Laboratoire d'Analyses et Modélisation de Systèmes pour l'Aide à la Décision
UMR 7243

CAHIER DU LAMSADE

332

Janvier 2013

Two Node-Disjoint Hop-Constrained Survivable Network Design and Polyhedra

Ibrahima Diarrassouba, Hakan Kutucu, A. Ridha Mahjoub

Two Node-Disjoint Hop-Constrained Survivable Network Design and Polyhedra

Ibrahima Diarrassouba[a], Hakan Kutucu[b], A. Ridha Mahjoub[c]

[a] *Department of Mathematics, Le Havre University, Le Havre, France*
[b] *Department of Mathematics, Izmir Institute of Technology, Izmir, Turkey*
[c] *Laboratoire LAMSADE, Université Paris-Dauphine, Paris, France*

Abstract

Given a weighted undirected graph G with a set of pairs of terminals $\{s_i, t_i\}, i = 1, ..., d$, and an integer $L \geq 2$, the two node-disjoint hop-constrained survivable network design problem (TNHNDP) is to find a minimum weight subgraph of G such that between every s_i and t_i there exist at least two node-disjoint paths of length at most L. This problem has applications to the design of survivable telecommunications networks with QoS-constraints. We discuss this problem from a polyhedral point of view. We present several classes of valid inequalities along with necessary and/or sufficient conditions for these inequalities to be facet defining. We also discuss separation routines for these classes of inequalities. Using this, we propose a Branch-and-Cut algorithm for the problem when $L = 3$, and present some computational results.

Keywords: Survivable network, node-disjoint paths, hop constraint, polyhedron, facet, branch and cut

1. Introduction

Given a weighted undirected graph $G = (N, E)$, an integer $L \geq 2$, and a set of demands $D \subseteq N \times N$, the two node-disjoint hop-constrained survivable network design problem (TNHNDP) consists in finding a minimum wieght subgraph of G containing at least two node-disjoint paths of at most L hops

Email addresses: diarrasi@univ-lehavre.fr (Ibrahima Diarrassouba), hakankutucu@iyte.edu.tr (Hakan Kutucu), mahjoub@lamsade.dauphine.fr (A. Ridha Mahjoub)

between each pair of nodes $\{s, t\}$ in D. The case in which there are one source and several destinations is called the rooted TNHNDP.

In this work, we consider the associated polytope. We give some classes of valid inequalities along with necessary and/or sufficient conditions for these inequalities of the formulation to define facets. We also derive separation procedures for these inequalities. Using these results, we finally develop a branch-and cut algorithm and discuss some computational results for $L = 3$.

Given an edge subset $F \subseteq E$, the $0 - 1$ vector $x^F \in \mathbb{R}^{|E|}$, such that $x^F(e) = 1$ if $e \in F$ and $x^F(e) = 0$ otherwise, is called the *incidence vector* of F. The convex hull of the incidence vectors of the solutions to the TNHNDP on G, denoted by $P(G, L)$, will be called the TNHNDP polytope. Given a vector $w \in \mathbb{R}^{|E|}$ and an edge subset $F \subseteq E$, we let $w(F) = \sum_{e \in F} w(e)$. If $W \subset N$ is a node subset of G, then the set of edges that have only one node in W is called a cut and denoted by $\delta(W)$. We will write $\delta(v)$ for $\delta(\{v\})$. A cut $\delta(W)$ such that $s \in W$ and $t \in N \setminus W$ will be called an st-cut. Given a node $z \in N$, the graph $G - z$ is the subgraph obtained from G by deleting the node z and all its incident edges (but not their other end nodes). Let $V_0, V_1, \ldots, V_{L+1}$ be a partition of N such that $s \in V_0$, $t \in V_{L+1}$, and $V_i \neq 0$ for all $i = 1, \ldots, L$. Let T be the set of edges $e = uv$, where $u \in V_i$, $v \in V_j$, and $|i - j| > 1$.

In [3], Gouveia et al. discuss the TNHNDP within the context of an MPLS (Multi-Protocol Label Switching) network design model. The authors propose two extended formulations involving one set of variables for each path between each pair of terminals. The first model uses standard flow variables, and the second uses hop-indexed variables. Each subsystem of constraints associated with a path is a flow model with additional cardinality constraints. The authors also introduce a third model involving one set of hop-indexed variables for each pair of terminals. They show that this aggregated and more compact model produces the same linear programming bound as the multipath hop-indexed model. They also present computational results for $L = 4, 5$, and 6 using these formulations. Unfortunately, as the number of variables of the resulting models grows with L (and the number of pairs of terminals), the size of the corresponding linear programming relaxation may lead to excessive computational time when more dense instances (or instances with a larger value of L or a larger number of nodes) are considered. As mentioned in [3], this points out the need for looking for formulations using only natural variables.

The edge version of the problem (TEHNDP) has been already investi-

gated by several authors when $L = 2, 3$. In particular, [4] give a complete and minimal linear description of the corresponding polytope when $L = 2, 3$ and $|D| = 1$. In [16] and [21], Huygens et.al. and Diarrassouba have studied the problem when $|D| \geq 2$ and when two and k edge-disjoint paths, respectively, are required. They devise Branch-and-Cut algorithms for the problem when $L = 2, 3$ and give some computational results. Also, Huygens and Mahjoub [2] have studied the two versions of the problem (TNHNDP and TEHNDP) when $L = 4$. They give an integer programming formulation for the problem in the two cases. In [6], Dahl considers the hop-constrained path problem, that is the problem of finding between two distinguished nodes s and t a minimum cost path with no more than L edges when L is fixed. He gives a complete description of the dominant of the associated polytope when $L \leq 3$. Dahl and Gouveia [1] consider the directed hop-constrained path problem. They describe valid inequalities and characterize the associated polytope when $L \leq 3$. In [5], Coullard et al. investigate the structure of the polyhedron associated with the *st*-walks of length L of a graph, where a walk is a path that may go through the same node more than once. Dahl et al. [7] also consider the hop-constrained walk polytope in directed graphs when $L = 4$. This is the first work that addresses a polyhedral analysis for a hop-constrained network design problem with $L = 4$. By introducing extended variables in addition to the design variables, the authors characterize the polytope. They also introduce a large class of facet-defining inequalities for the dominant of that polytope, which surprisingly shows that describing the hop-constrained walk polytope for $L = 4$ is easier than describing its dominant. Moreover, one of their conclusions is that the structure of the hop-constrained path polytope for $L = 4$ is considerably more complicated.

Further hop constrained survivable network design problems are studied in [9, 10, 11, 13, 14, 15]. A survey of survivability, which also includes a section with hop-constraints, can be found in [12].

To the best of our knowledge, this paper is the first work for node case of the problem from a polyhedral point of view.

The following linear system along with the integrality constraints formulates the THNDPP as an integer program when $L = 2, 3, 4$ (see [2]).

$$x(\delta(W)) \geq 2, \qquad \text{for all } st\text{-cuts } \delta(W), \tag{1.1}$$

$$x(\delta_{G-z}(W)) \geq 1, \quad \text{for all } st\text{-cuts } \delta_{G-z}(W), \text{ for all } z \in N \backslash \{s, t\}, \tag{1.2}$$

$$x(T) \geq 2, \qquad \text{for all } L\text{-path-cuts } T, \tag{1.3}$$

$$x(T_{G-z}) \geq 1, \qquad \text{for all } L\text{-path-cuts } T_{G-z}, \text{ for all } z \in N \setminus \{s,t\}, \tag{1.4}$$

$$x(e) \leq 1, \qquad \text{for all } e \in E, \tag{1.5}$$

$$x(e) \geq 0, \qquad \text{for all } e \in E. \tag{1.6}$$

Inequalities (1.1),(1.2),(1.3) and (1.4) are called *st-cut inequalites*, *st-node-cut inequalites*, *L-path-cut inequalites* and *L-path node-cut inequalites*, respectively. The remaining inequalities are called *trivial inequalites*.

This formulation is called the *Natural Formulation*. We will also call inequalities (1.1)-(1.4) the *basic inequalities* (here "basic" means that they are necessary in the natural formulation).

The above formulation is not complete that is its linear relaxation is not integral when $L = 3, 4$. Consider the graph $G = K_5$ in Figure 1, where the edges in solid lines have value 1/2, the ones in bold have value 1, and the remaining edges have value zero. It is easy to verify that this solution is a fractional extreme point of the polyhedron given by the linear relaxation of the node TNHNDP with $L = 3$ and $\{s,t\} = \{1,4\}$. This point can be cut off by the following valid inequality

$$x(e_1) + 2x(e_2) + x(e_3) + 2x(e_4) + x(e_5) + 2x(e_8) \geq 3.$$

Moreover, this inequality is facet defining for the polytope on this graph. A solution which is a fractional extreme point of the polyhedron given by the LP relaxation of TNHNDP when $L = 4$ is given in [2]. Therefore, for a complete description of TNHNDP when $L = 3, 4$, some valid inequalities addition to (1.1)-(1.6) are necessary.

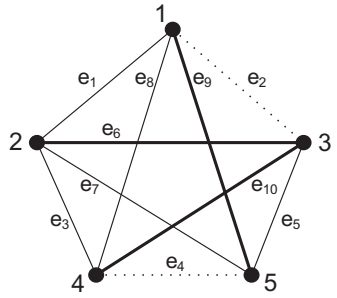

Figure 1: A fractional extreme point of the linear relaxation of the TNHNDP with $L = 3$ and $\{s,t\} = \{1,4\}$.

Theorem 1. *If the graph G is complete and all edge costs are equal to 1, the rooted TNHNDP can be solved in polynomial time for every $L \geq 2$.*

Proof. In [16], Huygens at al. proposed a polynomial algorithm for the rooted two-edge connected hop-constrained network design problem if the graph is complete and all edge costs are equal to 1. This algorithm actually produces two-node connected graph. ■

The paper is organized as follows. In the next section, we present a new class of valid inequalities for TNHNDP. In Section 3, we give further valid inequalities for the rooted TNHNDP. Necessary and sufficient conditions for these inequalities to be facet defining are discussed in Section 4. In Section 5 and 6, we study the separation of these inequalities and propose a heuristic algorithm to find good lower bound, respectively. In Section 7, we derive a branch-and-cut algorithm and present our experimental results. Finally, we give some concluding remarks in Section 8.

2. Valid Inequalities

In this section we present valid inequalities for the TEHNDP that have been introduced in the literature. As we will see in the following, they are also valid for the TNHNDP.

2.1. Generalized L-path-cut Inequalities

Theorem 2. *[1] Let $V_0, \ldots, V_{L+r}$ be a partition of N such that $r \geq 1, s \in V_0$ and $t \in V_{L+r}$. Then the generalized jump inequality is*

$$\sum_{e \in [V_i, V_j], i \neq j} \min(|i-j|-1, r)x(e) \geq r$$

is valid for the L-hop-constrained path problem.

