CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

A Branch-and-Cut Algorithm for the Minimum Branch Vertices Spanning Tree Problem

Selene Silvestri
Gilbert Laporte
Raffaele Cerulli

November 2015

CIRRELT-2015-60

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

A Branch-and-Cut Algorithm for the Minimum Branch Vertices Spanning Tree Problem

Selene Silvestri[1,2,*], Gilbert Laporte[1,3], Raffaele Cerulli[4]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

[2] Department of Computer Science, University of Salerno, Via Giovanni Paolo II 138, 84084, Fisciano, Italy

[3] Department of Management Sciences, HEC Montréal, 3000 chemin de la Côte-Sainte-Catherine, Montréal, Canada H3T 2A7

[4] Department of Mathematics, University of Salerno, Via Giovanni Paolo II 138, 84084, Fisciano, Italy

Abstract. Given a connected undirected graph *G = (V;E)*, the Minimum Branch Vertices Problem (MBVP) asks for a spanning tree of *G* with the minimum number of vertices having degree greater than two in the tree. These are called branch vertices. This problem, which has an application in the context of optical networks, is known to be NP-hard. We model the MBVP as an integer linear program, with undirected variables, we derive valid inequalities and prove than some these are facet defining. We then develop a hybrid formulation containing undirected and directed variables. Both models are solved by branch-and-cut. Comparative computational results show the superiority of the hybrid formulation.

Keywords. Spanning tree, branch vertices, branch-and-cut.

Acknowledgements: This research was partly supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) under grant 2015-06189. This support is gratefully acknowledged.

---

* Corresponding author: selene.silvestri@gmail.com

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2015

1. Introduction

Given a connected undirected graph $G = (V, E)$, with $n = |V|$ vertices and $m = |E|$ edges, the Minimum Branch Vertices Problem (MBVP) aims to find a spanning tree T of G with the minimum number of branch vertices, i.e. vertices having a degree greater than two. For the input

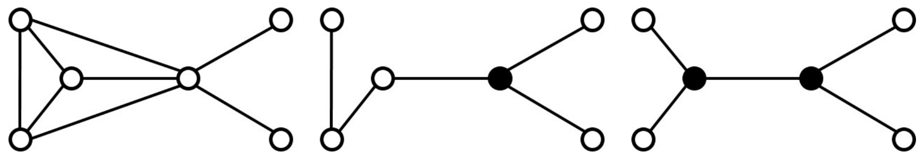

Figure 1: For a given graph on the left, two spanning trees with one and two branch vertices.

graph given on the left of Figure 1, we depict two spanning trees with different numbers of branch vertices. The spanning tree in the middle has one branch vertex and the one on the right has two. The best known application of MBVP arises in the context of optical networks. In such networks, an optical signal has to be split whenever it enters a node having degree greater than two. The split has to be performed using an appropriate network switch. These switches must be located at all the branch vertices, which can significantly increase the cost of the network.

The MBVP was introduced by Gargano et al. [6], who proved that it is NP-hard. Since then, the problem has been extensively investigated by several authors [2], [3], [4], [9], [10], [15], [16], [17]. Carrabs et al. [2] consider four IP formulations. The first formulation contains the well-know Dantzig et al. [5] subtour elimination constraints. Due to the exponential number of constraints, the authors consider this formulation not suitable to be tested on instances of significant size, but they solve it in a Lagrangian relaxation fashion. The second formulation is the most studied in the literature. It guarantees connection by sending from a source vertex one unit of flow to every other vertex of the graph. The third formulation is based on a multi-commodity flows. The fourth formulation makes use of the Miller-Tucker-Zemlin subtour elimination constraints [11]. Finally, Marín [9] presents a branch-and-cut algorithm based on a strengthened single commodity flow formulation. The author also provides a two-stage heuristic to reduce the computational time and to produce good feasible solutions when the optimum cannot be found within a reasonable time.

Our aim is to develop new formulations and a polyhedral-based exact branch-and-cut algorithm for the MBVP. The remainder of the paper is organized as follows. In Section 2, the problem is formulated as an integer linear program with undirected variables. In this section, we also investigate some properties of the problem and we analyze its LP relaxation. In Section 3, we derive the dimension of the polyhedron as well as some facet related results, and we introduce some valid inequalities. In Section 4, we present a directed graph reformulation and we adapt to this formulation several properties of the problem and some valid inequalities to yield an hybrid formulation. The branch-and-cut algorithm is described in Section 5. Comparative computational results and conclusions are presented in Section 6 and 7, respectively.

2. Undirected formulation, properties and bounds

The MBVP can be formulated as an integer linear program (ILP) with undirected variables as follows. Let x_e be a binary variable equal to 1 if and only if edge $e \in E$ belongs to the spanning tree T. For each vertex $v \in V$, let y_v be a binary variable equal to 1 if and only if vertex v has degree greater than equal to 3 in T, i.e. v is a branch vertex. In addition, for $S \subset V$, define $E(S) = \{e = (v, u) \in E : v, u \in S\}$ and $\delta(S) = \{e = (v, u) \in E : v \in S,\ u \in V \setminus S\}$. If $S = \{v\}$, we simply write $\delta(v)$ instead of $\delta(\{v\})$. The ILP formulation is then

$$\text{minimize } z = \sum_{v \in V} y_v \tag{1}$$

subject to

$$\sum_{e \in E(S)} x_e \le |S| - 1 \qquad S \subset V,\ |S| \ge 2 \tag{2}$$

$$\sum_{e \in E} x_e = n - 1 \tag{3}$$

$$\sum_{e \in \delta(v)} x_e - 2 \le (|\delta(v)| - 2) y_v \qquad v \in V \tag{4}$$

$$2y_v \le \sum_{e \in \delta(v)} x_e - 1 \qquad v \in V \tag{5}$$

$$x_e \in \{0, 1\} \qquad e \in E \tag{6}$$

$$y_v \in \{0, 1\} \qquad v \in V. \tag{7}$$

In this formulation, constraints (2) are the classical Dantzig, Fulkerson and Johnson [5] subtour elimination constraints. They guarantee that the edges in the solution cannot form cycles. Constraint (3) forces the selection of exactly $n-1$ edges. Constraints (4) and (5) are logical constraints linking the binary variables x_e with the binary variables y_v. They ensure that y_v is equal to 1 if and only if vertex v is branch. The objective function (1) requires the minimization of the number of branch vertices. Note that constraints (5) are necessary in order to make variables y_v represent exactly a set of branch vertices, but this condition is satisfied for any optimal solution even if we remove them.

2.1. Spanning tree properties

We now present some properties that a spanning tree must satisfy and we make some observations that will allow us to preprocess the instances. For a given vertex v, we can write the set of incident edges $\delta(v)$ as

$$\delta(v) = \delta_L(v) \cup \delta_I(v), \tag{8}$$

where $\delta_L(v) = \{(v, u) \in \delta(v) : |\delta(u)| = 1\}$ and $\delta_I(v) = \{(v, u) \in \delta(v) : |\delta(u)| > 1\}$. A first observation is that each edge belonging to the set $\delta_L(v)$, for a given vertex v, must belong to the optimal tree T:

$$x_e = 1 \qquad v \in V, \quad e \in \delta_L(v). \tag{9}$$

Moreover, it is easy to see that

$$y_v = 0 \qquad v \in V :\ |\delta(v)| \le 2 \tag{10}$$

$$y_v = 1 \qquad v \in V :\ |\delta_L(v)| \ge 2. \tag{11}$$

Note that for each vertex v such that $|\delta_L(v)| = 1$, constraints (4) and (5) become respectively

$$\sum_{e\in\delta_I(v)} x_e - 1 \le (|\delta_I(v)| - 1)y_v \qquad v \in V: \ |\delta_L(v)| = 1 \tag{12}$$

$$y_v \le \sum_{e\in\delta_I(v)} x_e - 1 \qquad v \in V: \ |\delta_L(v)| = 1. \tag{13}$$

