPN clients and V is the set of network routers. Each edge in E represents a link between either two routers or a router and a client. We can assume, without loss of generality, that every client $p \in P$ is connected to a single router $i \in V$ and that the edges in E have unlimited capacities.

In the very popular *hose workload model* introduced by Duffield *et al.* [1], the bandwidth requirements of VPN clients can be modeled by defining thresholds $b_p^+ \geq 0$ and $b_p^- \geq 0$, $p \in P$, which represent the expected amount of data that client p can send and receive, respectively. Accordingly to this model, a demand traffic matrix $D \in \mathbb{R}_+^{P \times P}$ is said to be feasible if $\sum_{q \in P} D_{pq} \leq b_p^+$ and $\sum_{p \in P} D_{pq} \leq b_q^-$, for every $p, q \in P$, where the matrix entry D_{pq} represents the amount of information that client p can send to client q.

The minimum VPNDP consists in finding a subgraph $G' = (V' \cup P, E')$ of G spanning all the clients in P and a bandwidth allocation u_e, $e \in E'$, capable of routing any feasible demand traffic matrix, such that, the total bandwidth $\sum_{e \in E'} u_e$ is minimum. When G' is acyclic, we have a VPN tree. In this case, the problem is also referred as the minimum VPN Tree Problem (or VPNTP, for short). This problem has a wide range of practical applications, specially for Internet Service Providers [2].

Accordingly to the hose thresholds, one can say that the VPNTP is *symmetric* when $b_p^+ = b_p^-$ for all $p \in P$, and *balanced* when $\sum_{p \in P} b_p^+ = \sum_{p \in P} b_p^-$ (clearly, any symmetric VPNTP is also balanced, but the converse is not always true). In both cases, the problem has been shown to be polynomially solvable by Gupta *et al.* [2] and Italiano *et al.* [3], respectively. However, in its *asymmetric* version, *i.e.*, when $\sum_{p \in P} b_p^+ \neq \sum_{p \in P} b_p^-$, Gupta *et al.* [2] have shown that the Steiner Tree can be reduced to VPNTP, implying that the latter is NP-hard.

In this paper, we are interested in the *Asymmetric VPN Tree Problem* (AVPNTP, for short) from a polyhedral point of view. We propose two different MILP formulations and show that they are equivalent for the AVPNTP. Then, we discuss the polytope associated with one of these formulations, describe several classes of valid inequalities and derive necessary and sufficient conditions under which these inequalities are facet defining. We also discuss separation routines. These results are used afterwards to develop a Branch-and-Cut algorithm along with computational results are presented.

Although the problems related to designing efficient VPN networks have attracted much of attention in the past decade, only little effort has been dedicated in developing exact approaches. Gupta *et al.* [2] show that the AVPNTP could be solved as a sequence of $|V|$ MILP, each one consisting in computing a Steiner tree where the terminals correspond to the VPN clients. In [4], Diarrasouba *et al.* propose a Branch-and-Cut algorithm based on the formulation in [2]. In [5], Altin *et al.* consider the VPNDP without any constraint on the solution topology, *i.e.*, the solution can be any connected subgraph of $G = (V \cup P, E)$. In this case, they propose a multicommodity flow formulation to the problem. Very recently, in [6], Moradi *et al.* consider the latter VPNDP over the first Chvátal closure and demonstrate that strong LP relaxations can be obtained by adding rank-1 Chvátal-Gomory cuts. The authors also introduce a new ILP formulation and show that zero-half cuts are very efficient for solving hard instances of the problem.

The paper is organized as follows. In Section 2, a MILP formulation for the AVPNTP along with its relaxation are introduced. In Section 3, we discuss this relaxation and the polytope associated with it. Moreover, we describe the valid inequalities and discuss their facial aspects. In Section 4, we present the Branch-and-Cut algorithm and some computational results. Finally, in Section 5 some conclusions and perspectives for future work are given.

In what follows, we give the basic notations we shall use along the paper. We denote by $E(W) \subseteq E$ the subset of edges having both endpoints in W. An edge cut-set, denoted by $\delta(W) \subseteq E$, is the subset of edges having one endpoint in W and another in $V \setminus W$. For a given vertex $v \in V$, we denote by $\delta(v) \subseteq E$ the subset of edges incident on v.