These inequalities can be easily extended to the TNHNDP as follows:

$$\sum_{e \in [V_i, V_j], i \neq j} \min(|i-j|-1, r)x(e) \geq 2r. \tag{2.7}$$

Note that inequalities (2.7) generalize the L-path-cut inequalities (by setting $r = 1$).

2.2. Rooted Partition Inequalities

Theorem 3. *[16] Let $L = 3$ and $T = \{t_1, \ldots, t_p\}$ be a subset of p destination nodes relative to node s. Let $\pi = (V_0, V_1, \ldots, V_p)$ be a partition of N such that $s \in V_0$, and $t_i \in V_i$, for all $i = 1, \ldots, p$. Then the inequality*

$$x(\delta(V_0, \ldots, V_p)) \geq \lceil 4p/3 \rceil \tag{2.8}$$

is valid for the two-edge connected hop-constrained network design problem.

2.3. Double Cut Inequalities

Theorem 4. *[21] Let $L = 3$ and $\Pi = \{V_0^1, V_0^2, V_1, V_2, V_3, V_4\}$ be a partition of V such that $\pi = (V_0^1, V_0^2 \cup V_1, V_2, V_3, V_4)$ induces a 3-st-path-cut, and V_1 induces a valid st-cut in G. If $F \subseteq [V_0^2 \cup V_1 \cup V_4, V_2]$ is chosen such that $|F|$ is odd, then the inequality*

$$x([V_0^1, V_1 \cup V_2 \cup V_3 \cup V_4]) + x([V_0^2, V_1 \cup V_3 \cup V_4]) + x([V_1, V_3 \cup V_4]) + x([V_0^2 \cup V_1 \cup V_4, V_2]) \geq \lceil 3 - |F|/2 \rceil \tag{2.9}$$

is valid for the two-edge connected hop-constrained network design problem.

Since every solution of the TNHNDP is also solution of the TEHNDP, Inequalities (2.8) and (2.9) are also valid for TNHNDP.

3. Further Valid Inequalities

In this section, we introduce several classes of inequalities that are valid for the TNHNDP polytope. As we see in Section 4, they define facets of P(G,L) under some conditions and reinforce the linear relaxation of the integer programming formulation presented in the previous section.

3.1. st-jump Inequalities

Theorem 5. *Let $L = 3$ and $G = (N, E)$ with $|N| \geq 5$ and $(V_0, V_1, \ldots, V_4)$ be a partition of N such that $s \in V_0$ and $t \in V_4$. Then the st-jump inequality*

$$\sum_{i=0}^{2} x([V_i, V_{i+2}]) + \sum_{i=0}^{1} \sum_{j \geq i+3}^{4} 2x([V_i, V_j]) + \sum_{i=0}^{1} x([V_i, V_{i+1} \setminus \{u_{i+1}\}]) + \sum_{i=2}^{3} x([V_i \setminus \{u_i\}, V_{i+1}]) \geq 3 \tag{3.10}$$

,where $u_i \in V_i, i = 1, 2, 3$ is valid for the TNHNDP polytope.

Proof. Let $u \in V_1, w \in V_2, z \in V_3$ and T_1, T_2, T_3 and T_4 be the *L-path-cuts* induced by $(V_0, \{u\}, V_2 \cup V_1 \setminus \{u\}, V_3, V_4)$, $(V_0, V_1, \{w\}, V_3 \cup V_2 \setminus \{w\}, V_4)$, $(V_0, V_1 \cup V_2 \setminus \{w\}, \{w\}, V_3, V_4)$ and $(V_0, V_1, V_2 \cup V_3 \setminus \{z\}, \{z\}, V_4)$, respectively. Then the following inequalities are valid for the TNHNDP:

$$\begin{aligned}
x(T_1) &\geq 2, \\
x(T_2) &\geq 2, \\
x(T_3) &\geq 2, \\
x(T_4) &\geq 2, \\
x(\delta_{G-u}(V_0)) &\geq 1, \\
x(\delta_{G-w}(V_0 \cup V_1)) &\geq 1, \\
x(\delta_{G-z}(V_4)) &\geq 1, \\
x(e) &\geq 0, \quad \text{for all } e \in [V_1 \setminus \{u\}, V_3] \cup [V_2 \setminus \{w\}, V_4] \cup [V_1, V_3 \setminus \{z\}], \\
3x(e) &\geq 0, \quad \text{for all } e \in [V_0, V_1 \setminus \{u\} \cup V_4] \cup [V_1, V_2 \setminus \{w\}] \cup [V_3 \setminus \{z\}, V_4], \\
4x(e) &\geq 0, \quad \text{for all } e \in [V_0 \cup V_2 \setminus \{w\}, V_3] \cup [V_1, V_4].
\end{aligned}$$

By summing these inequalities, dividing the sum by 5, and rounding up the right-hand side, we obtain inequality (3.10). ■

3.2. Rooted Node Partition Inequalities

Theorem 6. *Let $L = 3$ and $T = \{t_1, \ldots, t_p\}$ be a subset of p destination nodes relative to node s. Let $\pi = (V_0, V_1 ..., V_p)$ a partition of $N \setminus \{u\}$ such that $u \in N \setminus \{s\}$, $s \in V_0$, and $t_i \in V_i$, for all $i = 1, \ldots, p$. Then the rooted node partition inequality*

$$x(\delta_{G-u}(V_0, \ldots, V_p)) \geq p \tag{3.11}$$

is valid for the rooted TNHNDP.

Proof. Let u be a node in $N \setminus \{s\}$. $G - u$ is a complete graph after deleting u and its incident edges in G. $x(\delta_{G-u}(V_0, \ldots, V_p)) \geq p$ is valid inequality for the $1ECON(G - u)$ polytope [18]. We can easily say that the $TNHNDP(G - u)$ polytope is inside the $1ECON(G - u)$ polytope. Therefore, $x(\delta_{G-u}(V_0, \ldots, V_p)) \geq p$ is alo valid for the $TNHNDP(G - u)$ polytope and the $TNHNDP(G)$ polytope. ■

If p is 1, then inequality (3.11) is redundant with respect to the *st*-node-cut inequalities.

3.3. Node Double Cut-1

Theorem 7. *Let $\{s_1, t_1\}$ and $\{s_2, t_2\}$ be two demands and $\pi = (V_0, ..., V_4)$ a partition of the graph such that $s_1 \in V_0$, $t_1 \in V_4$, $t_2 \in V_2$ and $s_2 \notin V_2$. Let T be the 3-path-cut induced by $\{s_1, t_1\}$, $\Delta_1 = (T \setminus [V_2, V_4]) \cup ([V_2, V_3] \cup [V_1, V_2])$ and $\Delta_2 = [V_2, V_4]$. Also let $F \subseteq [V_1, V_2]$ such that $|F|$ is odd.*

Then, the inequality

$$x(\Delta_1 \setminus F) + 2x(\Delta_2) \geq 3 - \left\lfloor \frac{|F|}{2} \right\rfloor \tag{3.12}$$

is valid for the TNHNDP.

Proof. Let T be the 3-path-cut of G induced by the partition π and $\{s_1, t_1\}$. As T is a 3-s_1t_1-path-cut, and $s_2 \notin V_2$ and $s_1 \in V_4$, V_2 induces a valid s_2t_2-cut and $V_3 \cup V_4$ induce a valid s_1t_1-cut. Then, the inequalities below are valid for the TNHNDP polytope

$$\begin{aligned}
x(T) &\geq 2, \\
x(\delta(V_2) &\geq 2, \\
x(\delta(V_3 \cup V_4)) &\geq 2, \\
-x(e) &\geq -1, \quad \text{for all } e \in F, \\
x(e) &\geq 0, \quad \text{for all } e \in [V_1, V_2] \setminus F.
\end{aligned}$$

By summing these inequalities, dividing by 2 and rounding up the right hand side, we obtain inequality (3.12). ■

3.4. Node Double Cut-2

Theorem 8. *Let $\{s_1, t_1\}$ and $\{s_2, t_2\}$ be two demands and $\pi_u = (V_0, ..., V_4)$ a partition of $N \setminus \{u\}$, with $u \in N \setminus \{s_1, s_2, t_1, t_2\}$ and $s_1 \in V_0$, $t_1 \in V_4$, $t_2 \in V_2$ and $s_2 \notin V_2$. Let T_u be the 3-path-cut induced by $\{s_1, t_1\}$, $\Delta_1 = (T_u \setminus [V_2, V_4]) \cup ([V_2, V_3] \cup [V_1 \cup \{u\}, V_2])$ and $\Delta_2 = [V_2, V_4]$. Also let $F \subseteq [V_1, V_2]$ such that $|F|$ is odd.*

Then, the inequality

$$x(\Delta_1 \setminus F) + 2x(\Delta_2) \geq 3 - \left\lfloor \frac{|F|}{2} \right\rfloor \tag{3.13}$$

is valid for the TNHNDP.

Proof. Let T be the 3-path-cut of G induced by the partition π and $\{s_1, t_1\}$. As T is a 3-s_1t_1-path-cut, and $s_2 \notin V_2$ and $s_1 \in V_4$, V_2 induces a valid s_2t_2-cut and $V_3 \cup V_4$ induce a valid s_1t_1-cut. Then, the inequalities below are valid for the TNHNDP polytope

$$\begin{aligned} x(T_{G-u}) &\geq 1, \\ x(\delta_{G-u}(V_3 \cup V_4) &\geq 1, \\ x(\delta(V_2)) &\geq 2, \\ -x(e) &\geq -1, \quad \text{for all } e \in F, \\ x(e) &\geq 0, \quad \text{for all } e \in [V_1 \cup \{u\}, V_2] \setminus F. \end{aligned}$$

By summing these inequalities, dividing by 2 and rounding up the right hand side, we obtain inequality (3.13). ■

3.5. Optimality Constraints

In the following theorem, we introduce a new class of inequalities which allows to describe the optimal solutions of the TNHNDP in the case where the demands are rooted and when the edge weights satisfy the triangle inequalities.