To ensure the connectivity property, the inequalities

$$\sum_{e\in\delta_I(v)} x_e \ge 1 \qquad v \in V \tag{14}$$

must be satisfied.
Marín [9] defines a *bridge* as an edge $e \in E$ such that the graph $(V, E \setminus \{e\})$ becomes disconnected and defines 2-*cocycle* a set of two edges $\{e, f\} \subset E$ such that the graph $(V, E \setminus \{e, f\})$ becomes disconnected, but e and f are not bridges. It is easy to see that all bridges of a connected graph must belong to the edge set of any spanning tree:

$$x_e = 1 \qquad e \in E: \ (V, E \setminus \{e\}) \text{ is disconnected.} \tag{15}$$

Moreover, at least one of the edges of a 2-cocycle set must belong to any feasible solution:

$$x_e + x_f \ge 1 \qquad e, f \in E: \ \{e, f\} \text{ is a 2-cocycle.} \tag{16}$$

Note that all edges belonging to the set $\bigcup_{v\in V} \delta_L(v)$ are particular bridges. Removing any one of them isolates a vertex. Identifying bridges and 2-cocycle sets can be achieved by means of the algorithm proposed by Schmidt [14] which is used by Marín [9].
In this paper we extend the definition of *bridge* to the vertices. We define a *bridge vertex* as a vertex $v \in V$ such that the graph $G \setminus v = (V \setminus \{v\}, E \setminus \delta(v))$ is disconnected. Let $\bar{c}_v$ be the number of components of the graph $G \setminus v$ and let $C_i(v) = (V_{C_i}(v), E_{C_i}(v))$, $i = 1, \dots, \bar{c}_v$, be the corresponding components, such that $\bigcup_{i=1}^{\bar{c}_v} V_{C_i}(v) = V \setminus \{v\}$ and $\bigcup_{i=1}^{\bar{c}_v} E_{C_i}(v) = E \setminus \delta(v)$. For a given bridge vertex v, we can write the set of incident edges $\delta(v)$ as

$$\delta(v) = \bigcup_{i=1}^{\bar{c}_v} \delta_{C_i(v)}(v), \tag{17}$$

where $\delta_{C_i(v)}(v) = \{(v, u) \in \delta(v) : u \in V_{C_i}(v)\}$. If we denote V_B the set of bridge vertices, it is easy to see that

$$y_v = 1 \qquad v \in V_B: \ \bar{c}_v \ge 3 \tag{18}$$

$$\sum_{e\in\delta_{C_i(v)}(v)} x_e - 1 \le (|\delta_{C_i(v)}(v)| - 1)y_v \qquad v \in V_B: \ \bar{c}_v = 2, \ i = 1, 2 \tag{19}$$

$$\sum_{e\in\delta_{C_i(v)}(v)} x_e \ge 1 \qquad v \in V_B, \ i = 1, \dots, \bar{c}_v. \tag{20}$$

Note that inequalities (19) and (20) are a restricted version of (4) and (14) respectively. Bridge vertices are also called as *cut vertices* [1]. A connected graph G is *2-connected* if G contains no *cut* *vertex.* A connected graph G is called *2-disconnected* if it contains no bridge vertex v such that $G \setminus v$ is disconnected into more than two components. Note that if all the vertices of a graph are bridge vertices or have degree equal to one, the graph is a tree. For this reason, in the remainder of this section we assume that G contains at least one cycle.

Lemma 1. Let $G = (V, E)$ be a 2-disconnected graph. Then, for any $v \in V$, there exists a spanning tree T in G such that v is not a branch vertex in T.

Proof. Since G is 2-disconnected, $G \setminus v$ can be connected or disconnected into two components. If it is connected, there exists a spanning tree T_v in $G \setminus v$, therefore $T = T_v \cup \{e\}$ is a spanning tree in G, for any $e \in \delta(v)$, such that $\delta_T(v) = 1$. If $G \setminus v$ is disconnected, there exist two spanning trees T_1 and T_2 in $C_1(v)$ and $C_2(v)$, respectively. Hence, for an arbitrary $e_1 \in \delta_{C_1}(v)$ and $e_2 \in \delta_{C_2}(v)$, $T = T_1 \cup T_2 \cup \{e_1, e_2\}$ is a spanning tree in G such that $\delta_T(v) = 2$. □

2.2. Lower bounds

Let P_{STP} be the spanning tree polytope defined by (2), (3) and (6), and let P_u the intersection of P_{STP} with constraints (4) and (7). As previously observed, constraints (4) guarantee that a vertex v has to be branch whenever at least three edges incident to it are selected. Even if they do not explicitly force $y_v = 0$ when $\sum_{e\in\delta(v)} x_e \leq 2$ holds, this will be the case because of the objective function. Therefore, although P_u does not define the MBVP polytope,

$$\min\left\{\sum_{v\in V} y_v : (x, y) \in P_u\right\} \tag{21}$$

can be used to find optimal solutions for the problem. Moreover, valid lower bounds are given by the LP relaxation of (21). One important property is provided in the following result.

Proposition 1. The value of the LP relaxation of (21) can be obtained by solving

$$\min\left\{\sum_{e=(v,u)\in E}\Big(\frac{1}{|\delta(v)|-2} + \frac{1}{|\delta(u)|-2}\Big)x_e - \sum_{v\in V}\frac{2}{|\delta(v)|-2} : x_e \in P_{STP}\right\}. \tag{22}$$

In other words, an optimal solution to the LP relaxation of (21) is given by a least cost spanning tree of G, under the edge costs defined above. Note that, the right-most term in (22) is a constant.

Proof. It is easy to see that inequalities (4), suitably rewritten as

$$y_v \geq \sum_{e\in\delta(v)} \frac{1}{|\delta(v)|-2} x_e - \frac{2}{|\delta(v)|-2} \qquad v \in V \tag{23}$$

for the LP relaxation of (21), must be tight. Indeed, the objective function only contains variables y_v with positive cost coefficients. Therefore, replacing y_v in the objective function by the right-hand side of (23), for all $v \in V$, and dropping inequalities (4), MBVP can be obtained by solving (22). □

3. Polyhedral analysis of the undirected formulation

We now derive some polyhedral results for the Spanning Tree Problem with Bounded Number of Branch Vertices. In this section we assume that $G = (V, E)$ is a complete graph on $|V| = n$ vertices, so that $|E| = m = n(n-1)/2$. In order to provide our polyhedral results, we need some preliminary results.

Definition 1. A polyhedron $S = \{x \in \mathbb{R}^k : Ax \le b\}$ is full-dimensional if $dim(S) = k$, where (A, b) is an $m \times (k+1)$ matrix.

Let $M = \{1, \ldots, m\}$, $M^= = \{i \in M : a^i x = b_i \text{ for all } x \in S\}$ and $M^\le = \{i \in M : a^i x < b_i \text{ for some } x \in S\} = M \setminus M^=$. Let $(A^=, b^=)$ and $(A^\le, b^\le)$ the corresponding rows of (A, b). According to this notation, the following proposition holds true (see Proposition 2.4 of Nemhauser and Wolsey [18]):

Proposition 2. If $S \subseteq \mathbb{R}^k$, then $dim(S) + rank(A^=, b^=) = k$.

We represent subsets of vertices and edges by their characteristic vectors $y \in \mathbb{B}^n$ and $x \in \mathbb{B}^m$, respectively. Therefore, $V' \subseteq V$ is represented by the vector $y^{V'}$, where $y_v^{V'} = 1$ if $v \in V'$ and $y_v^{V'} = 0$ otherwise, and $E' \subseteq E$ is represented by the vector $x^{E'}$, where $x_e^{E'} = 1$ if $e \in E'$ and $x_e^{E'} = 0$ otherwise. Denote by P the polytope defined by the convex hull of feasible solutions, that is,

$$P = \{(x, y) \in \mathbb{R}^{|E|+|V|} : \ (x, y) \text{ satisfy } (2) - (7)\}. \tag{24}$$

Proposition 3. The dimension of the polytope P is $dim(P) = |E| + |V| - 1$.