2 AVPNTP formulation

In [2] it is shown that any asymmetric VPN tree can be completely characterized by a subset of nodes S, $S \subseteq V$, called the *core set*, such that the induced subgraph $G[S] = (S, E(S))$ is connected. Moreover, from a given core set S it is always possible to build a valid VPNT, $T(S)$, as follows: connect all the nodes in S by a spanning tree. Add the edges of this spanning tree to $T(S)$. Then, merge all the nodes of S into a single super-node, say w, and construct a breadth-first tree rooted at w connecting all the VPN clients in P as leaves. Finally, add the edges of this tree to $T(S)$.

Let $C(S)$, the cost associated with the solution $T(S)$, be defined as

$$C(S) = \hat{B}(|S| - 1) + \sum_{p \in P} \min_{v \in S} \{d_G(v, p)\} (b_p^+ + b_p^-), \tag{1}$$

where $\hat{B} = \min\{\sum_{p\in P} b_p^+,\ \sum_{p\in P} b_p^-\}$ and $d_G(v,p)$ is the length of the shortest path (in number of hops) between core node v and client p in G. In [2], Gupta *et al.* also proved that, for a core set S that minimizes (1), $T(S)$ is an optimal VPN tree for the asymmetric case with total bandwidth allocation $C(S)$.

Consequently, it is possible to restate the AVPNTP in terms of a weighted mixed graph $H = (V \cup P, E, A)$, where P and V are respectively the set of VPN clients and the set of VPN routers. The set E is the set of edges that represent the links between the core nodes while the set A is the set of arcs that represent all possible assignments of the clients to the routers, *i.e.*, $A = \{(p,v) \in P \times V\}$.

As suggested in [2], one can associate a cost $f_a = d_G(v,p)(b_p^+ + b_p^-)$ with every arc $a = (p,v) \in A$, representing the amount of bandwidth required to assign client p to core node v. Still, as proved in [2], the core edges must accommodate the worst case scenario of data flowing on the network, requiring a bandwidht reservation of $\hat{B}$ on these edges.

In this reformulated version, the AVPNTP is equivalent to solve the following problem: find a tree (S, E') in the *core graph* $G = (V,E) \subset H$ and an assignment $A' \subseteq A$, of each client $p \in P$ to a node $v \in S$, such that $\sum_{e\in E'} \hat{B} + \sum_{(p,v)\in A'} d_G(v,p)(b_p^+ + b_p^-)$ is minimum.

Let y_i be a variable that takes 1 if node $i \in V$ is selected in the core set and 0 otherwise. Let z_e be a variable which takes 1 if edge $e = ij$ is used to connect core nodes i and j. For an arc $(p,i) \in A$, let $x_{(p,i)}$ be a variable which takes 1 if p is assigned to i and 0 otherwise. Then, it is not hard to see that every solution of the AVPNTP satisfies the following inequalities

$$\sum_{i\in V} x_{(p,i)} = 1, \qquad \forall \quad p \in P \tag{2a}$$

$$\sum_{e\in E} z_e = \sum_{i\in I} y_i - 1, \tag{2b}$$

$$\sum_{e\in E(S)} z_e \le \sum_{i\in S\setminus\{j\}} y_i, \qquad \forall \quad S \subseteq V,\ |S| \ge 2,\ j \in S \tag{2c}$$

$$x_{(p,i)} \le y_i, \qquad \forall \quad (p,i) \in A \tag{2d}$$

$$z_e \le y_i,\ z_e \le y_j, \qquad \forall \quad e = ij \in E \tag{2e}$$

$$0 \le z_e \le 1, \qquad \forall \quad e \in E \tag{2f}$$

$$0 \le y_i \le 1, \qquad \forall \quad i \in V \tag{2g}$$

$$0 \le x_a \le 1, \qquad \forall \quad a \in A. \tag{2h}$$

Inequalities (2a) guarantee that every client p is assigned to exactly one core node. Inequalities (2b) and (2c) ensure that the solution is a tree. The former states that the number of edges in the solution is exactly one unit less than the number of nodes in the core set, while the latter are the generalized sub-tour elimination constraints that guarantee the solution is acyclic. Inequalities (2d) express the fact that if node i is not in the core set then no client p can be assigned to it. Similarly, inequalities (2e) indicate that if a vertex i is not in the solution, then any edge incident on it can not be in the solution either. Constraints (2f)–(2h) are the trivial inequalities.