Theorem 9. *Suppose that $L = 3$, all the demands are rooted in a node $s \in N$ and the weight of the edges of G satisfy the triangle inequalities. Let $u \in N$ be a steiner node and $F \subseteq E$ an optimal solution of the rooted TNHNDP w.r.t to these edges weights. If $F \cap [s, u] = \emptyset$, then there exists an optimal solution F' such that $F' \cap [s, u] = \emptyset$ and $\delta(u) \cap F' = \emptyset$.*

Proof. Consider that there exists an optimal solution induced by an edge set $F \subseteq E$ such that $F \cap [s, u] = \emptyset$, for some steiner node $u \in N$. If $F \cap \delta(u) = \emptyset$, then $F' = F$ and the result is obvious.
Hence, we consider that, for a given steiner node $u \in N$, $F \cap [s, u] = \emptyset$ and that $F \cap \delta(u) \neq \emptyset$. We will show that there exists an edge set $F' \subseteq E$ obtained from F, which induces a feasible solution for the rooted TNHNDP such that $F' \cap [s, u] =$ and $F' \cap \delta(u) = \emptyset$, and whose weight, $c(F')$, is equal to that of F.

Remark that, in the subgraph induced by any feasible solution of the problem, each terminal belongs to a cycle of length at most 4 and using the source

node s.

If node u does not belong to any of these cycles, then the edge set $F' = F \setminus \{uv\}$ induces a feasible solution with a lower weight, $c(F') < c(F)$. But this contradicts the fact that F is optimal.

Hence, u belongs to at least one of the cycles described above. Let $C_1, ..., C_{p-1}$, $p \geq 2$, denote these cycles. Since cycles C_i, $i = 1, ..., p-1$, share nodes s and u, they are formed by p paths $P_1, ..., P_p$. Let $u_1, ..., u_p$ be the nodes of these paths that are incident to u.

First, suppose that for some $p_0 \in \{1, ..., p-1\}$, the edge $u_{p_0}u_{p_0+1}$ belongs to F. W.l.o.g., we will suppose that, for $i = 1, ..., p_0$, edges u_iu_{i+1} belong to F, and for $i = p_0 + 1, ..., p-1$, edges u_iu_{i+1} do not belong to F. Then, $F' = F \setminus \{uu_1, ..., uu_{p_0}\}$ is feasible for the rooted TNHNDP. This point is obvious since u is not used in any other cycle of length at most 4 and all the terminals still belong to a cycle of length at most 4. Also, it is obvious that $c(F') < c(F)$, and that F is not optimal, a contradiction.

Thus, we suppose now that for all $i \in \{1, ..., p\}$, the edge u_iu_{i+1} does not belong to F. We denote by f_i and g_i edges of the form uu_i and su_i, respectively, and by e_{ij} an edge of the form u_iu_j, for all $i, j \in \{1, ..., p\}$. Recall that, by the triangle inequalities, we have that $c(e_{ij}) \leq c(f_i) + c(f_j)$ and $c(e_{ij}) \leq c(g_i) + c(g_j)$, for all $i = 1, ..., p$. We distinguish two cases.

Case 1. p is even (see Figure 2).

In this case, we let $F' = (F \setminus \bigcup_{i=1}^{p} \{f_i\}) \cup (\bigcup_{i=0}^{\frac{p}{2}-1} \{e_{2i+1,2i+2}\})$. Observe that $F' \cap \delta(u) = \emptyset$. Also, note that F' induces a feasible solution of the rooted TNHNDP since every terminal still belongs to a cycle of length at most 4. Moreover, $c(F') = c(F) + \sum_{i=0}^{\frac{p}{2}-1} [c(e_{2i+1,2i+2}) - c(f_{2i+1}) - c(f_{2i+2})]$, and by the triangle inequalities, we have $c(e_{2i+1,2i+2}) \leq c(f_{2i+1}) + c(f_{2i+2})$, for all $i = 0, ..., \frac{p}{2} - 1$. Therefore, $c(F') \leq c(F)$. As F is an optimal solution for the rooted TNHNDP, we obtain that F' is also optimal. Since $F' \cap [s, u] = \emptyset$ and $F' \cap \delta(u) = \emptyset$, the result holds in the case where p is even.

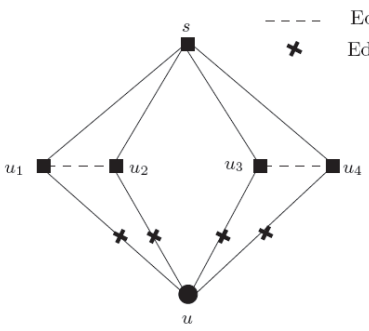

Fig. 2. Case where p is even (here $p = 4$)

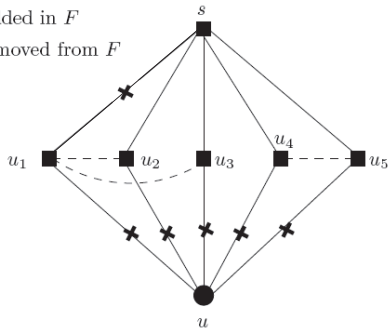

Fig. 3. Case where p is odd (here $p = 5$)

Case 2. p odd (see Figure 3).

Let, in this case,

$F' = (F \setminus (\bigcup_{i=1}^{p}\{f_i\} \cup \{g_1\})) \cup \{e_{12}, e_{13}\} \cup (\bigcup_{i=2}^{\frac{p-1}{2}}\{e_{2i,2i+1}\})$. It is not hard to see that F' induces a feasible solution of the rooted TNHNDP since every terminal belongs to a cycle of length at most 4. Also, we have that

$$c(F') = c(F) + [c(e_{12}) + c(e_{13}) - (c(f_1) + c(f_2) + c(f_3) + c(g_1))] + \sum_{i=2}^{\frac{p-1}{2}}[c(e_{2i,2i+1}) - c(f_{2i}) - c(f_{2i+1}))].$$

In the following, we will show that $c(F') \leq c(F)$, implying that $c(F') = c(F)$. As, by the triangle inequalities, $c(e_{2i,2i+1}) \leq c(f_{2i}) + c(f_{2i+1})$, for all $i = 2, ..., \frac{p-1}{2}$, to show that $c(F') \leq c(F)$, it suffices to show that $c(e_{12}) + c(e_{13}) \leq c(f_1) + c(f_2) + c(f_3) + c(g_1)$.

For this, we will show that $c(f_i) = c(g_i)$, for all $i \in \{1, ..., p\}$. Suppose, on the contrary, that $c(g_i) \neq c(f_i)$, for some $i \in \{1, ..., p\}$. If $c(g_i) < c(f_i)$, then, w.l.o.g., we suppose that $c(g_3) < c(f_3)$. Thus, $c(e_{13}) \leq c(g_1) + c(g_3) < c(g_1) + c(f_3)$. Also, by the triangle inequalities, $c(e_{12}) \leq c(f_1) + c(f_2)$. Therefore, we get $c(F') < c(F)$, a contradiction since F is optimal.
Now if $c(g_i) > c(f_i)$, then we suppose, w.l.o.g., that $c(g_1) > c(f_1)$. In a similar way, we have that $c(e_{12}) \leq c(f_1) + c(f_2) < c(f_2) + c(g_1)$. Also, by the

triangle inequalities, $c(e_{13}) \leq c(f_1) + c(f_3)$. This implies, that $c(F') < c(F)$, a contradiction. Therefore, we have that $c(f_i) = c(g_i)$, for all $i \in \{1, ..., p\}$.

This yields that $c(e_{12}) \leq c(f_1)+c(f_2)$ and $c(e_{13}) \leq c(f_1)+c(f_3) = c(g_1)+c(f_3)$, and hence, $c(F') \leq c(F)$. As F induces an optimal solution for the rooted TNHNDP, we obtain that $c(F') = c(F)$ and F' also induces an optimal solution. Since $F' \cap [s, u] = \emptyset$ and $F' \cap \delta(u) = \emptyset$, the result also holds in the case where p is odd, which ends the proof of the theorem. ■

Corollary 10. *Consider the rooted TNHNDP and let s denotes the root of the demand set. When $L = 3$ and when the weight of the edges satisfy the triangle inequalities, then the incidence vector of an optimal solution of the rooted TNHNDP satisfies*

$$x(su) \geq \sum_{uv \in \delta(u) \setminus \{su\}} x(uv), \text{ for every steiner node } u \in N. \tag{3.14}$$

Inequalities (3.14) are called *optimality constraints*. They are in polynomial number and, as we will see in Section 6, they are effective in solving the TNHNDP in the rooted case.

4. Facets of the TNHNDP Polytope

In this section, we give necessary and sufficient conditions for inequalities (1.1)-(1.6) and the other types of inequalities to be facet defining for $P(G, L)$. This conditions will be used later to derive efficient separation procedures. First, we discuss the dimension of $P(G, 3)$. Let $G = (N, E)$ be a graph. An edge e is essential if e belongs to an st-cut of cardinality 2 or 3-st-path cut of cardinality 2 and st-node cut of cardinality 1 or 3-st-path node cut of cardinality 1.

Theorem 11. *$dim(P(G, 3)) = |E| - |E^*|$, where $|E^*|$ is the set of essential edges.*

In the remainder of this paper, $G = (N, E)$ is a complete and simple graph with $|N| \geq 4$. Thus, $P(G, L)$ is full dimensional. It is clear that, this assumption can be met by adding missing edges with sufficiently high costs in the graph. L-path-cut inequalities can be revealed in the graph when $|N| \geq L+2$. L-path-node-cut inequalities can be revealed in the graph when $|N| \geq L + 3$.

Theorem 12. *Inequality $x(e) \leq 1$ defines a facet of $P(G,3)$ for all $e \in E$.*

Proof. As $|N| \geq 4$ and G is complete, the edge set $E_f = E \setminus \{f\}$ is a solution of TNHNDP, for all $f \in E \setminus \{e\}$. Hence, the sets E and E_f , for all $f \in E \setminus \{e\}$, constitute a set of $|E|$ solutions of the TNHNDP. Moreover, their incidence vectors satisfy $x(e) = 1$ and are affinely independent. ■

Theorem 13. *Inequality $x(e) \geq 0$ defines a facet of $P(G,3)$ if and only if either $|N| \geq 5$ or $|N| = 4$ and $e = uv$ with $\{u,v\} = N \setminus \{s,t\}$.*

Proof. Suppose that $|N| \geq 5$. Then G contains 4 node-disjoint st-paths (an edge of $[s,t]$ and 3 paths of the form $(s,u,t), u \in N \setminus \{s,t\}$). Hence any edge set $E \setminus \{f,g\}, f,g \in E$, contains at least 2 node-disjoint 3-st-paths. The sets $E \setminus \{e\}$ and $E_f = E \setminus \{e,f\}$, for all $f \in E \setminus \{e\}$, constitute a set of $|E|$ solutions of the TNHNDP. Moreover, their incidence vectors satisfy $x(e) = 0$ and are affinely independent.