Proof. A *Hamiltonian path* of the graph is a feasible solution to the MBV and the corresponding characteristic vector is $(x^H, \emptyset)$, where $H \subset E$ contains all the edges of the path. In a complete graph we can identify m Hamiltonian paths whose corresponding characteristic vectors are affinely independent. Moreover, for each vertex $v \in V$, the point $(x^{\delta(v)}, y^{\{v\}})$ lies in P. It is easy to see that the n points $(x^{\delta(v)}, y^{\{v\}})$, $v \in V$, and the m points corresponding to the Hamiltonian paths are affinely independent. Hence $dim(P) \ge |E| + |V| - 1$. Because all points of P satisfy the equality (3) we have $rank(A^=, b^=) \ge 1$; hence, by Proposition 2, $dim(P) \le |E| + |V| - 1$. Therefore $dim(P) = |E| + |V| - 1$. □

Proposition 4. The inequality $y_v \ge 0$ defines a facet of P.

Proof. It is easy to see that the characteristic vector associated to a Hamiltonian path satisfies $y_v = 0$. Therefore, if $F = \{x \in P : y_v = 0\}$, $dim(F) \ge m - 1$. Moreover, $(x^{\delta(w)}, y^{\{w\}}) \in F$, for all $w \in V$ such that $w \ne v$, hence $dim(F) \ge m + n - 2$. Being F a proper face of P, $dim(F) \le m + n - 2$. This allow us to conclude that $y_v \ge 0$ is a facet for any $v \in V$. □

Proposition 5. The inequalities $x_e \ge 0$ and $x_e \le 1$ define facets of P.

Proof. Given a complete graph $G = (V, E)$ and an edge $e = (u, w) \in E$, we can identify $m - 1$ Hamiltonian paths in G whose corresponding characteristic vectors are affinely independent and such that $x_e = 0$. Moreover, the points $(x^{\{\delta(u) \setminus \{e\} \cup \{(w,u_1)\}\}}, y^{\{u\}})$, $(x^{\{\delta(w) \setminus \{e\} \cup \{(u,w_1)\}\}}, y^{\{w\}})$, for some $u_1, w_1 \in V$, $(x^{\delta(v)}, y^{\{v\}})$, $v \in V \setminus \{u, w\}$ are n affinely independent points, feasible for P and such that $x_e = 0$. Therefore $x_e \ge 0$ defines a facet of P. The proof for the inequality $x_e \le 1$ proceeds in the same way. □

The following theorem is useful to establish whether a valid inequality is a facet (see Theorem 3.6 of Nemhauser and Wolsey [18]).

Theorem 1. Let $(A^=, b^=)$ be the equality set of $S \subseteq \mathbb{R}^k$ and let $F = \{x \in S : \pi x = \pi_0\}$ be a proper face of S. The following two statements are equivalent:

- F is a facet of S.
- If $\lambda x = \lambda_0$ for all $x \in F$ then

$$(\lambda, \lambda_0) = (\alpha\pi + uA^=, \alpha\pi_0 + ub^=) \text{ for some } \alpha \in \mathbb{R} \text{ and some } u \in \mathbb{R}^{|M^=|}. \tag{25}$$

Proposition 6. The valid inequalities (5) are facets for the MBVP.

Proof. We prove the result by showing that the conditions of Theorem (1) hold. Consider a fixed vertex $v \in V$. Without lost of generality, we can assume that $v = v_1$, where $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_{|\delta(v)|}, \ldots, e_m\}$, i.e. the first $|\delta(v)|$ edges of E belong to $\delta(v)$. We can therefore write the valid inequality (5) associated to v as

$$(1, \ldots, 1, 0, \ldots, 0)x^T + (-2, 0, \ldots, 0)y^T \geq 1, \tag{26}$$

and hence $F = \{(x, y) \in P : (1, \ldots, 1, 0, \ldots, 0)x^T + (-2, 0 \ldots, 0)y^T = 1\}$. In order for (x, y) to belong to F, only two cases are possible:

- If vertex v is a branch vertex, $y_v = 1$, then $\sum_{e\in\delta(v)} x_e$ has to be equal to 3, therefore $\sum_{e\in E\setminus\delta(v)} x_e = n - 4$.
- If vertex v is not a branch vertex, $y_v = 0$, then $\sum_{e\in\delta(v)} x_e$ has to be equal to 1, therefore $\sum_{e\in E\setminus\delta(v)} x_e = n - 2$.

Let T_1 and T_2 be two spanning trees of G such that, $|\delta_{T_1}(v)| = 3$ and $|\delta_{T_2}(v)| = 2$. The corresponding characteristic vectors are feasible solutions for the MBVP ensuring that the inequalities (5) are proper faces of P. Therefore, in order to prove that F represents a facet of P, from to Theorem 1, it is sufficient to show that if $\lambda(x, y)^T = \lambda_0$ for all $(x, y) \in F$, then (λ, λ_0) can be expressed as $(\alpha\pi + uA^=, \alpha\pi_0 + ub^=)$, for some $\alpha \in \mathbb{R}$, $u \in \mathbb{R}^{|M^=|}$. As showed above, in our case $(\pi, \pi_0) = (1, \ldots, 1, 0, \ldots, 0, -2, 0, \ldots, 0, 1)$, $(A^=, b^=) = (1, \ldots, 1, 0, \ldots, 0, n - 1)$ and $|M^=| = 1$. For convenience, we represent (λ, λ_0) as

$$(\lambda, \lambda_0) = (s_1, \ldots, s_{|\delta(v)|}, r_{|\delta(v)|+1}, \ldots, r_m, t_1, \ldots, t_n, \lambda_0). \tag{27}$$

Hence $\lambda(x, y)^T = \lambda_0$ can be expressed as

$$\sum_{e\in\delta(v)} s_e x_e + \sum_{e\in E\setminus\delta(v)} r_e x_e + t_1 y_1 + \sum_{w\in V\setminus v} t_w y_w = \lambda_0. \tag{28}$$

Let T be a spanning tree of $(V \setminus \{v\}, E \setminus \delta(v))$, and let T_e and T_f be the spanning trees obtained by adding to T the edges $e = (v, u)$ and $f = (v, w)$, respectively, where $|\delta_T(u)| \neq 2$ and $|\delta_T(w)| \neq 2$. Note that $|\delta_{T_e}(v)| = |\delta_{T_f}(v)| = 1$ and hence $y_v = 0$. It is then easy to see that $(x^{E_{T_e}}, y^{V_{T_e}}) \in F$ and $(x^{E_{T_f}}, y^{V_{T_f}}) \in F$, and therefore they satisfy (28). Consequently, $\lambda(x^{E_{T_e}}, y^{V_{T_e}})^T - \lambda(x^{E_{T_f}}, y^{V_{T_f}})^T = 0$. Through simple algebraic manipulations, we obtain $s_e = s_f$. Because T is a generic spanning

