A facial study of the asymmetric VPN tree problem

Ibrahima Diarrassouba[1], Pedro Henrique P. V. Liguori[2], A. Ridha Mahjoub[2]

[1] Université Le Havre Normandie, LMAH, 76600, Le Havre, France
{ibrahima.diarrassouba}@univ-lehavre.fr

[2] Université Paris-Dauphine PSL Research University, CNRS, UMR [7243], LAMSADE, 75016 Paris, France
{pedro-henrique.liguori,mahjoub}@lamsade.dauphine.fr

Keywords : *Asymmetric VPN tree, polyhedral approach, facet.*

1 Introduction

Given an uncapacitated physical network, represented by a graph $G = (V, E)$, and a set of clients P, for which it is also known the thresholds on the amount of traffic that each client can send (b_p^+) and receive (b_p^-), the *virtual private network design problem* asks for (1) a connected sub-network $G' = (V', E')$, with $V' \subseteq V$ and $E' \subseteq E$; (2) a client assignment (p, v), $p \in P$ and $v \in V'$, and; (3) a bandwidth allocation u_e, $e \in E'$, in order to accommodate any demand traffic matrix $D \in \mathbb{R}^{P \times P}$, with $D_{pq} \geq 0$, that respects client thresholds, *i.e.*, $\sum_{q \in P} D_{pq} \leq b_p^+$ and $\sum_{p \in P} D_{pq} \leq b_q^-$. When G' is acyclic, we have a VPN tree (VPNT). The *Minimum VPN Tree Problem* (VPNTP) asks for a VPNT whose bandwidth allocation, *i.e.*, $\sum_{e \in E'} u_e$, is minimum. It has been shown by [1] that, when client thresholds are asymmetric, *i.e.*, $\sum_{p \in P} b_p^+ \neq \sum_{b \in P} b_p^-$, the VPNTP belongs to NP-hard.

In this work, we are interested in the *asymmetric* version of VPNTP (hereafter AVPNTP). We give a MILP for the AVPNTP and derive necessary and sufficient conditions under which the basic inequalities define facets.

The MILP formulation introduced here deeply relies on some structural properties of asymmetric VPN trees that can be found in [1]. Let y_i be a variable that takes 1 if router $i \in V$ is in the solution and 0 otherwise. Let z_e be a variable which takes 1 if edge $e = ij$ is used to connect routers i and j. For a pair (p, i), with $p \in P$ and $i \in V$, let $x_{(p,i)}$ be a variable which takes 1 if client p is assigned to router i in the solution and 0 otherwise. Thus, it follows that a relaxation to AVPNTP can be given by

$$\min \quad \sum_{a \in A} d_G(a) B_p x_a + \hat{B} \sum_{e \in E} z_e$$

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

$$x_a \leq y_i, \qquad \forall \quad a = (p, i) \in A \tag{1b}$$

$$z_e \leq y_i, \ z_e \leq y_j, \qquad \forall \quad e = ij \in E \tag{1c}$$

$$\sum_{e \in \delta(S)} z_e \geq y_i + y_j - 1, \qquad \forall \quad S \subseteq V, \ |S| \geq 2, \ i \in S, \ j \in V \setminus S \tag{1d}$$

$$\sum_{e \in E} z_e \geq \sum_{i \in V} y_i - 1 \tag{1e}$$

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

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

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

where $A = P \times V$, $B_p = b_p^+ + b_p^-$, $\hat{B} = \min\{\sum_{p \in P} b_p^+,\ \sum_{p \in P} b_p^-\}$, and $d_G(v,p)$ is the length of the shortest path (in number of hops) between router v and client p.

Inequalities (1a) guarantee that every client p is assigned to at least one router. Inequalities (1b) and (1c) state that if router i is not in the solution then no client p can be assigned to it neither any edge having it as an endpoint can be in the solution. Inequalities (1d) ensure that the solution is connected. Inequality (1e) is a tree lower bound stating that the number of edges in any solution must be at least equal the number of vertices in the solution minus one unit. Constraints (1f)–(1h) are the trivial inequalities.

We point out that formulation (1a)–(1h) is close related to the one proposed by [1]. However, in order to obtain a more adequate formulation to the polyhedral investigation here conducted, constraints (1a) and (1e) are relaxations of the original constraints found in [1].

2 Polyhedral analysis

Let $Q(G)$ be the convex hull of all integer solutions of (1a)–(1h), *i.e.*,

$$Q(G) = \text{conv}\left\{(\mathbf{x},\mathbf{y},\mathbf{z}) \in \mathbb{Z}^{|P|\times|V|+|V|+|E|} : (\mathbf{x},\mathbf{y},\mathbf{z}) \text{ satisfies } (1a) - (1h)\right\}.$$

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

Theorem 1 *The polyhedron $Q(G)$ is full-dimensional if and only if the graph $G = (V,E)$ contains a cycle.*

The next theorems characterize when inequalities (1a)–(1h) define facets of $Q(G)$.

Theorem 2 *For a given edge $uv \in E$, inequality $z_{uv} \geq 0$ defines a facet for $Q(G)$ if and only if (i) both vertices u and v have degree greater than 1 in $G = (V,E)$ and (ii) there must exist a cycle in G that does not contain uv.*

Theorem 3 *For a given router v, inequality $y_v \leq 1$ defines a facet for $Q(G)$ if and only if the graph $G = (V,E)$ is 2-edge connected.*

Theorem 4 *For a given assignment (q,u), of client q to router u, inequality $x_{(q,u)} \geq 0$ defines a facet for $Q(G)$ if and only if node u has a degree of at least 2 in the graph $G = (V,E)$.*

Theorem 5 *Inequalities* (1a)*,* (1b)*,* (1c) *and* (1e) *define facets for $Q(G)$.*

Theorem 6 *For a given disjoint partition $(S,\bar{S})$ of the vertex set V, and a pair u,v of vertices lying in S and $\bar{S}$, respectively, cut-set inequality (1d) defines a facet for $Q(G)$ if and only if both subgraphs $G[S]$ and $G[\bar{S}]$, induced by S and $\bar{S}$, are 2-edge connected.*

3 Conclusions and future work

In this work we presented a MILP formulation to the AVPNTP and give necessary and sufficient conditions under which the basic inequalities define facets. Besides the fact that it can be interesting looking for new classes of valid inequalities and develop a Branch-and-Cut algorithm, we think that 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 formulation here presented.

References

[1] 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.