Now suppose that $|N| = 4$. If $e \neq uv$ with $\{u,v\} = N \setminus \{s,t\}$. then $x(e) \geq 0$ is redundant with respect to the inequalities

$$\begin{array}{rll} x(\delta(s)) & \geq 2 & , \\ -x(f) & \geq -1 & \text{for all } f \in \delta(s) \setminus \{e\}. \end{array}$$

and cannot hence be facet defining. If $e = uv$ with $\{u,v\} = N \setminus \{s,t\}$, then the edge sets $E \setminus \{e\}$ and $E_f, f \in E \setminus \{e\}$, introduced above, are still solutions of TNHNDP. Moreover, their incidence vectors satisfy $x(e) = 0$ and are affinely independent. ■

Theorem 14. *Every st-cut inequality defines a facet of $P(G,3)$.*

Theorem 15. *Every st-node-cut inequality defines a facet of $P(G,3)$ other than those induced by $\{s\}$ or $\{t\}$.*

Proof. Let $G = (N,E)$ be a complete graph. Diarrassouba et al. show that st-cut inequalities for kHNDP (k-edge disjoint hop constrained network design problem) are facets for any $k \geq 1$ [17]. The st-node cut inequality $x(\delta_{G-u}(W)) \geq 1$ is a valid st-cut inequality for $1HNDP(G-u)$, where $u \in N$ and $W \subset N - u$. This implies that $x(\delta_{G-u}(W)) \geq 1$ is a facet for $1HNDP(G-u)$. Since $1HNDP(G-u)$ is full dimensional, there are $|E| - (|N| - 1)$ solutions whose incidence vectors are affinely independent and satisfy $x(\delta_{G-u}) = 1$. Let $x'_1, \ldots, x'_m$ be these solutions, where $m' =$

$|E| - (|N| - 1)$. In each solution, there is exactly one path of length at most 3 between s and t. The system : $\sum_{i=1}^{m'} \lambda_i x'_i = 0$ and $\sum_{i=1}^{m'} \lambda_i = 0$ has a unique solution $\lambda_i = 0$ for $i = 1, \ldots, m'$. Each column of the matrix X of size $m \times m(m = |E|)$ in Figure 4 induce a solution of TNHNDP(G) and its incidence vector satisfies the *st*-node inequality with equality. The entries below x'_1, x'_2 and so on belong to the deleted node u, since the graph is completed, u has $|N| - 1$ edges. The last two rows represent the edges of u adjacent to the source node s and to the destination node t.

$$
\mathrm{X_{mxm}}= \left(\begin{array}{cccc|cccc|cc}
 & & & & \multicolumn{4}{c|}{\overbrace{\qquad\qquad}^{\text{|V|-3}}} & & \\
\mathrm{x_1} & \mathrm{x_2} & \cdots & \mathrm{x_{m'}} & \mathrm{x_{m'}} & \mathrm{x_{m'}} & \cdots & \mathrm{x_{m'}} & \mathrm{x_1} & \mathrm{x_2} \\
1 & 1 & \cdots & 1 & 0 & 1 & \cdots & 1 & 1 & 1 \\
 & & & & 1 & 0 & \cdots & 1 & \vdots & \vdots \\
\vdots & & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & & \\
 & & & & 1 & 1 & \cdots & 0 & 1 & 1 \\
1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 0 & 1 \\
1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 1 & 0
\end{array}\right)
$$

Figure 4

If the system of $\sum_{i=1}^{m} \lambda_i x_i = 0$ and $\sum_{i=1}^{m} \lambda_i x_i = 0$ has a unique solution $\lambda_i = 0$ for $i = 1, \ldots, m$, then there are m affinely independent vectors satisfy the *st*-node cut inequality that concludes it is a facet defining inequality.

$$
\begin{aligned}
\sum_{i=1}^{m'} \lambda_i x'_i + \lambda_{m'+1} x'_m + \lambda_{m'+2} x'_m + \cdots + \lambda_{m-2} x'_m + \lambda_{m-1} x'_1 + \lambda_m x'_1 = 0,& \\
\sum_{i=1}^{m'} \lambda_i + 0 + \lambda_{m'+2} + \cdots + \lambda_m = 0,& \\
\sum_{i=1}^{m'} \lambda_i + \lambda_{m'+1} + 0 + \cdots + \lambda_m = 0,& \\
\cdots& \\
\sum_{i=1}^{m'} \lambda_i + \lambda_{m'+1} + \lambda_{m'+2} + \cdots + 0 = 0,& \\
\sum_{i=1}^{m'} \lambda_i + \lambda_{m'+1} + \lambda_{m'+2} + \cdots + \lambda_m = 0.&
\end{aligned}
$$

The system above can be simplified by algebraic operations as follows: $\sum_{i=1}^{m'} \lambda_i x'_i = 0$, $\sum_{i=1}^{m'} \lambda_i = 0$ and $\lambda_i = 0$ for $i = m'+1, \ldots, m$. We know that the system $\sum_{i=1}^{m'} \lambda_i x'_i = 0$ and $\sum_{i=1}^{m'} \lambda_i = 0$ has a unique solution $\lambda_i = 0$ for $i = 1, \ldots, m'$. Therefore $\lambda_i = 0$ for $i = 1, \ldots, m', m'+1, \ldots, m$. This completes the proof.

■

Theorem 16. *Let $L = 3$ and $G = (N, E)$ be a complete graph with $|N| \geq 5$, st-jump inequality (3.10) defines a facet of $P(G, 3)$ different from a trivial inequality, if one of the following conditions meet*

1. $|V_1| = |V_3| = 1$,
2. $|V_1| = |V_2| = |V_4| = 1$,
3. $|V_0| = |V_2| = |V_3| = 1$.

Proof. Let us denote the inequality (3.10) by $a^T x \geq \alpha$, and let $b^T x \geq \beta$ be a facet defining inequality of $P(G, 3)$ such that

$$\mathcal{F} = \{x \in P(G,3) | a^T x = \alpha\} \subseteq \{x \in P(G,3) | b^T x = \beta\}.$$

We will show that $b = \rho a$ for some $\rho > 0$.

1. Suppose that $|V_1| = |V_3| = 1$.
   Let T be the *st*-jump cut induced by the partition $(V_0, \ldots, V_4)$. Let $a^T x \geq \alpha$ denote the *st*-jump cut inequality produced by T. Let $E_1 = E \setminus T = (\bigcup_{i=0}^{2}(E(V_{2i}))) \cup (\bigcup_{i=0}^{3}[V_i, V_{i+1}]) \setminus ([V_2 \setminus \{v_2\}, V_1 \cup V_3])$, $E_2 = T \setminus ([V_0, V_3 \cup V_4] \cup [V_1, V_4])$ and $E_3 = T \setminus \bigcup_{i=0}^{2}[V_i, V_{i+2}]$. Figure 5 shows the edges of *st*-jump inequality for this case. Let $v_1 \in V_1, v_2 \in V_2$ and $v_3 \in V_3$.
   Let $e_1 = sv_2, e_5 = v_1v_3, e_8 = v_2t$ and $e_2, e_3, e_4, e_6, e_7, e_9, e_{10}, e_{11}$ are any edge of the form $sV_2 \setminus \{v_2\}, V_0 \setminus \{s\}v_2, V_0 \setminus \{s\}V_2 \setminus \{v_2\}, v_1V_2 \setminus \{v_2\}, V_2 \setminus \{v_2\}v_3, v_2V_4 \setminus \{t\}, V_2 \setminus \{v_2\}t, V_2 \setminus \{v_2\}V_4 \setminus \{t\}$, respectively. Notice that $e_1, \ldots, e_{11}$ form all edges of E_2. As the graph G is complete, it is easy to see that the sets given by $F_1 = E_1 \cup \{e_1, e_5, e_8\}, F_2 = E_1 \cup \{e_1, e_5, e_9\}, F_3 = E_1 \cup \{e_1, e_2, e_{10}\}, F_4 = E_1 \cup \{e_1, e_2, e_{11}\}, F_5 = E_1 \cup \{e_1, e_6, e_{10}\}, F_6 = E_1 \cup \{e_2, e_5, e_8\}, F_7 = E_1 \cup \{e_2, e_7, e_8\}, F_8 = E_1 \cup \{e_3, e_5, e_8\}, F_9 = E_1 \cup \{e_4, e_5, e_{10}\}, F_{10} = E_1 \cup \{e_4, e_8, e_{10}\}$ and $F_{11} = E_1 \cup \{e_6, e_8, e_{10}\}$ induce a solution of TNHNDP and their incidence vector satisfies $a^T x \geq \alpha$ with equality. Thus, $b^T x^{F_1} = b^T x^{F_2} = b^T x^{F_3} =$

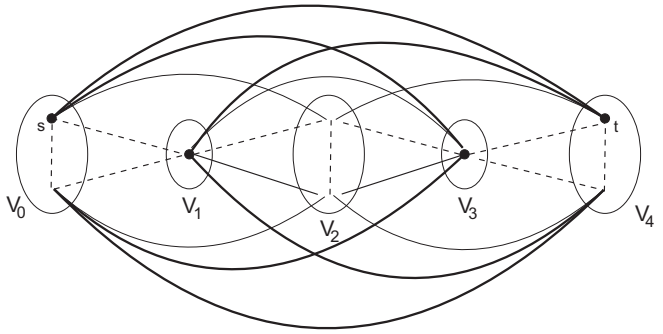

Figure 5: Support graph of a st-jump cut inequality when $|V_1| = |V_3| = 1$.