tree, we can conclude that $s_1 = \ldots = s_{|\delta(v)|}$. From now on, we will denote this coefficient vector as s.
Let T_g be a spanning tree of G and a let $g \in E_{T_g}$ be an edge such that $g = (u, w)$, with $|\delta_{T_g}(u)| \neq 3$ and $|\delta_{T_g}(w)| \neq 3$. The characteristic vector $(x^{E_{T_g}}, y^{V_{T_g}})$ is feasible for the MBVP and (5) is satisfied as an equality. Let T_h be the spanning tree of G obtained by removing the edge g from the set E_{T_g} and by adding an edge $h = (\bar{u}, \bar{w})$, such that $|\delta_{T_h}(\bar{u})| \neq 3$ and $|\delta_{T_g}(\bar{w})| \neq 3$. Again, $(x^{E_{T_h}}, y^{V_{T_h}})$ is feasible for the MBVP and (5) is satisfied as an equality. Therefore, $\lambda(x^{E_{T_g}}, y^{V_{T_g}})^T - \lambda(x^{E_{T_h}}, y^{V_{T_h}})^T = 0$. Note that the two trees differ in just one edge and $y^{V_{T_g}} = y^{V_{T_h}}$. Then through simple algebraic manipulations, we obtain $r_g = r_h$. Since T_g is a generic spanning tree, and g and h are two generic edges, we can conclude that $r_{|\delta(v)|+1} = \ldots = r_m$. From now on, we will denote this coefficient vector as r.
Let T_b a spanning tree of G such that $|\delta_{T_b}(v)| = 3$. It is easy to see that $(x^{E_{T_b}}, y^{V_{T_b}}) \in F$. Without loss of generality we can assume that $\delta_{T_b}(v) = \{e_1, e_2, e_3\}$. The graph $(V, E_{T_b} \setminus \{e_2, e_3\})$ contains three acyclic components C_1, C_2 and C_3, one of which includes vertex v. We can assume $v \in V_{C_3}$. Let $u_1 \in \{V_{C_1} \cup V_{C_2}\}$ and $u_2 \in V_{C_3}$, such that $|\delta_{T_b}(u_1)| \neq 2$ and $|\delta_{T_b}(u_1)| \neq 2$. If $e_1 = (v, w_1)$ and $e_2 = (v, w_2)$, the spanning tree $T_d = (V, E_{C_1} \cup E_{C_2} \cup E_{C_3} \cup \{(u_1, u_2), (w_1, w_2)\})$ will be a feasible solution for the MBV satisfying (5) as an equality. Therefore $\lambda(x^{E_{T_b}}, y^{V_{T_b}})^T - \lambda(x^{E_{T_d}}, y^{V_{T_d}})^T = 0$. Note that $y^{V_{T_b}} = y^{V_{T_d}}$, then through simple algebraic manipulations, we obtain $t_1 = -2(s - r)$.
Let T_q be a spanning tree of G such that $|\delta_{T_q}(v)| = 1$ and let $(\bar{u}, \bar{w}) \in E_{T_q}$, where $|\delta_{T_q}(\bar{u})| = 3$ and $|\delta_{T_q}(\bar{w})| \neq 3$. Let T_p be the spanning tree $(V, \{E_{T_q} \setminus \{(\bar{u}, \bar{w})\} \cup \{(\bar{u_1}, \bar{w_1})\}\})$, where $|\delta_{T_p}(\bar{u_1})| \neq 3$ and $|\delta_{T_p}(\bar{w_1})| \neq 3$. It is easy to see that $(x^{E_{T_q}}, y^{V_{T_q}}) \in F$ and $(x^{E_{T_p}}, y^{V_{T_p}}) \in F$. Assuming without loss of generality that $\bar{u} = v_2$, by calculating $\lambda(x^{E_{T_q}}, y^{V_{T_q}})^T - \lambda(x^{E_{T_p}}, y^{V_{T_p}})^T = 0$, we obtain $t_2 = 0$. Note that this is true for a generic spanning tree T_q and a generic vertex $\bar{u}$. Therefore we conclude that $t_i = 0$, $i = 2, \ldots, n$. Substituting $(x^{E_{T_q}}, y^{V_{T_q}})$ in (28), we obtain $\lambda_0 = -s + nr$ and

$$(\lambda, \lambda_0) = (s, \ldots, s, r, \ldots, r, -2(s - r), 0, \ldots, 0, -s + nu). \tag{29}$$

Note that

$$(\alpha\pi + uA^{=}, \alpha\pi_0 + ub^{=}) = (\alpha + u, \ldots, \alpha + u, u, \ldots, u, -2\alpha, 0, \ldots, 0, \alpha + un - u). \tag{30}$$

Hence, setting $\alpha = s - r$ and $u = r$, we obtain

$$(\lambda, \lambda_0) = (\alpha + u, \ldots, \alpha + u, u, \ldots, u, -2\alpha, 0, \ldots, 0, \alpha + un - u). \tag{31}$$

Thanks to Theorem (1), the proof is thus complete.

□

Proposition 7. For $v \in V$ and $S \subseteq \delta(v)$ with $|S| \geq 3$,

$$\sum_{e \in S} x_e - 2 \leq (|S| - 2) y_v \tag{32}$$

is valid for P.

Proof. It is easy to see that for any subset S of $\delta(v)$, if more than two edges belong to the optimal solution, then vertex v has to be branch. Note that, for $S = \delta(v)$ we obtain constraints (4), therefore (32) represent a generalized version. □

Proposition 8. For any bridge vertex $v \in V_B$ with $\bar{c}_v = 2$, for $C_i(v)$, $i = 1, 2$ such that $|V_{C_i}(v)| \geq 2$, for $D \subseteq \delta_{C_i(v)}(v)$ with $|D| = 2$,

$$\sum_{e \in D} x_e \leq 1 + y_v \tag{33}$$

is valid for P.

Proof. Note that, v being a bridge vertex with $\bar{c}_v = 2$, as stated above, at least one edge connecting v with $C_i(v)$ for $i = 1, 2$, has to be selected. As soon as a second edge connecting v with one of the two components is selected, vertex v becomes a branch vertex and y_v has to be activated. □

Proposition 9. For $v \in V$ and $Q \subset \delta(v)$ such that $|Q| = |V| - 2$

$$y_v \leq \sum_{e \in Q} x_e \tag{34}$$

is valid for P.

Proof. This inequality means that if there exists at least one $Q \subset \delta(v)$ such that all the edges in Q do not belong to the spanning tree, then vertex v cannot be branch. □

Proposition 10. Let $R = (V_R, E_R)$ be a cycle of cardinality three, i.e. $V_R = \{a, b, c\}$ and $E_R = \{f_{ab}, f_{ac}, f_{bc}\}$. For $v \in V_R$ such that $|\delta(v)| = 3$, without lost of generality assume that $v = a$,

$$y_a + x_{f_{bc}} \leq 1 \tag{35}$$

is valid for P. Moreover, if there exist at least two vertices a and b in the cycle having degree 3 in the graph, then

$$y_a + y_b \leq x_{f_{ab}} \tag{36}$$

is valid for P. Finally, if the three vertices all have degree 3, then

$$y_a + y_b + y_c \leq 1. \tag{37}$$

Proof. Constraints (35) state that if a is branch, then the edge f_{bc} cannot be selected for otherwise the solution would contain a cycle. Conversely, if edge f_{bc} belongs to the solution, then a will not be a branch vertex. Constraints (36) impose that only one vertex between a and b can be branch whenever edge f_{ab} is selected. If the three vertices have degree 3, then constraints (37) state that only one of them can be a branch vertex. □

4. Directed and hybrid reformulations

Problems originally defined over undirected graphs can often be reformulated over corresponding directed graphs. In this section we consider a directed integer programming reformulation (DIP) of MBVP as a spanning arborescence problem. To develop a model for this directed version of the problem, we choose an arbitrary vertex $r \in V$ as the root vertex and we consider the directed graph $D = (V, A)$ obtained by replacing each edge $(v, u) \in E$ by arcs (v, u) and (u, v) in A. In addition to the previously defined variables y_v, $v \in V$, for each arc $a \in A$, we define z_a as a binary variable equal to 1 if and only if arc a belongs to the spanning arborescence A. In association with graph D, we define $\delta^+(w) = \{(v,u) \in A : v = w\}$ and $\delta^-(w) = \{(v,u) \in A : u = w\}$. The DIP formulation is then

$$\text{minimize } z = \sum_{v \in V} y_v \tag{38}$$

subject to

$$\sum_{e \in A(S)} z_a \leq |S| - 1 \qquad S \subset V,\ |S| \geq 2 \tag{39}$$

$$\sum_{a \in A} z_a = n - 1 \tag{40}$$

$$\sum_{a \in \delta^-(v)} z_a = 1 \qquad v \in V \setminus \{r\} \tag{41}$$

$$\sum_{a \in \delta^+(v)} z_a - 1 \leq (|\delta^+(v)| - 2) y_v \qquad v \in V \setminus \{r\} \tag{42}$$

$$\sum_{a \in \delta^+(r)} z_a - 2 \leq (|\delta^+(r)| - 2) y_r \tag{43}$$

$$2y_v \leq \sum_{a \in \delta^+(v)} z_a \qquad v \in V \setminus \{r\} \tag{44}$$

$$2y_r \leq \sum_{a \in \delta^+(r)} z_a - 1 \tag{45}$$

$$z_a \in \{0,1\} \qquad a \in A \tag{46}$$

$$y_v \in \{0,1\} \qquad v \in V. \tag{47}$$

Constraints (39), (40) and (46) characterize the spanning arborescence polytope. Note that the inequalities (4) and (5), for the undirected graph formulation, are split into inequalities (42), (43) and (44), (45), respectively, for the directed graph reformulation. Also observe that due to (41), one unit is subtracted in the left-hand side of (42) instead of two units in the corresponding inequalities (4).