Let $\mathcal{R}_1$ be the polyhedron given by (2a)–(2h). The AVPNTP is equivalent to the following integer program

$$\min\left\{\sum_{a\in A} d_G(a)B_p x_a + \hat{B}\sum_{e\in E} z_e \;:\; (\mathbf{x},\mathbf{y},\mathbf{z}) \in \mathcal{R}_1 \cap \mathbb{Z}^{|A|+|V|+|E|}\right\}, \tag{3}$$

where $B_p = (b_p^+ + b_p^-)$.

Remark that variables $\mathbf{x}$ can be considered continuous. In fact, when variables $(\mathbf{y},\mathbf{z})$ are restricted to $\{0,1\}$, so it is variables $\mathbf{x}$.

A relaxation of formulation (3) can be obtained by first substituting constraints (2a) for a lighter counterpart of covering inequalities of the form

$$\sum_{i\in V} x_{(p,i)} \geq 1, \quad \forall\, p \in P. \tag{4}$$

Moreover, it is also possible to relax the requirement that the underlying solution is a tree. With this purpose, one can enforce the solution connectivity by replacing (2b) and (2c) by cut-set inequalities of the form

$$\sum_{e\in\delta(S)} z_e \geq y_i + y_j - 1, \quad \forall\, S \subset V,\ S \neq \emptyset,\ i \in S,\ j \in V \setminus S, \tag{5}$$

assuring that, for any edge cut-set induced by S, whenever two variables associated with routers y_i and y_j, lying in opposite sides of the edge cut-set, take value one, then at least one edge in cut must also be set to one.

Then, consider the alternative polyhedron $\mathcal{R}_2$ given by (2d) - (2h), (4) and (5). It follows that a relaxation for AVPNTP (hereafter called r-AVPNTP) is given by

$$\min \left\{ \sum_{a \in A} d_G(a) B_p x_a + \hat{B} \sum_{e \in E} z_e \ : (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathcal{R}_2 \cap \mathbb{Z}^{|A|+|V|+|E|} \right\}. \tag{6}$$

It is clear that all solutions of (3) are also solutions of (6). Furthermore, given that all the weights are non-negative, every optimal solution of (6) must satisfy constraints (2a), (2b) and (2c). Hence, it is also a solution of (3). This implies that, for the AVPNTP, both formulations (3) and (6) are equivalent.

In the rest of the paper, we are going to consider the formulation (6) for the AVPNTP.

3 Polyhedral analysis

In this section we are going to investigate some polyhedral aspects of problem (6). Let $Q(H)$ be the convex hull of all integer solutions of (6), *i.e.*,

$$Q(H) = \text{conv}\left\{ (\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathbb{Z}^{|A|+|V|+|E|} : (\mathbf{x}, \mathbf{y}, \mathbf{z}) \text{ satisfies (2d)} - \text{(2h)}, \text{(4) and (5)} \right\}.$$

First, we characterize the dimension of $Q(H)$.

Theorem 3.1 *The polytope $Q(H)$ is full-dimensional if and only if the core graph $G = (V, E) \subset H$ contains a cycle.*

In what follows, we consider that the core graph $G = (V, E) \subset H$ contains at least one cycle, implying that $Q(H)$ is full-dimensional. The next theorems characterize when inequalities (2d) - (2h), (4) and (5) define facets of $Q(H)$.

Theorem 3.2 *For an edge $uv \in E$, inequality $z_{uv} \geq 0$ defines a facet for $Q(H)$ if and only if (i) both vertices u and v have degree of at least 2 in the core graph $G = (V, E) \subset H$ and (ii) there must exist a cycle in the core graph G that does not contain uv.*

Theorem 3.3 *For a core node v, inequality $y_v \leq 1$ defines a facet for $Q(H)$ if and only if the core graph $G = (V, E) \subset H$ is 2-edge connected.*

Theorem 3.4 *For an assignment (q, u) of client q to core node u, inequality $x_{(q,u)} \geq 0$ defines a facet for $Q(H)$ if and only if node u has degree of at least 2 in the core graph $G = (V, E) \subset H$.*

Theorem 3.5 *Inequalities (2d), (2e) and (4) define facets for $Q(H)$.*

Theorem 3.6 *For a disjoint partition $(S, V \setminus S)$ of the vertex set V and a pair u, v of vertices lying in S and $V \setminus S$, respectively, the cut-set inequality (5) defines a facet for $Q(H)$ if and only if both induced subgraphs $G[S] = (S, E(S))$ and $G[V \setminus S]$ are 2-edge connected.*