$b^T x^{F_4} = b^T x^{F_5} = b^T x^{F_6} = b^T x^{F_7} = b^T x^{F_8} = b^T x^{F_9} = b^T x^{F_{10}} = b^T x^{F_{11}}$. As e is an arbitrary edge, we then obtain that

$$b(e) = \rho \quad \text{for all } e \in E_2, \text{ for some } \rho \in \mathbb{R}. \tag{4.15}$$

Consider now $F^* = E_1 \cup \{e_1, e_5, e_8\}$, where $e_1, e_5, e_8 \in E_2$. It is obvious that F^* induces a solution of TNHNDP and its incidence vector satisfies $a^T x \geq \alpha$ with equality. Moreover, $F^{**} = (F^* \setminus \{e_5, e_8\}) \cup \{st\}$ induces a solution of TNHNDP and its incidence vector belongs to $\mathcal{F}$. Thus, $0 = b^T x^{F^*} - b^T x^{F^{**}} = b(e_5) + b(e_8) - b(st)$, implying that $b(st) = 2\rho$ since by (4.15).
Let $F^* = E_1 \cup \{e_1, e\}$, where $e_1 \in E_2$ and $e \in E_3$. It is easy to see that F^* induces a solution of TNHNDP and its incidence vector satisfies $a^T x \geq \alpha$ with equality. Moreover, $F^{**} = (F^* \setminus \{e\}) \cup \{st\}$ induces a solution of TNHNDP and its incidence vector belongs to $\mathcal{F}$. Hence, $0 = b^T x^{F^*} - b^T x^{F^{**}} = b(e) - b(st)$. Since $b(st) = 2\rho$, we then obtain that

$$b(e) = 2\rho \quad \text{for all } e \in E_3. \tag{4.16}$$

Now, we shall show that $b(e) = 0$ for all $e \in E_1$. Suppose first that $e \in [V_0, V_1]$. Consider an edge $f \in [s, w]$ with $w \in V_2$. $F^* = E_1 \cup \{st, f\}$ and $F^* \setminus \{e\}$ induce a solution of TNHNDP and their incidence vector satisfies $a^T x \geq \alpha$ with equality. Thus, $0 = b^T x^{F^*} - b^T x^{F^* \setminus \{e\}} = b(e)$. Similarly, we obtain that $b(e) = 0$ for all $e \in \bigcup_{i=0}^{3} [V_i, V_{i+1}]$.
Consider now an edge $e = uv \in E(V_2)$. Denote by f the edge between v and t. Obviously, $F^* = E_1 \cup \{st, f\}$ and $F^* \setminus \{e\}$ induce

a solution of TNHNDP. As $a^T x^{F^*} = a^T x^{F^* \setminus \{e\}} = \alpha$, we have that $b^T x^{F^*} = b^T x^{F^* \setminus \{e\}} = \alpha$. Thus, we obtain

$$b(e) = 0 \quad \text{for all } e \in E_1. \tag{4.17}$$

From (4.15)-(4.17), we have

$$b(e) = \begin{cases} \rho & \text{for all } e \in E_2, \\ 2\rho & \text{for all}, e \in E_3, \\ 0 & \text{if not.} \end{cases}$$

Since $ax \geq \alpha$ is not a trivial inequality, we have that $\rho > 0$, and therefore, $b = \rho a$.

2. Suppose that $|V_1| = |V_2| = |V_4| = 1$. Let T be the st-jump cut induced by the partition $(V_0, \ldots, V_4)$. Let $a^T \geq \alpha$ denote the st-jump cut inequality produced by T. Let $E_1 = E \setminus T = E(V_0) \cup E(V_3) \cup (\bigcup_{i=0}^{3}[V_i, V_{i+1}]) \setminus ([V_3 \setminus \{v_3\}, V_4])$, $E_2 = T \setminus ([V_0, V_3 \cup V_4] \cup [V_1, V_4])$ and $E_3 = T \setminus \bigcup_{i=0}^{2}[V_i, V_{i+2}]$. Figure 6 shows the edges of st-jump inequality for this case.

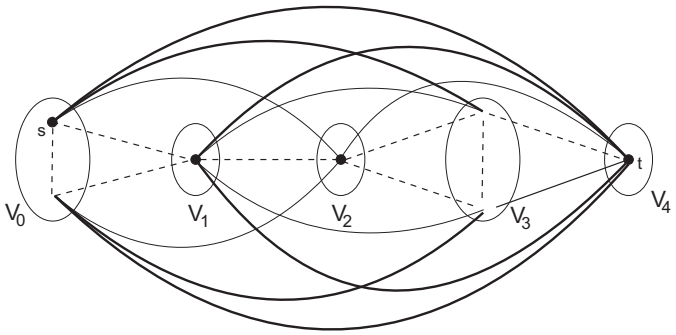

Figure 6: Support graph of a st-jump cut inequality when $|V_1| = |V_2| = |V_4| = 1$.

Let $v_1 \in V_1, v_2 \in V_2$ and $v_3 \in V_3$. Let $e_1 = sv_2, e_2 = v_1v_3, e_3 = v_2t$ and e_5 is any edge of the form $V_0 \setminus \{s\}v_2$, e_4 is any edge of the form $v_1V_3 \setminus \{v_3\}$, e_6 is any edge of the form $V_3 \setminus \{v_3\}t$. Notice that $e_1, e_2, e_3, e_4, e_5, e_6$ form all edges of E_2. As the graph G is complete, it is easy to see that the sets given by $F_1 = E_1 \cup \{e_1, e_4, e_6\}, F_2 = E_1 \cup \{e_1, e_2, e_3\}, F_3 = E_1 \cup \{e_2, e_3, e_5\}, F_4 = E_1 \cup \{e_1, e_2, e_6\}, F_5 = E_1 \cup \{st, e_1\}$ and $F_6 = E_1 \cup \{st, e_2\}$ induce a solution of TNHNDP and its incidence vector

satisfies $a^T x \geq \alpha$ with equality. Thus, $b^T x^{F_1} = b^T x^{F_2} = b^T x^{F_3} = b^T x^{F_4} = b^T x^{F_5} = b^T x^{F_6}$. As e is an arbitrary edge, we then obtain that

$$b(e) = \rho \quad \text{for all } e \in E_2, \text{ for some } \rho \in \mathbb{R}. \tag{4.18}$$

Consider now $F^* = E_1 \cup \{e_1, e_2, e_3\}$, where $e_1, e_2, e_3 \in E_2$. It is obvious that F^* induces a solution of TNHNDP and its incidence vector satisfies $a^T x \geq \alpha$ with equality. Moreover, $F^{**} = (F^* \setminus \{e_1, e_2\}) \cup \{st\}$ induces a solution of TNHNDP and its incidence vector belongs to $\mathcal{F}$. Thus, $0 = b^T x^{F^*} - b^T x^{F^{**}} = b(e_1) + b(e_2) - b(st)$, implying that $b(st) = 2\rho$ since by (4.18).
Let $F^* = E_1 \cup \{e_2, e\}$, where $e_2 \in E_2$ and $e \in E_3$. It is easy to see that F^* induces a solution of TNHNDP and its incidence vector satisfies $a^T x \geq \alpha$ with equality. Moreover, $F^{**} = (F^* \setminus \{e\}) \cup \{st\}$ induces a solution of TNHNDP and its incidence vector belongs to $\mathcal{F}$. Hence, $0 = b^T x^{F^*} - b^T x^{F^{**}} = b(e) - b(st)$. Since $b(st) = 2\rho$, we then obtain that

$$b(e) = 2\rho \quad \text{for all } e \in E_3. \tag{4.19}$$

Now, we shall show that $b(e) = 0$ for all $e \in E_1$. Suppose first that $e \in [V_0, V_1]$. Consider an edge $f \in [s, w]$ with $w \in V_2$. $F^* = E_1 \cup \{st, f\}$ and $F^* \setminus \{e\}$ induce a solution of TNHNDP and their incidence vector satisfies $a^T x \geq \alpha$ with equality. Thus, $0 = b^T x^{F^*} - b^T x^{F^* \setminus \{e\}} = b(e)$. Similarly, we obtain that $b(e) = 0$ for all $e \in \bigcup_{i=0}^{3} [V_i, V_{i+1}]$.
Consider now an edge $e = uv \in E(V_0)$. Denote by f the edge between v and t. Obviously, $F^* = E_1 \cup \{st, f\}$ and $F^* \setminus \{e\}$ induce a solution of TNHNDP. As $a^T x^{F^*} = a^T x^{F^* \setminus \{e\}} = \alpha$, we have that $b^T x^{F^*} = b^T x^{F^* \setminus \{e\}} = \alpha$. Thus, we obtain

$$b(e) = 0 \quad \text{for all } e \in E_1. \tag{4.20}$$

From (4.18)-(4.20), we have

$$b(e) = \begin{cases} \rho & \text{for all } e \in E_2, \\ 2\rho & \text{for all}, e \in E_3, \\ 0 & \text{if not.} \end{cases}$$

Since $ax \geq \alpha$ is not a trivial inequality, we have that $\rho > 0$, and therefore, $b = \rho a$.

3. This case is the symmetric of the case 2. Hence, the proof follows by symmetry. Figure 7 shows the edges of *st*-jump inequality for this case.

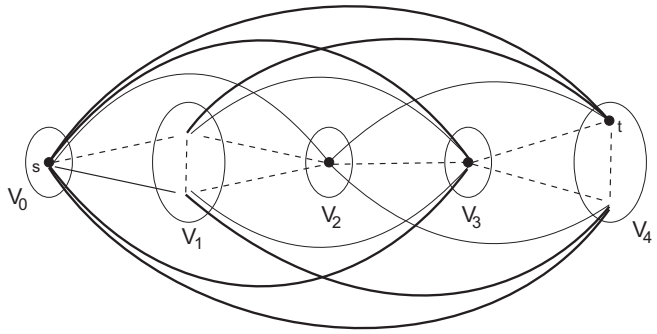

Figure 7: Support graph of a *st*-jump cut inequality when $|V_0| = |V_2| = |V_3| = 1$.

■

Theorem 17. *Every L-path-cut inequality defines a facet of* $P(G, L)$ *if and only if* $|V_0| = 1$ *and* $|V_{L+1}| = 1$.

Proof. *Necessity:* We will show that $x(T) \geq 2$ does not define a facet if $|V_0| \geq 2$. The case where $|V_{L+1}| \geq 2$ follows by symmetry. Suppose that $|V_0| \geq 2$ and consider the partition $(V'_0, \ldots, V'_{L+1})$ given by

$$\begin{aligned} V'_0 &= \{s\}, \\ V'_1 &= V_1 \cup (V_0 \backslash \{s\}), \\ V'_i &= V_i, \quad i = 2, \ldots, L+1. \end{aligned}$$

This partition induces the L-path-cut inequality $x(T') \geq 2$, where $T' = T \backslash [V_0 \backslash \{s\}, V_2]$. Since G is complete, $[V_0 \backslash \{s\}, V_2] \neq \emptyset$ and T' is strictly contained in T. Thus, $x(T) \geq 2$ is redundant with respect to the inequalities

$$\begin{aligned} x(T') &\geq 2 \\ x(e) &\geq 0, \quad \text{for all } e \in [V_0 \backslash \{s\}, V_2], \end{aligned}$$

and hence cannot define a facet of $P(G, L)$.

Sufficiency: Suppose that $|V_0| = |V_{L+1}| = 1$ such that $V_0 = \{s\}$ and $V_{L+1} = \{t\}$. Let us denote the inequality (1.3) by $a^T x \geq \alpha$, and let $b^T x \geq \beta$ be a facet defining inequality of $P(G, L)$ such that

$$\{x \in P(G, L) | a^T x = \alpha\} \subseteq \{x \in P(G, L) | b^T x = \beta\}.$$

We will show that $b = \rho a$ for some $\rho > 0$.