It is easy to see that several of the properties described for the undirected formulation are easily adaptable to the directed one. Moreover, with the only exception of the root vertex r, no more than one outwards pointing arc may be incident to a no branch vertex. Hence the inequalities

$$\sum_{a \in W} z_a - 1 \leq (|W| - 1) y_v \qquad v \in V \setminus \{r\},\ W \subset \delta^+(v) : |W| \geq 2 \tag{48}$$

are clearly valid for the directed formulation. Now, denote by P_D the polytope defined by the convex hull of feasible solution in the directed graph, that is:

$$P_D = \{(z,y) \in \mathbb{R}^{|A|+|V|} :\ (z,y) \text{ satisfy } (39) - (47)\}. \tag{49}$$

Proposition 11. The undirected and the directed formulation for the Minimum Branch Vertex Spanning Tree Problem are equivalent if constraints $x_e \geq 0$ and

$$x_e = z_{vu} + z_{uv} \qquad e = (v, u) \in E \tag{50}$$

are introduced in the DIP model.

Proof. Constraints $x_e \geq 0$ and $x_e = z_{vu} + z_{uv}$, for $e = (v, u) \in E$, together with (39), (40) and (46), yield an alternative description of the P_{STP} (see [8] for the details). Moreover, because $x_e = z_{vu} + z_{uv}$, for any $v \in V \setminus \{r\}$, summing up (41) and (42) we obtain (4), and summing up (41) and (44) we obtain (5). Therefore P and P_D are equivalent and this concludes the proof. □

Thanks to Proposition 11, the polytope defined by constraints (2)−(7), (42), (43), (46) and (50) defines the set of feasible solutions for the MBVP. We refer to it as the *hybrid reformulation*.

5. Branch-and-cut algorithm

We solve the MBVP by means of a branch-and-cut algorithm which is summarized in Algorithm 1. Before executing the algorithm we apply a preprocessing phase in which the graph is reduced by exploiting the properties introduced in Section 2.1. In line 1, an initial feasible solution is identified by searching a minimum spanning tree using Prim's algorithm [13]. With any edge $e = (v, u)$ we associate weight $w_e = n$ if $\min\{|\delta(v)|, |\delta(u)|\} \leq 2$, otherwise $w_e = n - \max\{|\delta(v)|, |\delta(u)|\}$. In line 3, the first subproblem is obtained by relaxing the subtour elimination constraints (2), except for the case where $|S| = 3$, as well as the integrality constraints on the variables. We also identify all the bridges, the cocycles and the bridge vertices of the graph and we add the correspondent constraints (16), (19) and (20). In line 13, a search for violated constraints (2) is performed on the integer solutions by identifying the connected components and by adding the subtour elimination constraints induced by the subsets of vertices of all the components containing at least one cycle. In line 17, at a non-integer solution, constraints (2) are separated using the max-flow algorithm proposed by Padberg and Wolsey [12]. The max-flow obtained with this algorithm is $f = |\bar{S}| - \sum_{e \in E(\bar{S})} x_e + kost$, where $\{\bar{S}, V \setminus \bar{S}\}$ represents the cut-set associated to the max-flow and *kost* is a constant value depending on the vertex set V, therefore a constraint is violated if $f - kost$ is less than 1. To avoid adding constraints with a small violation, a constraint is generated whenever $f - kost$ is less than $1 - \epsilon$, for a fixed ϵ depending on the instances. For the non-integer solutions, we run the max-flow procedure only on the root node.
The branch-and-cut algorithm was applied to both undirected and hybrid formulations. In the first case, in line 15, a search for violated inequalities (32) and (33) is performed. Valid inequalities (34), (35), (36) and (37) turned out to be ineffective and were not considered. A subset of the most violated inequalities (33) is added to the cut-pool. The separation procedure used for inequalities (32) is that of Lucena et al. ([7]). Let $(\bar{x}, \bar{y})$ be a feasible solution for the linear programming relaxation, and for every $v \in V$ such that $|\delta(v)| \geq 3$, order the elements in $\{\bar{x}_e : e \in \delta(v)\}$ in decreasing value. Then, for $(\bar{x}, \bar{y})$, $v \in V$, and every $k \in \{3, \ldots, |\delta(v)| - 1\}$, compute $\sum_1^k \bar{x}_{e_k} - (k-2)\bar{y}_v$. This procedure identifies a set S of cardinality k with the largest value for the left-hand side of (32) for vertex v. If that value is greater than 2, it has identified the most violated inequality, otherwise, no violated inequality (32) exists for v. For any vertex $v \in V$, having $|\delta(v)| \geq 3$, we first consider all $S \subseteq \delta(v)$ such that $|S| = 3$ and we add a subset of the most violated inequalities (32) by the current relaxed solution. Moreover, we run the procedure previously described for

Algorithm 1: Branch-and-cut algorithm

Input: integer program P.
Output: an optimal solution of P.

```
Identify initial feasible solution T_0. Get number b_0 of branch vertices in T_0
ub ← b_0, L = ∅
Define a first subproblem and insert it in the list L
while L is not empty do
    chose the subproblem and remove it from L
    solve the subproblem to obtain the lower bound lb
    if lb < ub then
        if the solution is integer then
            if the solution is feasible then
                ub ← lb
                update incumbent solution
            else
                search and add SEC on integer solutions
        else
            search violated constraints
            if root node then
                search SEC on non-integer solutions
            if violated constraints are identified then
                add them to the model
            else
                branch on a variable and add the corresponding subproblems in L
```

$k \in \{4, \ldots, |\delta(v)| - 1\}$ and we add at most one violated constraint for each value of k. In line 19 all the violated constraints identified are added to the model. In the implementation for the hybrid formulation, in line 15, a search for violated inequalities (48) is also performed. The separation procedure is the same described for inequalities (32). As for the previous case, we first look for all subsets $W \subset \delta^+(v)$ such that $|W| = 2$ and a subset of the most violated inequalities is added to the cut-pool, then the separation procedure is performed for $k \geq 3$. In line 21, branching takes place in priority on the y_v variables.

6. Computational results

The branch-and-cut algorithm was coded in C and solved using IBM ILOG CPLEX 12.5.1. The computational experiments were performed on a 64-bit GNU/Linux operating system, 96 GB of RAM and one processor Intel Xeon X5675 running at 3.07 GHz. In our tests the *MIPEmphasis* parameter is set on the best bound value and the others parameters as default. For all the instances the constant ϵ introduced to identify violated constrains (2) on the non-integer solutions is set equal to 0.7. Experiments for the MBVP were conducted on benchmark instances. Carrabs et al. [2] generated instances with n between 20 and 1000 and different densities. Note that dense graphs often can contain a Hamiltonian path, therefore the authors generated sparse graphs. These instances were also used by Marín [9]. In his paper the author divides the instances into two groups: medium instances (with $n \leq 500$) and large instances (with $n \geq 600$). Here we call *small* the instances with $n \leq 200$, *medium* those with $250 \leq n \leq 5000$ and *large* those with $n \geq 600$. Table 1 and 2 report the results for the undirected formulation applied to the small and medium instances. In the tables each line represents an average over five instances having the same number of vertices and of edges. In both tables the first two columns represent the instances, columns **ub**, **opt** and **sec** report the average of the upper bounds found with Prim's algorithm, the average of the optimal solution values and the average of the computational time needed to compute them. Moreover, whenever α instances of a group are not solved to optimality within the time limit of one hour, we write (α) appears close to the solution value. The numbers of bridges, cocycles and bridge vertices are also reported. Columns **nodes** and **cuts** represent the number of nodes in the search tree and the number of cuts added.
Results for small, medium and large instances for the hybrid formulation are reported in Table 3 and Table 4. In Table 3 each line represent an average over 25 instances having the same number of vertices. The table reports the results for both small and medium instances. In Table 4 each line represents an average over five instances having the same number of vertices and edges. The two tables have the same structure described above. Note that in this case, all the instances were solved optimally. Our experimental results show that the hybrid formulation is most efficient and faster. It allows us to solve all the small and medium instances within less than 10 seconds, while the undirected formulation could not find an optimal solution on 13 instances after one hour. Moreover, we can solve all the large instances, up to $n = 1000$ within an average time of 90.5 seconds. Finally, note that the number of nodes in the search tree is relatively small and all families of cuts are useful.