3.1 Valid inequalities

3.1.1 Tree lower bound inequality

The following inequality is clearly valid for r-AVPNTP and its validity relies on the fact that any connected subgraph has at least $n-1$ active edges in the solution (where n is the number of active vertices):

$$\sum_{e \in E} z_e \geq \sum_{i \in V} y_v - 1. \tag{7}$$

Theorem 3.7 *The tree lower bound inequality (7) defines facet for $Q(H)$.*

3.1.2 Degree-one assignment inequalities

Let $i \in V$ be a vertex such that $|\delta(i)| = 1$. Then, the following inequalities are valid for r-AVPNTP

$$x_{(p,i)} \geq y_i - z_e, \quad \forall\, p \in P,\ i \in V : |\delta(i)| = 1,\ e \in \delta(i). \tag{8}$$

Theorem 3.8 *Degree-one assignment inequalities (8) define facets for $Q(H)$.*

3.1.3 Spanning tree inequalities

Denote by $T = (V, E_T)$, with $E_T \subseteq E$, a spanning tree of the core graph $G = (V, E) \subset H$. Then, the inequalities below are valid for the r-AVPNTP

$$\sum_{e \in E_T} z_e \leq \sum_{i \in V} y_i - 1, \quad \forall\, T \subseteq G. \tag{9}$$

Theorem 3.9 *Spanning tree inequalities (9) define facets for $Q(H)$.*

4 Branch-and-Cut algorithm

One of the pillars of the development of a good Branch-and-Cut algorithm is to devise efficient separation routines. As it is well-known in the literature, cut-set inequalities of type (5) can be efficiently separated by using *push-relabel* algorithm to compute the max-flow through every pair of vertices $u, v \in V$. In order to separate spanning tree inequalities of type (9), we propose a slightly adaptation of *Kruskal algorithm* to find a maximum spanning tree in the support graph induced by a fractional solution $(\bar{\mathbf{z}}, \bar{\mathbf{y}})$. To this purpose, it suffices to sort the edges in decreasing order of values $\bar{\mathbf{z}}$ and then proceed with regular Kruskal algorithm to find a maximum spanning tree T in G. If the weight of the maximum spanning tree, $\sum_{e \in E(T)} \bar{z}_e$, is greater than $\sum_{v \in V} \bar{y}_v - 1$, a violated cut has been found. Otherwise, all spanning tree inequalities are verified by $(\bar{\mathbf{z}}, \bar{\mathbf{y}})$. We remark that this algorithm is exact and runs in $\mathcal{O}(|E| \log |V|)$.

The testbed is composed by a subset of *Waxman* instances introduced in [4], where the clients and their thresholds are kept the same for all instances while the core graph is augmented both in number of vertices and edges. The Branch-and-Cut algorithm was implemented using C++ language, having CPLEX 12.6 as MILP solver engine. All the CPLEX heuristics as well as automatic cut generation were turned off. The tests were conducted in a personal computer with processor Intel Core i7 and 8 Gb RAM. A time limit of 300 seconds was given to all instances.

Table 1 summarizes some preliminary results. The first three columns give general information about the instances: the quantity of VPN clients followed by the quantity of vertices and edges in the core graph. Next column, *Opt.*, gives the optimal integer solution value for AVPNTP. It is followed by column *LP* $\mathcal{R}_2$ which gives the optimal solution for the linear relaxation of (6). Column *LP* $\mathcal{R}_2^R$ presents the value for the linear relaxation of the same formulation strengthened by additional valid inequalities (7)–(9). The next column contains the relative gap in percentage between the integer solution (column *Opt.*) and the solution over $\mathcal{R}_2^R$ (column *LP* $\mathcal{R}_2^R$).

From the analysis of the linear relaxation performance indicated in Table 1, we remark that the linear relaxation of (6) depends only on the thresholds of clients in P and does not vary with the changes in the core graph, indicating that formulation performs very poorly. This behavior can be explained by the fact that polyhedron $\mathcal{R}_2$ admits the following trivial fractional solution, which is an optimal solution with respect to the AVPNTP objective function: for all core vertices $i \in V$ make $y_i = 0.5$. Then, for each client $p \in P$, chose two of the nearest core vertices u and v in V, and make $x_{(p,u)} = x_{(p,v)} = 0.5$. Every core edge $uv \in E$ can be set to zero, *i.e.*, $z_{uv} = 0$. However, things change dramatically in $\mathcal{R}_2^R$, indicating that the valid inequalities have played an important role strengthening the previous formulation. It is easy to verify