Let T be the L-path-cut induced by the partition $(V_0, \ldots, V_{L+1})$. Let $a^T \geq \alpha$ denote the L-path-cut inequality produced by T. Let $E' = E \setminus T = (\bigcup_{i=1}^{L}(E(V_i))) \cup (\bigcup_{i=0}^{L}[V_i, V_{i+1}])$. Let $f \in [s,t]$ and $T_f = T \backslash \{f\}$. As the graph G is complete, it is easy to see that the sets given by

$$F_e = E' \cup \{st, e\} \text{ for all } e \in T_f$$

induce solutions of the TNHNDP whose incidence vectors satisfy $a^T x \geq \alpha$ with equality. Hence, we have $b^T x^{F_e} - b^T x^{F_{e'}} = b(e) - b(e') = 0$ for all $e, e' \in T_f$. Thus,

$$b(e) = b(e') \quad \text{for all } e, e' \in T_f. \tag{4.21}$$

Now let $g \in [V_0, V_L]$, $g' \in [V_1, V_{L+1}]$, and $F^* = E' \cup \{g, g'\}$. Clearly, F^* induces a solution whose incidence vector satisfies $a^T x \geq \alpha$ with equality. We have that $b^T x^{F^*} - b^T x^{F_g} = b(g') - b(f) = 0$. This together with (4.21) yields

$$b(e) = \rho \quad \text{for all } e \in T, \text{ for some } \rho \in \mathbb{R}. \tag{4.22}$$

Now, we shall show that $b(e) = 0$ for all $e \in E'$. Suppose first that $e \in [V_0, V_1]$. Consider an edge $h \in [s, w]$ with $w \in V_2$. $F_h \backslash \{e\}$, where F_h is as defined above. It is obvious that $F_h \backslash \{e\}$ still induces a solution of TNHNDP and its incidence vector satisfies $a^T x \geq \alpha$ with equality. Thus, $0 = b^T x^{F_h} - b^T x^{F_h \backslash \{e\}} = b(e)$. Similarly, we obtain that $b(e) = 0$ for all $e \in \bigcup_{i=0}^{L}[V_i, V_{i+1}]$.

Consider now an edge $e \in E(V_i),\ i \in \{1, \ldots, L\}$. Let $v \in V_L$ and $h' \in [s, v]$. Obviously, $F_{h'} \setminus \{e\}$ induces a solution of TNHNDP. As $a^T x^{F_{h'}} = a^T x^{F_{h'} \backslash \{e\}} = \alpha$, we have that $b^T x^{F_{h'}} = b^T x^{F_{h'} \backslash \{e\}} = \alpha$. Thus, we obtain

$$b(e) = 0 \quad \text{for all } e \in E'. \tag{4.23}$$

From (4.22)-(4.23), we have

$$b(e) = \begin{cases} \rho & \text{for all } e \in T, \\ 0 & \text{if not.} \end{cases}$$

Since $ax \geq \alpha$ is not a trivial inequality, we have that $\rho > 0$, and therefore, $b = \rho a$.

■

Theorem 18. *Every L-path-node-cut inequality defines a facet of $P(G, L)$.*

Proof. The proof can be done in a similar way with the proof of theorem 15. ■

Theorem 19. *The rooted-partition inequality 2.8 defines a facet of $P(G, L)$ if $|V_i| = 1$ for all $i = 1, \ldots, p$ and p is not a multiple of 3.*

Proof. *Sufficiency:* Suppose that $|V_i| = 1$ for all $i = 1, \ldots, p$ and p is not a multiple of 3. Let us denote the inequality (2.8) by $a^T x \geq \alpha$, and let $b^T x \geq \beta$ be a facet defining inequality of $P(G, 3)$ such that

$$\{x \in P(G,3) | a^T x = \alpha\} \subseteq \{x \in P(G,3) | b^T x = \beta\}.$$

We will show that $b = \rho a$ for some $\rho > 0$.

Let $p = 3q + r$, where q is a positive integer and r is either 1 or 2. Consider the edge sets
$F_0 = \bigcup_{i=0}^{q-1} \{st_{3i+1}, t_{3i+1}t_{3i+2}, t_{3i+2}t_{3i+3}, t_{3i+3}s\} \bigcup \{st_{3q+1}, t_{3q+r}t_1\} \bigcup_{i=1}^{r-1} \{t_{3q+i}t_{3q+i+1}\} \bigcup E(V_0)$,
$F_e = (F_0 \setminus \{t_p t_1\}) \cup \{e\}$, for all $e \in \delta(t_p) \setminus \{st_p\}$.
$F'_e = (F_0 \setminus \{t_p t_1\}) \cup \{e\}$, for some $e \in \delta(t_p)$.

If $p = 3q + 1$, it is easy to see that F_0 and F_e induce a solution whose incidence vector satisfies $a^T x \geq \alpha$ with equality. Hence, we have $b^T x^{F_0} - b^T x^{F_e} = b(e) - b(t_1 t_p) = 0$ for all $e \in \delta(t_p) \setminus \{st_p\}$. Thus,

$$b(e) = b(t_1 t_p) \quad \text{for all } e \in \delta(t_p) \setminus \{st_p\}. \tag{4.24}$$

Furthermore, let $F'_0 = (F_0 \setminus \{st_p\}) \cup \{t_p t_{p-1}\}$. Clearly, F'_0 induces a solution of TNHNDP and its incidence vector satisfies $a^T x \geq \alpha$ with equality. As $b^T x^{F'_0} = \beta$, we obtain that $b(st_p) = b(t_1 t_{p-1})$, This together with (4.24) yields

$$b(e) = b(e') \quad \text{for all } e, e' \in \delta(t_p).$$

If $p = 3q+2$, it is easy to see that F_0 and F'_e induce a solution whose incidence vector satisfies $a^T x \geq \alpha$ with equality. Hence, we have $b^T x^{F_0} - b^T x^{F'_e} = b(e) - b(t_1 t_p) = 0$ for all $e \in \delta(t_p)$. Thus,

$$b(e) = b(e') \quad \text{for all } e, e' \in \delta(t_p).$$

By exchanging the role of t_p and t_i's, $i \neq p$, we get

$$b(e) = \rho \quad \text{for all } e \in \delta(V_0, \ldots, V_p), \text{ for some } \rho \in \mathbb{R}.$$

Now assume that $E(V_0) \neq \emptyset$ and let $e \in E(V_0)$. It is obvious that $F_0 \backslash \{e\}$ still induces a solution of TNHNDP and its incidence vector satisfies $a^T x \geq \alpha$ with equality. Thus, $0 = b^T x^{F_0} - b^T x^{F_0 \backslash \{e\}} = b(e)$. Consequently, we have

$$b(e) = \begin{cases} \rho & \text{for all } \delta(V_0, \ldots, V_p), \\ 0 & \text{for all } E(V_0). \end{cases}$$

Since $ax \geq \alpha$ is not a trivial inequality, we have that $\rho > 0$, and therefore, $b = \rho a$.

■

5. Branch-and-Cut Algorithm

In this section we devise a Branch-and-Cut algorithm for solving the TNHNDP. In order to solve the linear relaxation of the natural formulation of the TNHNDP, we need to address the separation problem associated with the basic inequalities (1.1)-(1.4), since they are in exponential number. One may also add further valid inequalities in the Branch-and-Cut algorithm in order to reinforce the linear relaxation of the natural formulation. To do this, we consider Double Cut and Rooted Partition inequalities (2.9) and (2.8), as well as the Optimality constraints (3.14). Notice that constraints (3.14) are in polynomial number. Thus, we add all these constraints in the linear program associated with the linear relaxation.

In the following, we discuss the separation problem associated with the basic inequalities (1.1)-(1.4), the Double Cut and Rooted Partition inequalities.

Another important issue in a Branch-and-Cut algorithm is to determine a good upper bound from a fractional solution of the problem. For this we devise a primal heuristic for the problem. This point is also discussed in the following.

5.1. Separation procedures

5.1.1. Separation procedures for the basic inequalities

To separate *st*-cut, *st*-node-cut, *L*-path-cut and *L*-path-node-cut inequalities (1.1)-(1.4), we consider the following argument from [17]. Consider an undirected graph $G = (N, E)$. Let $V = N \setminus \{s, t\}, V'$ be a copy of V and $\tilde{N} = V \cup V' \cup \{s, t\}$. The copy of a node $u \in V$ will be denoted by u' in V'.

Let $\tilde{G} = (\tilde{N}, \tilde{A})$ be the directed graph such that $\tilde{N} = V \cup V' \cup \{s, t\}$ and $\tilde{A}$ is obtained from as follows. For the edge between s and t, we associate an arc from s to t in $\tilde{G}$. To each edge $su \in E$ (resp. $vt \in E$), we associate in $\tilde{G}$ the arc $(s, u), u \in V$ (resp. $(v', t), v' \in V'$). To each edge $uv \in E$, with $u, v \notin \{s, t\}$, we associate two arcs (u, v') and (v, u'), with $u, v \in V$ and $u', v' \in V'$. Finally, to each node $u \in N \setminus \{s, t\}$, we associate an arc (u, u') in $\tilde{G}$. (see Figure 8 for an illustration).

Note that any st-dipath in $\tilde{G}$ is of length no more than 3. Also note that each 3-st-path in G corresponds to an st-dipath in $\tilde{G}$ and vice versa. All the separation procedures are performed on the directed graphs obtained using the above transformation for every $\{s, t\} \in D$.

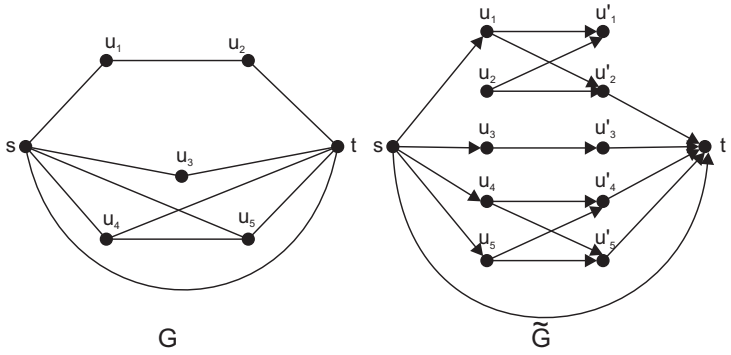

Figure 8: Construction of the supported directed graph.

The separation of the st-cut inequalities (1.1) and L-path-cut inequalities (1.3) can be performed by computing minimum weight st-dicut in $\tilde{G}_{st}$ for every $\{s, t\} \in D$. By minimum cut-maximum relationship, computing a minimum weight st-dicut of $\tilde{G}_{st}$ is equivalent to computing a maximum flow separating s and t. The maximum flow computations are handled by the Edmonds-Karp algorithm [20] that runs in $O(m^2 n)$ time where m and n are the number of edges and nodes in $\tilde{G}_{st}$, respectively. Since this operation is repeated $|D|$ times, the whole algorithm runs in $O(m^2 n|D|)$, and hence is polynomial time.