6.1. LP lower bounds and duality gaps

We present the LP lower bounds obtained by adding one valid inequality each time to the hybrid formulation. In Table 5 each line is an average over 25 instances, while in Table 6 the average is computed over five instances. In both the tables the first two columns represent the

Table 1: Undirected formulation: computational results for small instances

n	m	ub	opt	bridge	cocycle	bridge vertex	nodes	cuts	sec
20	27	4.6	2.4	6.2	5.6	4.2	0.0	3.2	0.0
20	34	5.4	1.2	2.4	5.0	2.2	0.0	5.2	0.0
20	42	3.6	0.2	0.8	2.2	0.8	0.0	0.0	0.0
20	49	3.6	0.0	0.0	1.2	0.0	0.0	0.4	0.0
20	57	3.0	0.0	0.2	0.0	0.2	0.0	1.8	0.0
40	50	12.2	7.4	16.2	13.4	9.2	0.0	15.8	0.0
40	60	9.2	3.4	7.4	13.6	5.6	0.4	22.2	0.0
40	71	10.8	1.6	5.2	7.4	4.6	0.0	18.2	0.0
40	81	8.4	0.8	2.2	6.6	2.2	0.0	19.4	0.0
40	92	8.2	0.6	2.2	4.4	2.2	0.0	13.8	0.0
60	71	19.6	13.0	28.4	27.6	15.6	0.0	16.0	0.0
60	83	18.0	8.2	17.8	21.0	11.6	1.8	56.0	0.1
60	95	15.4	5.4	12.0	18.8	9.8	25.0	189.8	0.4
60	107	15.6	3.4	7.2	13.2	6.4	1.0	126.4	0.2
60	119	12.8	1.6	4.8	11.6	4.8	7.6	151.6	0.3
80	93	24.0	16.4	40.8	35.0	21.2	1.6	27.0	0.1
80	106	23.6	12.0	27.4	30.0	17.4	7.0	77.8	0.1
80	120	22.4	8.8	19.4	23.4	13.8	36.4	195.4	0.5
80	133	21.0	5.6	12.4	25.0	10.6	12.8	186.6	0.8
80	147	18.6	3.4	9.8	19.2	8.4	17.2	199.6	0.4
100	114	31.6	23.8	56.8	38.6	27.4	4.2	25.4	0.1
100	129	32.0	16.4	38.6	35.6	22.4	8.0	109.2	0.5
100	144	29.8	11.8	26.2	32.2	18.0	18.8	189.2	0.6
100	159	27.4	8.4	18.6	32.4	14.8	47.4	334.2	1.1
100	174	24.4	6.2	15.4	25.2	11.8	4937.0	2220.6	126.9
120	136	39.6	29.6	69.8	45.6	33.4	10.4	36.4	0.1
120	152	38.8	21.8	48.4	48.4	27.8	19.2	124.4	0.4
120	169	34.6	16.0	36.4	38.4	23.2	28.6	214.4	0.8
120	185	33.2	11.6	25.4	41.4	18.2	162.0	455.4	1.8
120	202	31.8	8.6	20.4	34.6	15.0	93.4	442.8	2.3
140	157	45.4	34.2	79.8	71.0	38.6	14.0	64.0	0.3
140	175	43.6	25.8	59.0	57.6	33.4	15.4	141.6	0.7
140	193	40.6	18.8	41.6	52.8	28.4	124.8	329.8	1.8
140	211	39.2	15.2	35.6	41.2	24.0	128.2	466.4	1.9
140	229	36.0	10.6	23.8	43.0	19.2	253.0	750.4	4.8
160	179	52.6	39.8	94.0	64.6	44.8	0.0	28.8	0.2
160	198	49.4	31.2	69.2	68.2	37.8	45.6	179.8	1.1
160	218	47.2	23.4	50.2	62.8	31.2	112.6	359.2	1.9
160	237	44.6	17.4	39.4	50.8	27.4	198.0	542.6	2.9
160	257	44.0	13.4	32.2	43.0	24.8	248.4	799.0	6.6
180	200	58.6	46.4	111.6	76.4	51.4	12.0	52.4	0.4
180	221	55.6	35.0	79.8	67.0	44.2	99.6	215.0	1.5
180	242	54.2	25.4	58.8	69.6	37.0	204.4	491.2	3.7
180	263	53.2	21.0	46.6	61.0	32.4	805.0	905.8	14.5
180	284	47.6	17.6	39.4	56.0	29.6	528.2	1100.8	11.9
200	222	63.6	50.6	127.8	74.8	57.0	14.6	69.4	0.6
200	244	62.0	39.4	92.4	77.8	49.6	49.8	174.0	1.3
200	267	59.4	30.4	69.0	72.0	40.2	130.8	390.0	3.7
200	289	56.4	24.8	56.8	68.8	38.4	2166.4	1185.8	56.6
200	312	57.2	(1)25.8	42.2	57.6	30.2	5464.6	3942.2	732.5

Table 2: Undirected formulation: computational results for medium instances

n	m	ub	opt	bridge	cocycle	bridge vertex	nodes	cuts	sec
250	273	81.4	66.0	164.4	100.2	71.4	1.4	51.0	0.8
250	297	78.6	53.0	120.8	110.8	60.8	312.0	318.8	5.0
250	321	75.8	43.4	101.8	93.8	57.6	277.4	514.8	7.4
250	345	74.6	34.4	76.2	90.0	47.8	1616.2	930.2	47.9
250	369	70.8	26.2	60.0	85.2	40.2	732.4	1352.8	42.7
300	326	97.4	81.0	203.0	121.8	87.4	30.2	127.2	1.9
300	353	95.0	67.8	160.2	116.4	78.6	171.4	323.6	6.2
300	380	92.6	54.6	124.8	114.0	69.0	572.4	785.4	21.9
300	407	89.6	46.2	104.6	103.4	61.8	1808.0	1619.6	75.9
300	434	85.0	37.2	86.4	89.4	56.2	1657.2	1933.0	143.8
350	378	113.4	94.6	238.8	143.2	102.8	70.2	152.6	5.0
350	406	111.6	80.6	190.0	145.6	93.6	452.4	476.8	10.1
350	435	108.0	65.6	151.0	150.8	84.4	2016.2	1379.4	85.6
350	463	107.2	56.6	124.2	128.4	75.8	11731.0	1945.2	663.4
350	492	102.6	45.4	103.6	123.8	67.2	5569.6	2322.0	444.0
400	429	130.8	111.8	282.6	167.2	119.6	56.2	123.6	3.9
400	459	128.0	94.0	226.4	165.0	109.4	851.6	782.6	21.0
400	489	126.2	$^{(1)}$88.4	184.8	152.4	99.0	2315.8	5068.8	742.9
400	519	122.2	68.4	154.2	154.4	88.4	9878.8	2517.8	979.4
400	549	118.4	56.0	131.2	141.2	80.2	3204.6	2962.6	350.2
450	482	148.6	125.8	318.6	177.8	135.4	33.6	116.0	4.8
450	515	146.0	107.4	250.6	202.8	121.6	1298.6	846.2	75.4
450	548	140.0	90.4	208.8	184.2	110.4	3686.0	4059.0	835.4
450	581	139.2	$^{(1)}$77.6	176.6	167.8	100.4	12719.2	2901.4	1363.9
450	614	133.2	$^{(3)}$66.4	151.8	153.8	93.8	17717.4	3686.4	2766.6
500	534	164.6	141.6	361.0	191.2	150.6	70.4	149.2	10.2
500	568	160.8	120.8	294.2	187.0	137.2	948.6	770.0	53.8
500	603	158.2	105.6	246.0	198.4	126.8	3089.4	1981.2	260.3
500	637	151.6	$^{(2)}$117.2	210.6	181.2	116.8	4850.8	6615.4	1902.9
500	672	148.4	$^{(5)}$122.8	170.0	194.6	104.4	13407.2	11163.8	3600.0