$\|P\|$	$\|V\|$	$\|E\|$	Opt.	LP relax.			B&C $\mathcal{R}_2^R$			
				LP $\mathcal{R}_2$	LP $\mathcal{R}_2^R$	gap	*NSub*	*cs*	*sp*	t(s)
10	10	20	105	50.5	105	0.03	2	0	0	0
10	15	30	137	50.5	147.33	0.07	9	156	0	0.03
10	20	40	152	50.5	142.75	0	17	631	0	0.11
10	25	50	134	50.5	127	0	0	0	0	0
10	30	60	208	50.5	186.29	0.03	8	114	0	0.03
10	35	70	184	50.5	191.88	0.16	121	50,604	0	24.23
10	40	80	194	50.5	192.14	0.14	25	7,875	0	1.92
10	45	90	201	50.5	205.67	0.1	41	14,248	0	3.67
10	50	100	218	50.5	214.46	0.13	145	142,894	0	206.3
10	55	110	241	50.5	215.25	0.12	67	164,782	0	178.98

Table 1
Branch-and-Cut performance

that valid inequality (3.7) cuts-off the previous optimal fractional solution.

The last four columns of Table 1 present general information about the Branch-and-Cut algorithm. Column $NSub$ indicates the total number of nodes explored in the Branch-and-Bound tree. Column *cs* gives the total amount of cut-set inequalities (5) separated, and it is followed by column *sp* which indicates the total amount of spanning tree inequalities (9) separated. Last column gives the execution time in seconds. We report that the performance of the Branch-And-Cut based on our formulation outperforms the results presented in [4], giving linear relaxation values of at least half of the values for the three biggest instances while keeping the same average quality for the smaller ones. Also, when it comes to execution time, our algorithm was able to solve faster all but one instance. Unfortunately, the authors do no report in [4] the number of explored nodes in the Branch-and-Bound tree.

Although spanning tree inequalities (9) always define facets, as indicated by Theorem 3.9, no violated inequality of that type was separated for any of the instances in the testbed. This can be explained by the specific structure of the objective function of AVPNTP, which seeks to minimize the total amount of edges in the solution. However, we point out that this behavior can easily change when different configurations of objective function are used. For example, multiplying by -1 all the costs associated with variables $\mathbf{z}$ is sufficient to make appear violated inequalities of type (9).

5 Conclusion

In this work we have shown how the polyhedral approach can be used for designing an efficient algorithm for the AVPNTP. We have introduced two different formulations and conducted a polyhedral investigation for one of them. Moreover, we have presented some classes of valid inequalities and studied their facial aspects. Also, the computational experiments have shown that this can be indeed a good approach for solving these problems. Besides the fact that it can be interesting looking for new classes of valid inequalities, it can also be a good perspective for future work to investigate some correlated problems as, for example, the *Connected Facility Location Problem*, for which we can easily adapt the formulations here presented.

Acknowledgement

The author Pedro Henrique P. V. Liguori is partially funded by CNPq grant 234402/2014-8.

References

[1] Duffield, N., Goyal, P., Greenberg, A., Mishra, P., Ramakrishnan, K., and van der Merwe, J., *A flexible model for resource management in virtual private networks*, ACM SIGCOMM Computer Communication Review. **29** (1999), 95–108.

[2] Gupta, A., Kleinberg, J., Kumar, A. Rastogi, R., and Yener, B., *Provisioning a virtual private network: A network design problem for multicommodity flow*, Proceedings of the thirty-third annual ACM symposium on Theory of computing. (2001), 389–398.

[3] Italiano, G., Leonardi, S., and Oriolo, G., *Design of trees in the hose model: The balanced case*, Operations Research Letters **6** (2006), 601–606.

[4] Diarrasouba, I., Mahjoub, A. R., Lourimi, A., and Youssef, H., *Hose workload based exact algorithm for the optimal design of virtual private networks*, Computer Networks **57** (2013), 2766–2774.

[5] Altin, A., Amaldi, E., Belotti, P., and Pinar, M., *Provisioning virtual private networks under traffic uncertainty*, Networks **49** (2007), 100–115.

[6] Moradi, A., Lodi, A., and Mehdi Hashemi , S., *Virtual private network design over the first Chvátal closure*, Operations Research **49** (2015), 569–588.