The separation of the st-node-cut inequalities (1.2) and L-path-node-cut inequalities (1.4) can be performed by deleting the edges one by one in the graph and using the procedure explained above.

5.1.2. Separation procedure for the Double cut inequalities

To separate the Double Cut inequalities (2.9), we use the separation heuristic developed by [21]. Note that the complexity of the separation problem of these inequalities is still an open question.

5.1.3. Separation procedure for Rooted Partition inequalities

Now, we discuss the separation problem of the Rooted Partition inequalities (2.8). Recall that a rooted partition inequality is induced by a set of p terminals $T = \{t_1, ..., t_p\}$ which are destinations relatively to a source node s and a partition $\pi = (V_0, V_1, ..., V_p)$ such that $s \in V_0$ and $t_i \in V_i$, $i = 1, ..., p$.

In the following we will describe a separation algorithm for these inequalities when $L = 3$ and $|V_i| = 1$, for all $i \in \{1, ..., p\}$. To do this, we will reduce the separation problem in this case to finding a maximum flow with lower bounds in a special graph.

First, we can easily see that Inequality (2.8) is equivalent to

$$x(\delta(V_0)) + \sum_{u \in N \setminus V_0} x(\delta(u)) \geq \frac{8}{3}p + \alpha \tag{5.25}$$

with

$$\alpha = \begin{cases} \frac{4}{3} & \text{if } p = 3q + 1, \text{ for some } q \in \mathbb{N} \\ \frac{3}{3} & \\ \frac{2}{3} & \text{if } p = 3q + 2, \text{ for some } q \in \mathbb{N}. \end{cases}$$

Since $p = |N \setminus V_0|$ and $x(\delta(V_0)) = x(\delta(N \setminus V_0))$, Inequality (5.25) is equivalent to

$$x(\delta(N \setminus V_0)) \geq \sum_{u \in N \setminus V_0} y_u + \alpha \tag{5.26}$$

where $y_u = \frac{8}{3} - x(\delta(u))$, for all $u \in N \setminus V_0$.

Now we describe our separation algorithm. We start by contracting with node s all the steiner nodes, every terminal u such that $\{s, u\} \notin D$, and every terminal u such that $\{s, u\} \in D$ and $y_u < 0$.

In order to find violated Rooted Partition inequalities, we look for violated inequalities (5.26) by looking for feasible flow with lower bounds in a directed graph $\widetilde{G} = (N, \widetilde{A})$. This graph is as follows.

For every edge $uv \in E$ with $\overline{x}(uv) > 0$, we add two arcs (u, v) and (v, u) in $\widetilde{G}$ with capacities $(0, \overline{x}(uv))$. For every node $u \in N \setminus \{s\}$, we add an arc (s, u) in $\widetilde{G}$ with capacities $(y_u, +\infty)$. Finally we add an arc (s, u_0) for some node $u_0 \neq s$, with capacities $(0, +\infty)$. This latter arc transforms the feasible flow problem into a feasible circulation problem (see [22] for more details). For each arc a of $\widetilde{G}$, we denote by L_a and U_a the lower and upper bounds of the flow on this arc.

Since the flows in the graph are subject to lower bounds, a flow from a terminal, say $t_1 \in N \setminus \{s\}$ to s in $\widetilde{G}$ may not be feasible with respect to these lower bounds. We have two cases.

Case 1. If there is no feasible flow from t_1 to s, then there exists a node set $W \subseteq N$ in $\widetilde{G}$ with $W = \{t_1, t_2, ..., t_p\}$, $s \in N \setminus W$ and $\sum_{a \in \delta^+_{\widetilde{G}}(W)} U_a < \sum_{a \in \delta^-_{\widetilde{G}(W)}} L_a$.

It is not hard to see that

$$\sum_{a \in \delta^+_{\widetilde{G}(W)}} U_a = \sum_{e \in \delta_G(W)} \overline{x}(e) \text{ and } \sum_{a \in \delta^-_{\widetilde{G}(W)}} L_a = \sum_{t \in W} \overline{y}_t.$$

From W, we built a rooted partition $\pi = (V_0, V_1, ..., V_p)$ with $V_0 = N \setminus W$, $V_i = \{t_i\}$, $i = 1, ..., p$. Obviously, the inequality (5.26), and hence the rooted partition inequality, induced by π is violated by $\overline{x}$.

Case 2. If there exists a feasible flow from t_1 to s, then we compute a maximum feasible flow by augmenting as much as possible the current feasible flow from t_1 to s. If $W = \{t_1, ..., t_p\}$, with $p \geq 1$, denotes a node set obtained from this maximum flow, then we built the partition π as in the previous case and check if Inequality (5.26) is violated or not. If it violated, then we have found a violated rooted partition inequality. If it is not violated, then we choose another terminal node, say t_2, and compute a maximum feasible flow. We repeat this procedure until we find a violated rooted partition inequality or we have selected all the terminals.

This procedure is heuristic and runs in polynomial time since the maximum feasible flow problem is solved in polynomial time.

5.2. Primal Heuristic

An important issue in the efficiency of the Branch-and-Cut algorithm is to compute a good upper bound at each node of the Branch-and-Cut tree. We improve a primal heuristic described in Algorithm 1 to find good upper bounds. We first round up to 1 all the variables that are smaller than a threshold value (0.9 in the beginning), otherwise round down to 0. Then, we check whether the resulting integer solution is feasible as follows. In the supported directed graph $\tilde{G}$ for every demand, we calculate the node-disjoint paths between the source and the destination. If the number of node-disjoint paths is greater than 1 for every demand, then the integer solution is feasible. If not, we decrease the threshold value by 0.5 and apply the same procedure for the initial fractional solution. After finding a feasible solution, we try to improve this solution by deleting all edges which are not used in any path.

6. Computational Results

Based upon the previous theoretical results, we have developed a branch-and-cut algorithm to solve efficiently the TNHNDP when $L = 3$. The algorithm has been implemented in C++, using ABACUS to manage the branching tree and CPLEX 12.2 as the linear solver. It was tested on a Xeon Quad-Core E5507 machine at 2.27 GHz with 8GB RAM, running under Linux. The maximum CPU run time has been fixed to 5 hours. The test problems are complete graphs with Euclidian edge costs taken from TSPLIB [19] and complete graphs randomly generated on a 250×250 grid. The demands used in these tests are randomly generated. Each set of demand is either rooted at the same node s or arbitrary having multiple sources and destinations. The random graphs were generated with the same size as TSPLIB graphs. For each random problem, five instances were tested and the results given are rounded up averages.

If an optimal solution x^* of the linear relaxation of the TNHNDP is not feasible, the Branch-and-Cut algorithm generates additional valid inequalities of $P(G, 3)$ violated by x^*. Their separations are performed in the following order:

1. *st*-cut inequalities,
2. L-path-cut inequalities,
3. *st*-node-cut inequalities,
4. L-path-node-cut inequalities,

Algorithm 1: Primal Heuristic Algorithm for TNHNDP

Data: An undirected graph $G = (N, E)$, demands $D \subseteq N \times N$, a fractional solution x

Result: An Upper Bound Up

begin
- $\alpha \leftarrow 0.9$;
- round up to 1 every variable corresponding to an edge with $x(e) > \alpha$ in the fractional solution x;
- round down to 0 every variable corresponding to an edge with $x(e) \leq \alpha$;
- **if** *the resultant solution is not feasible* **then**
  - $\alpha \leftarrow \alpha - 0.05$;
  - go to Step 3;
- build the directed graph $\widetilde{G} = (\widetilde{N}, \widetilde{A})$ from $G(x)$ and give capacity 1 for each arc;
- **for** *each demand* $\{s, t\} \in D$ **do**
  - find 2-node disjoint paths between s and t in $\widetilde{G}$;
  - **for** *all* $a \in \widetilde{A}$ **do**
    - **if** *a is used in the 2-node disjoint paths* **then**
      - increment $counter(e)$, where e is the corresponding edge;
- $Up \leftarrow$ sum up the costs of edges $e \in G$ with $counter(e) \geq 1$;
- return Up.

end

5. Rooted Partition inequalities.
6. double cut inequalities.

We note that all inequalities are global (i.e., valid at every node of the branch-and-cut tree) and several of them can be added at each iteration. Furthermore, we go to the next class only if we do not find any violated inequalities in the current class. In order to store the generated inequalities, we use a dynamic pool. Inequalities in the pool can be removed from the current linear program when they are not active. Also, they are the first inequalities to be separated. If all the inequalities in the pool are satisfied by the current solution, we then separate the classes of inequalities according to the order given above.

Abbreviations used in tables are:

$\|N\|$	:	the number of nodes of the problem,
$\|D\|$	:	the number of demands,
NC	:	the number of generated st-cut inequalities
NNC	:	the number of generated st-node-cut inequalities
LPC	:	the number of generated L-path-cut inequalities
LPNC	:	the number of generated L-path-node-cut inequalities
RP	:	the number of generated rooted-partition inequalities
RNP	:	the number of generated rooted node partition inequalities
DC	:	the number of generated double cut inequalities
SP	:	the number of subproblems in the branch-and-cut tree
Gap1	:	the relative error between the best upper bound and the lower bound obtained at the root node of the branch-and-cut tree
Gap2	:	the relative error between the best upper bound and the best lower bound of the branch-and-cut tree
COpt	:	optimum solution (value of the best upper bound)
CPU	:	total time in hours:min.sec.