Table 3: Hybrid formulation: computational results for small and medium instances

n	m	ub	opt	bridge	cocycle	bridge vertex	nodes	cuts	sec
20	41.8	4.0	0.8	1.9	2.8	1.5	0.0	1.8	0.0
40	70.8	9.8	2.8	6.6	9.1	4.8	0.2	33.6	0.1
60	95.0	16.3	6.3	14.0	18.4	9.6	0.0	67.4	0.5
80	119.8	21.9	9.2	22.0	26.5	14.3	1.0	83.9	0.7
100	144.0	29.0	13.3	31.1	32.8	18.9	1.7	108.7	1.0
120	168.8	35.6	17.5	40.1	41.7	23.5	2.6	135.0	1.1
140	193.0	41.0	20.9	48.0	53.1	28.7	6.2	178.8	2.0
160	217.8	47.6	25.0	57.0	57.9	33.2	2.8	165.6	1.9
180	242.0	53.8	29.1	67.2	66.0	38.9	9.0	212.3	2.5
200	266.8	59.7	32.6	77.6	70.2	43.1	6.8	213.4	3.1
250	321.0	76.2	44.6	104.6	96.0	55.6	5.8	209.8	3.1
300	380.0	91.9	57.4	135.8	109.0	70.6	6.0	230.2	4.2
350	434.8	108.6	68.6	161.5	138.4	84.8	7.9	298.8	6.9
400	489.0	125.1	81.8	195.8	156.0	99.3	21.0	355.2	9.1
450	548.0	141.4	93.4	221.3	177.3	112.3	17.5	333.7	9.5
500	602.8	156.7	106.7	256.4	190.5	127.2	10.3	332.0	9.8

Table 4: Hybrid formulation: computational results for large instances

n	m	ub	opt	bridge	cocycle	bridge vertex	nodes	cuts	sec
600	637	197.6	183.8	493.6	68.8	188.0	0.0	74.0	3.2
600	674	192.6	167.2	437.4	71.6	176.4	0.0	148.8	8.7
600	712	188.0	150.6	394.4	68.6	168.6	1.6	229.0	10.3
600	749	182.2	138.8	363.4	55.6	161.0	21.2	335.6	17.6
600	787	173.8	125.8	333.6	49.4	153.2	18.2	333.8	16.2
700	740	232.0	214.4	576.8	91.4	218.6	0.0	100.4	8.7
700	780	224.8	198.0	518.4	89.2	206.8	2.6	176.6	11.0
700	821	218.0	180.0	470.2	79.4	198.2	0.6	257.2	12.5
700	861	212.4	164.0	436.6	62.8	191.4	3.2	291.0	17.4
700	902	205.0	154.2	403.2	63.6	183.2	1.0	293.6	14.7
800	843	265.4	245.6	666.8	90.6	252.2	0.0	102.0	10.3
800	886	256.8	227.6	599.4	98.8	237.4	1.8	169.0	11.2
800	930	253.6	208.4	546.6	89.2	228.8	10.2	321.6	22.7
800	973	245.2	194.2	505.8	82.0	221.4	72.4	658.4	48.8
800	1017	232.2	176.2	468.2	71.4	212.8	23.8	479.8	37.1
900	944	300.6	279.6	756.4	105.4	284.8	0.0	118.4	12.6
900	989	290.0	259.2	685.6	110.4	271.4	188.8	339.6	66.2
900	1034	286.6	240.6	633.0	105.0	262.2	28.2	405.4	30.2
900	1079	281.4	223.2	583.6	98.0	251.2	12.6	489.8	90.5
900	1124	269.0	206.0	547.6	83.2	242.4	2.0	372.0	30.7
1000	1047	332.6	312.0	849.6	110.2	317.0	8.4	148.8	26.2
1000	1095	323.2	290.0	767.0	121.0	303.2	0.0	209.2	17.0
1000	1143	318.6	271.2	705.0	121.2	290.2	74.2	613.4	57.1
1000	1191	310.4	251.0	657.6	109.8	279.8	53.6	621.2	75.4
1000	1239	303.8	235.2	609.8	105.6	268.4	45.8	735.4	62.6

Table 5: Hybrid formulation: lower bounds for MBVP on small and medium instances

n	m	opt	w(H)	w(H1)	w(H2)	w(H3)	w(H4)	w(H5)
20	41.8	0.8	0.59	0.65	0.64	0.60	0.61	0.62
40	70.8	2.8	2.16	2.31	2.29	2.21	2.23	2.25
60	95.0	6.3	5.18	5.64	5.56	5.33	5.42	5.44
80	119.8	9.2	7.79	8.44	8.25	8.05	8.19	8.10
100	144.0	13.3	11.75	12.47	12.28	12.06	12.23	12.16
120	168.8	17.5	15.60	16.46	16.24	16.02	16.20	16.14
140	193.0	20.9	18.64	19.64	19.36	19.18	19.47	19.17
160	217.8	25.0	22.74	23.75	23.55	23.25	23.55	23.31
180	242.0	29.1	26.39	27.62	27.30	27.08	27.41	27.06
200	266.8	32.6	30.00	31.22	30.91	30.55	30.94	30.68
250	321.0	44.6	41.72	43.06	42.68	42.43	42.96	42.42
300	380.0	57.4	53.74	55.55	54.93	54.73	55.23	54.51
350	434.8	68.6	63.96	65.93	65.36	65.40	66.16	65.00
400	489.0	81.8	77.31	79.33	78.68	78.96	79.62	78.21
450	548.0	93.4	88.32	90.66	89.75	90.04	90.79	89.29
500	602.8	106.7	101.75	104.26	103.43	103.48	104.16	102.86

instances and the third one the optimal solution. The next columns provide lower bounds $w(H)$,

Table 6: Hybrid formulation: lower bounds for MBVP on large instances

n	m	opt	w(H)	w(H1)	w(H2)	w(H3)	w(H4)	w(H5)
600	637	183.8	180.40	180.94	180.70	182.03	182.79	180.63
600	674	167.2	163.83	164.63	164.47	164.97	166.11	164.45
600	712	150.6	147.24	148.31	147.99	148.10	148.96	147.73
600	749	138.8	136.09	136.97	136.82	136.50	136.83	136.75
600	787	125.8	123.87	124.66	124.66	124.17	124.46	124.85
700	740	214.4	211.03	211.56	211.30	212.89	213.68	211.08
700	780	198.0	193.67	194.79	194.56	195.11	196.52	194.30
700	821	180.0	175.72	177.14	176.91	177.11	178.28	176.56
700	861	164.0	160.81	161.82	161.73	161.29	161.87	161.77
700	902	154.2	151.16	152.43	152.42	151.71	152.17	152.50
800	843	245.6	242.02	242.55	242.25	244.04	245.08	242.17
800	886	227.6	223.44	224.24	224.15	224.82	226.47	223.67
800	930	208.4	204.26	205.36	205.26	205.55	206.92	204.82
800	973	194.2	189.87	191.48	191.15	190.73	191.68	191.12
800	1017	176.2	172.37	173.72	173.63	172.99	173.49	173.79
900	944	279.6	275.15	275.76	275.33	277.72	278.88	275.17
900	989	259.2	253.97	255.29	254.81	256.15	257.71	254.26
900	1034	240.6	235.66	236.88	236.84	237.38	239.09	236.37
900	1079	223.2	218.02	219.98	219.59	219.52	220.61	219.09
900	1124	206.0	202.19	203.78	203.50	202.87	203.41	203.47
1000	1047	312.0	307.48	308.28	307.97	310.08	311.33	307.63
1000	1095	290.0	283.03	284.62	284.23	286.72	288.50	283.78
1000	1143	271.2	265.37	266.94	266.76	267.40	269.43	266.37
1000	1191	251.0	244.99	246.92	246.68	246.82	248.22	246.29
1000	1239	235.2	230.27	232.16	231.90	231.10	231.92	231.74

$w(H1)$, $w(H2)$, $w(H3)$, $w(H4)$ and $w(H5)$, where H denotes the polytope obtained by relaxing the integrality constraints in the hybrid formulation, while $H1$ and $H2$ denote the intersection of H with (48) for $W \subset \delta^+(v)$ such that $|W| = 2$ and $|W| \geq 3$, respectively. Moreover, $H3$ denotes the intersection of P with (33), while $H4$ and $H5$ with (32) for $S \subseteq \delta(v)$ such that $|S| = 3$ and $|S| \geq 4$, respectively. It is easy to see from the tables that all the cuts help improve the lower bound, in particular $w(H1)$ and $w(H4)$ seems to yield the best lower bounds in most cases. Inequalities (48) for $W \subset \delta^+(v)$ such that $|W| = 2$ and (32) for $S \subseteq \delta(v)$ such that $|S| = 3$ are the most useful cuts. This is evident in Table 7 and 8 which provide the duality gap with respect to the optimal solution on the six polytopes.

7. Conclusions

We have modeled and solved the Minimum Branch Vertices Spanning Tree Problem. We have provided two mathematical formulations based on an undirected and on a directed graph, respectively and an hybrid formulation obtained by merging the first two. Moreover, we have derived some properties and some valid inequalities for the problem. A branch-and-cut approach was proposed on the undirected and on the hybrid formulations. Results show that the hybrid formulation is superior to the undirected formulation and that our branch-and-cut algorithm applied to it solves all benchmark instances to optimality.

Table 7: Hybrid formulation: duality gap on small and medium instances

n	m	opt	gH(%)	gH1(%)	gH2(%)	gH3(%)	gH4(%)	gH5(%)
20	41.8	0.8	29.1	16.8	17.9	26.4	25.6	23.3
40	70.8	2.8	28.0	19.5	20.7	24.8	23.9	22.4
60	95.0	6.3	22.1	12.1	13.6	18.5	16.6	16.3
80	119.8	9.2	18.6	9.5	11.9	14.7	12.9	14.1
100	144.0	13.3	13.4	6.8	8.5	10.4	8.9	9.5
120	168.8	17.5	12.3	6.4	7.9	9.4	8.2	8.6
140	193.0	20.9	12.3	6.5	8.1	9.1	7.5	9.1
160	217.8	25.0	10.1	5.4	6.3	7.7	6.3	7.4
180	242.0	29.1	10.2	5.3	6.5	7.4	6.1	7.5
200	266.8	32.6	8.8	4.5	5.6	6.8	5.5	6.4
250	321.0	44.6	6.9	3.6	4.5	5.1	3.8	5.1
300	380.0	57.4	6.7	3.3	4.4	4.8	3.9	5.2
350	434.8	68.6	7.2	4.0	4.9	4.8	3.6	5.5
400	489.0	81.8	5.9	3.2	4.0	3.6	2.8	4.6
450	548.0	93.4	5.7	3.0	4.0	3.7	2.8	4.6
500	602.8	106.7	4.9	2.4	3.2	3.1	2.5	3.8

Table 8: Hybrid formulation: duality gap on large instances

n	m	opt	gH(%)	gH1(%)	gH2(%)	gH3(%)	gH4(%)	gH5(%)
600	637	183.8	1.9	1.6	1.7	1.0	0.6	1.8
600	674	167.2	2.1	1.6	1.7	1.4	0.7	1.7
600	712	150.6	2.3	1.5	1.8	1.7	1.1	1.9
600	749	138.8	2.0	1.3	1.4	1.7	1.4	1.5
600	787	125.8	1.6	0.9	0.9	1.3	1.1	0.8
700	740	214.4	1.6	1.3	1.5	0.7	0.3	1.6
700	780	198.0	2.2	1.6	1.8	1.5	0.8	1.9
700	821	180.0	2.4	1.6	1.7	1.6	1.0	1.9
700	861	164.0	2.0	1.3	1.4	1.7	1.3	1.4
700	902	154.2	2.0	1.2	1.2	1.6	1.3	1.1
800	843	245.6	1.5	1.3	1.4	0.6	0.2	1.4
800	886	227.6	1.9	1.5	1.5	1.2	0.5	1.8
800	930	208.4	2.0	1.5	1.5	1.4	0.7	1.7
800	973	194.2	2.3	1.4	1.6	1.8	1.3	1.6
800	1017	176.2	2.2	1.4	1.5	1.9	1.6	1.4
900	944	279.6	1.6	1.4	1.6	0.7	0.3	1.6
900	989	259.2	2.1	1.5	1.7	1.2	0.6	1.9
900	1034	240.6	2.1	1.6	1.6	1.4	0.6	1.8
900	1079	223.2	2.4	1.5	1.6	1.7	1.2	1.9
900	1124	206.0	1.9	1.1	1.2	1.5	1.3	1.2
1000	1047	312.0	1.5	1.2	1.3	0.6	0.2	1.4
1000	1095	290.0	2.5	1.9	2.0	1.1	0.5	2.2
1000	1143	271.2	2.2	1.6	1.7	1.4	0.7	1.8
1000	1191	251.0	2.5	1.7	1.8	1.7	1.1	1.9
1000	1239	235.2	2.1	1.3	1.4	1.8	1.4	1.5

Acknowledgements

This research was partly supported by the Canadian Natural Sciences and Engineering Research Council under grant 2015-06189. This support is gratefully acknowledged.

References

[1] J. A. Bondy and U. S. R. Murty. *Graph theory with applications*, volume 290. Macmillan, London, 1976.

[2] F. Carrabs, R. Cerulli, M. Gaudioso, and M. Gentili. Lower and upper bounds for the spanning tree with minimum branch vertices. *Computational Optimization and Applications*, 56:405–438, 2013.

[3] C. Cerrone, R. Cerulli, and A. Raiconi. Relations, models and a memetic approach for three degree-dependent spanning tree problems. *European Journal of Operational Research*, 232:442–453, 2014.

[4] R. Cerulli, M. Gentili, and A. Iossa. Bounded-degree spanning tree problems: models and new algorithms. *Computational Optimization and Applications*, 42:353–370, 2009.

[5] G. B. Dantzig, D. R. Fulkerson, and S. M. Johnson. Solution of a large-scale traveling-salesman problem. *Journal of the operations research society of America*, 2:393–410, 1954.

[6] L. Gargano, P. Hell, L. Stacho, and U. Vaccaro. Spanning trees with bounded number of branch vertices. In *Automata, Languages and Programming*, pages 355–365. Springer Berlin Heidelberg, 2002.

[7] A. Lucena, N. Maculan, and L. Simonetti. Reformulations and solution algorithms for the maximum leaf spanning tree problem. *Computational Management Science*, 7:289–311, 2010.

[8] T. L. Magnanti and L. A. Wolsey. Optimal trees. In M. O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser, editors, *Network Models. Handbooks in Operations Research and Management Science 6*, pages 503–615. North-Holland, Amsterdam, 1995.

[9] A. Marín. Exact and heuristic solutions for the minimum number of branch vertices spanning tree problem. *European Journal of Operational Research*, 245:680–689, 2015.

[10] M. Merabet, S. Durand, and M. Molnar. Minimization of branching in the optical trees with constraints on the degree of nodes. In *ICN'12: The Eleventh International Conference on Networks*, pages 235–240, Saint-Gilles, Réunion Island, 2012.

[11] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the Association for Computing Machinery*, 7:326–329, 1960.

[12] M. W. Padberg and L. A. Wolsey. Trees and cuts. In *Combinatorial Mathematics, Annals of Discrete Mathematics 17*, pages 511–517. North-Holland, Amsterdam, 1983.

[13] R. C. Prim. Shortest connection networks and some generalizations. *Bell System Technical Journal*, 36:1389–1401, 1957.

[14] J. M. Schmidt. A simple test on 2-vertex- and 2-edge-connectivity. *Information Processing Letters*, 113:241–244, 2013.

[15] D. M. Silva, R. M. A. Silva, G. R. Mateus, J. F. Gonçalves, M. G. C. Resende, and P. Festa. An iterative refinement algorithm for the minimum branch vertices problem. In *Experimental Algorithms*, pages 421–433. Springer, Berlin Heidelberg, 2011.

[16] R. M. A. Silva, D. M. Silva, M. G. C. Resende, G. R. Mateus, J. F. Gonçalves, and P. Festa. An edge-swap heuristic for generating spanning trees with minimum number of branch vertices. *Optimization Letters*, 8:1225–1243, 2014.

[17] S. Sundar, A. Singh, and A. Rossi. New heuristics for two bounded-degree spanning tree problems. *Information Sciences*, 195:226–240, 2012.

[18] L. A. Wolsey and G. L. Nemhauser. *Integer and Combinatorial Optimization.* Wiley, New York, 2014.