Table 1 presents the results obtained for instances with graphs having up to 96 nodes. For each instance, we give the type of the demand set ("a" for arbitrary, "r" for rooted). Remark that a value of 0 for Gap2 indicates that COpt is the optimal solution of the problem, 15 instances over 28 has been solved to optimality. For the instances solved to optimality, the CPU time varies from 1 sec to 1h36min. We notice that a small number of double cut

Table 1: Results for TSPLIB instances when $L = 3$

$\|N\|$	$\|D\|$	NC	NNC	LPC	LPNC	RP	DC	Copt	Gap1	Gap2	SP	CPU
a 10	5	9	2	188	1	0	15	3500	9.48	0	49	0:00:01
r 10	5	12	4	12	0	1	0	2387	0.77	0	5	0:00:01
r 10	7	4	4	42	0	2	0	3022	1.26	0	3	0:00:01
a 14	10	45	13	5978	517	0	23	4421	14.7	0	563	0:01:19
a 14	7	4	0	191	0	0	0	3938	2.14	0	9	0:00:01
r 14	10	37	15	257	15	4	0	4085	3.58	0	47	0:00:01
r 14	7	17	7	56	6	1	0	3237	3.44	0	21	0:00:01
a 17	45	475	72	43186	138487	0	43	2807	24.5	5.46	1639	5:00:00
a 17	8	32	6	17330	694	0	80	3233	14.18	0	1801	0:14:01
r 17	16	74	33	9522	1659	52	0	3051	8.22	0	1915	0:07:36
a 30	10	78	2	3108	35	0	0	6307	6.19	0	97	0:00:52
a 30	15	19	2	52744	13	0	0	13882	43.97	42.91	101	5:00:00
r 30	10	318	751	509	72	8	0	5002	6.38	0	443	0:01:17
r 30	15	778	1435	14098	1726	92	0	7161	11.53	0	7411	1:30:53
a 48	10	111	18	76852	18	0	0	17824	42.14	40.01	13	5:00:00
a 48	15	68	0	28895	0	0	0	33525	58.9	58.79	7	5:00:00
r 48	10	2425	1411	641	139	18	0	8546	8.1	0	387	1:05:13
r 48	15	631	897	4001	601	55	0	10029	9.62	0	1649	0:43:42
a 52	10	202	4	41755	11	0	0	7268	19.15	15.87	35	5:00:00
a 52	20	57	2	73270	0	0	0	21250	56.35	56.11	35	5:00:00
r 52	10	1188	2583	562	81	8	0	5002	6.38	0	451	1:36:51
r 58	20	183	215	27780	3125	91	0	35582	22.27	16.5	2339	5:00:00
r 58	30	69	31	21317	284	38	0	53144	41.42	38.9	823	5:00:00
r 58	40	244	9	23881	127	23	0	122549	71.88	70.76	215	5:00:00
r 76	20	26	46	14097	2288	37	0	551	27.72	23.68	1635	5:00:00
r 76	40	14	1	12171	116	10	0	1337	53.88	53.01	305	5:00:00
r 96	20	110	4	15828	272	12	0	47389	52.19	48.57	405	5:00:00
r 96	40	63	0	10227	1	28	0	96355	56.71	55.77	121	5:00:00

inequalities are generated and for almost all the instances, the rooted partition inequalities are generated. Also, we notice that for the instances which are not solved to optimality, the gaps (Gap1 and Gap2) are relatively high while the number of nodes in the Branch-and-Cut tree are relatively small (see for example "r-96-40"). Also, we notice that a large number of basic inequalities are generated. This let us suppose that the algorithm spents a lot of time in the separation of the different inequalities and does not have enough time to explore more solutions in the Branch-and-Cut tree.

The observations are also the same for random instances (Table 2). Almost 37% of the instances are solved to optimality. For several instances, the gaps (Gap1 and Gap2) are relatively important while the number of nodes in the Branch-and-Cut tree are relatively small (see for example "r-96-40"). Also, a large number of basic inequalities are generated during the resolution of each of these instances. In fact, one can observe that the randomly

Table 2: Results for random instances when $L = 3$

$\|N\|$	$\|D\|$	NC	NNC	LPC	LPNC	RP	DC	Copt	Gap1	Gap2	SP	CPU
a 10	5	6	1	333	16	0	13	984	10	0	81	0:00:01
r 10	5	7	4	19	1	1	1	701	4	0	12	0:00:01
r 10	7	10	9	90	13	2	0	878	40	0	59	0:00:01
a 14	10	49	21	11046	1380	0	34	1252	19	0	1436	0:04:02
a 14	7	10	4	8638	762	0	70	1325	14	0	2005	0:04:44
r 14	10	5	9	603	143	10	3	1256	9	0	341	0:00:05
r 14	7	11	13	180	37	2	5	1055	9	0	122	0:00:01
a 17	45	529	31	90540	31007	0	310	1238	31	8	8018	4:02:31
a 17	8	13	1	63937	91	0	10	1452	21	14	361	3:20:31
r 17	16	29	46	28895	4470	153	0	1384	12	2	16335	3:21:23
a 30	10	39	0	42109	17955	0	2	2611	42	40	67	5:00:00
a 30	15	10	0	35577	8031	0	0	3922	49	48	98	5:00:00
r 30	10	132	108	866	172	13	0	1122	10	0	437	0:00:43
r 30	15	66	140	20527	4175	132	0	1452	12	0	14389	2:07:12
a 48	10	85	1	35143	8705	0	0	2071	34	33	35	5:00:00
a 48	15	55	0	42698	2	0	0	6001	59	58	15	5:00:00
r 48	10	219	96	1314	331	10	0	1171	7	0	802	0:06:44
r 48	15	48	227	27376	6496	156	0	1509	15	4	13431	5:00:00
a 52	10	103	0	44476	3	0	0	3042	54	53	9	5:00:00
a 52	20	48	0	49967	0	0	0	10073	77	77	4	5:00:00
r 52	10	1177	277	698	100	14	0	1077	10	0	341	0:08:47
r 52	20	142	139	28072	2455	75	0	1830	23	16	3886	5:00:00
r 58	20	76	31	19821	1547	89	0	2051	27	22	3548	5:00:00
r 58	30	86	5	13881	280	20	0	4041	51	50	1000	5:00:00
r 58	40	134	3	13080	109	7	0	5006	54	53	348	5:00:00
r 76	20	137	147	13457	957	41	0	2060	29	25	1044	5:00:00
r 76	40	141	15	9709	160	19	0	5432	60	59	208	5:00:00
r 96	20	219	10	9309	333	29	0	171270	40	36	612	5:00:00
r 96	40	113	13	10604	49	23	0	440184	62	61	148	5:00:00

generated instances are as difficult as the TSPLIB instances.

We have also checked the efficiency of the different constraints used in the Branch-and-Cut algorithm, especially the rooted partition inequalities and the optimality constraints. For this, we have tried to solve, for both random and TSPLIB instances, the problem without rooted partition inequalities on the one hand and without optimality constraint in the other hand (keeping rooted partition inequalities). For each constraint, we remark that the efficiency of the algorithm is significantly decreased, with an increase of the CPU time and the number of nodes in the Branch-and-Cut tree. Moreover, for some instances, we do not have the optimal solution when one of these constraints are removed while the algorithm is able to obtain the optimal solution when they are included.

7. Conclusion

In this work, we have studied the two-node connected hop-constrained network design problem (TNHNDP). We have presented an integer programming formulation for the TNHNDP when $L = 3$, and study the polytope associated with this formulation. We have then introduced several new classes of inequalities and study the conditions under which these inequalities define facets.

We have discussed the separation problem of the basic inequalities, the Rooted Partition and the Double Cut inequalities based on the polyhedral results, and devised a Branch-and-Cut algorithm for the problem when $L = 3$. The computational results presented in this paper show that the Branch-and-Cut algorithm is effective in solving the problem when $L = 3$, even if some improvments are needed in the separation of the basic inequalities, especially for instances with a large sized graphs and with a large number of demands. The results also shows that the rooted partition and optimality constraints are very effective in the solving the problem.

The polyedral and compuational study realized in this paper raise several questions. First, it would be interesting to further investigate the polytope of the problem in order to find more facets especially when the demand set contains multiple sources. Also, the separation of the different inequalities may be improved, in particular for the basic inequalities. Finally, the case where $L = 4$ (and $L > 3$ in general) should also be addressed since in telecommunication networks, the QoS-constraint may be be set up by paths with more than 3 hops.

8. Acknowledgments

The research of the second author has been supported by a TUBITAK-BIDEB fellowship.

[1] G. Dahl and L. Gouveia, On the directed hop-constrained shortest path problem, Oper Res Lett 32 (2004), 15-22.

[2] D. Huygens and A.R. Mahjoub, Integer programming formulations for the two 4-hop-constarined paths problem, Networks 49 (2007), 135-144

[3] L. Gouveia, P. Patricio, and A. de Sousa, Compact models for hop-constrained node survivable network design, an application to MPLS, Telecommunications planning: Innovations in pricing, network design and management, G. Anandalingam and S. Raghavan (Editors), Vol. 33, Springer, Berlin, 2005, pp. 167-180.

[4] D. Huygens, A.R. Mahjoub, and P. Pesneau, Two edge-disjoint hop-constrained paths and polyhedra, SIAM J Disc Math 18 (2004), 287-312.

[5] G. Dahl, The 2-hop spanning tree problem, Oper Res Lett 23 (1998), 21-26.

[6] G. Dahl, Notes on polyhedra associated with hop-constrained paths, Oper Res Lett 25 (1999), 97-100.

[7] G. Dahl, N. Foldnes, and L. Gouveia, A note on hopconstrained walk polytopes, Oper Res Lett 32 (2004), 345-349.

[8] G. Dahl, D. Huygens, A.R. Mahjoub, and P. Pesneau, On the k-edge disjoint 2-hop-constrained paths polytope, Oper Res Lett 34 (2006), 577-582.

[9] A. Balakrishnan and K. Altinkemer, Using a hop-constrained model to generate alternative communication network design, ORSA J Comput 4 (1992), 192-205.

[10] W. Ben-Ameur, Constrained length connectivity and survivable networks, Networks 36 (2000), 17-33.

[11] W. Ben-Ameur and E. Gourdin, Internet routing and related topology issues, SIAM J Discrete Math 17 (2003), 18-49.

[12] H. Kerivin and A.R. Mahjoub, Design of survivable networks: A survey, Networks 46 (2005), 1-21.

[13] L.J. Leblanc and R. Reddoch, Reliable link topology/capacity and routing in backbone telecommunication networks,Working paper 90-08, Owen Graduate School of Management, Vanderbilt University, Nashville, TN, 1990.

[14] L.J. Leblanc, R. Reddoch, J. Chifflet, and P. Mahey, Packet routing in telecommunications networks with path and flow restrictions, INFORMS J Comput 11 (1999), 188-197.

[15] H. Pirkul and S. Soni, New formulations and solution procedures for the hop constrained network design problem, Eur J Oper Res 148 (2003), 126-140.

[16] Huygens D., Labbé M., Mahjoub A.R., Pesneau P., The two-edge connected hop-constrained network design problem: Valid inequalities and branch-and-cut. Networks 49(1) (2007), 116-133.

[17] Bendali F., Diarrassouba I., Mahjoub A.R., J. Mailfert, The k edge-disjoint 3-hop-constrained paths polytope, Discrete Optimization 7 (2010), 222-233.

[18] Grötschel M. and Monma C.L., Integer polyhedra associated with certain network design problems with connectivity constraints, SIAM Journal on Discrete Mathematics, 3:502-523, 1990.

[19] TSPLIB, http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95

[20] Edmonds J., Karp, R.M., Theoretical improvements in algorithmic efficiency for network flow problems, Journal of the ACM (Association for Computing Machinery) 19 (2), 248-264, 1972.

[21] Diarrassouba I., Survivable Network Design Problems with High Connectivity Requirements, PhD Thesis, Université Blaise Pascal, France, 2009.

[22] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin, "Network Flows: Theory, Algorithms, and Applications", *New Jersey : Prentice Hall*, 1993.