



**HAL**  
open science

# A Generalization of the Minimum Branch Vertices Spanning Tree Problem

Massinissa Merabet, Jitamitra Desai, Miklós Molnár

► **To cite this version:**

Massinissa Merabet, Jitamitra Desai, Miklós Molnár. A Generalization of the Minimum Branch Vertices Spanning Tree Problem. ROADEF 2018 - 19e Congrès de la Société Française de Recherche Opérationnelle et d'Aide à la Décision, Feb 2018, Lorient, France. lirmm-01888086

**HAL Id: lirmm-01888086**

**<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01888086>**

Submitted on 16 Apr 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Generalization of the Minimum Branch Vertices Spanning Tree Problem

Massinissa Merabet<sup>1</sup>, Jitamitra Desai<sup>2</sup>, and Miklos Molnar<sup>3</sup>

<sup>1</sup> Energy Research Institute

Nanyang Technological University (NTU), Singapore

`mmerabet@ntu.edu.sg`

<sup>2</sup> Manufacturing and Industrial Engineering Cluster

School of Mechanical and Aerospace Engineering

Nanyang Technological University (NTU), Singapore

`jdesai@ntu.edu.sg`

<sup>3</sup> The Montpellier Laboratory of Informatics, Robotics and Microelectronics

University of Montpellier, France

`miklos.molnar@lirmm.fr`

**Abstract.** Given a connected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , a vertex  $v \in \mathcal{V}$  is said to be a *branch vertex* if  $d(v) > 2$ , where  $d(v)$  denotes the degree of vertex  $v$ . The *Minimum Branch Vertices Spanning Tree* (MBVST) problem is to find a spanning tree of  $\mathcal{G}$  with the minimum number of branch vertices. This problem has been extensively studied in the literature and has well-developed applications notably related to routing in optical networks. In this paper, we propose a generalization of this problem, where we begin by introducing the notion of a *k-branch vertex*, which is a vertex with degree strictly greater than  $k + 2$ , and the goal is to determine a spanning tree of  $\mathcal{G}$  with the minimum number of *k-branch vertices* (*k-MBVST* problem). In the context of optical networks, the parameter  $k$  can be seen as the limiting capacity of optical splitters to split the input light signal to  $k$  sub-trees. Proofs of NP-hardness and non-inclusion in the APX class of the *k-MBVST* problem are established for a generic value of  $k$ , and then an ILP formulation of the *k-MBVST* problem based on single commodity flow balance constraints is derived. Computational results based on randomly generated graphs show that the number of *k-branch vertices* included in the spanning tree increases with the size of the vertex set  $\mathcal{V}$ , but decreases with  $k$  as well as graph density. We also show that when  $k \geq 4$ , the number of *k-branch vertices* in the optimal solution is close to zero, regardless of the size and the density of the underlying graph.

**Keywords:** Spanning Tree, Minimization of Branch Vertices, Integer Linear Programming, MBVST, *k-MBVST*, Optical Networks.

## 1 Introduction

Given a connected graph  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ , a vertex  $v \in \mathcal{V}$  is defined to be a *branch vertex* if its degree (denoted  $d(v)$ ) is strictly greater than two, i.e.,  $d(v) > 2$  for any branch vertex. The *Minimum Branch Vertices Spanning Tree (MBVST)* problem is to find a spanning tree of graph  $\mathcal{G}$  with the minimum number of branch vertices. This NP-hard and non-APX problem [GHSV02] has been well-studied in the literature and Cerrulli et al. [CGI09] were the first to formulate this problem as an *integer linear program (ILP)*, wherein they used single commodity flow balance constraints to guarantee connectivity. In [CCGG13], Carabbs et al. provided two alternative ILP formulations based on multi-commodity flow balance constraints and the well-known cycle eliminating Miller-Tucker-Zemlin constraints, respectively. They also determined lower and upper bounds for the MBVST using the Lagrangian relaxation method. In [Mar15], Marin presented a branch-and-cut algorithm based on an enforced integer programming formulation for the MBVST problem. In [CCR14], Cerrone et al. determined a unified memetic algorithm for three related problems, namely the MBVST; the problem of minimizing the degree sum of branch vertices (MDST); and the minimum leaves problem.

Recently, Merabet et al. proved in [MDM13b] that the set of optimal solutions for MBVST and the set of optimal solutions for MDST are disjoint. They also proposed two variants of these problems, where specifically in [MDM13a], a more flexible graph structure, namely the so-called *hierarchy* is proposed. A hierarchy, which can be seen as a generalization of trees, is defined as a *homomorphism* of a tree in a graph [Mol08], and as minimizing the number of branch vertices in a hierarchy has no practical relevance, the authors determined the minimum cost spanning hierarchy such that the number of branch vertices is less than or equal to a given integer  $r$ .

The most widespread application of such MBVST problems arises in Wavelength-Division Multiplexing (WDM), which is an effective technique to exploit the available bandwidth of optical fibers to meet the explosive growth of bandwidth demand across the Internet [HGCT02]. Now, a multicast technique consists of simultaneously transmitting information from one source to multiple destinations in a bandwidth efficient way (duplicating the information only when required). From a computational viewpoint, multicast routing protocols in WDM networks are mainly based on *light-trees* [SM99], which require intermediate nodes to have the ability to split and direct the input signal to multiple outputs as and when necessary. A node which has the ability to replicate an input signal on any wavelength to any subset of output fibers is referred to as a Multicast-Capable (MC) node [MZQ98]. (A light-splitting switch has to be placed in the optical device to perform such a task at an MC node.) On the other hand, a node which has the ability to tap into a signal and forward it to only one output is called a Multicast-Incapable (MI) node. As light-splitter switches are rather expensive devices, it is imperative to minimize the number of MC nodes in the light-tree,

and hence this problem lends itself to being expressed as the MBVST problem (see Gargano et al. in [GHSV02]).

Extending this, if a light signal is split into  $k$  copies (at an MC node), then the signal power of each resultant copy is reduced by, at least, a factor of  $1/k$  of the original signal power [MJ00]. If  $k$  is too large, then the information cannot be read at the destinations due to the signal strength dropping below the minimum threshold value, and therefore,  $k$  functions as a tolerance parameter.

**Definition 1.** *A  $k$ -branch vertex is a vertex with degree strictly greater than  $k + 2$ .*

Therefore, given a  $k$ -branch vertex, it is useful to look for a light-tree in the WDM network with the minimum number of  $k$ -branch vertices, where  $k$  is fixed as the tolerance parameter. If the light-tree contains some  $k$ -branch vertices, an optical amplifier must be installed near each  $k$ -branch vertex to guarantee the efficiency of the broadcast/multicast.

**Definition 2.** *Let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  be a graph. The  $k$ -MBVST problem consists of finding a spanning tree  $T$  of  $\mathcal{G}$  such that the number of  $k$ -branch vertices in  $T$  is minimized.*

The remainder of this paper is organized as follows. In Section 2, proofs of NP-hardness and non-inclusion in the APX class of the  $k$ -MBVST problem are established for any generic value of  $k$ . Then, an ILP formulation of this problem based on single commodity flow balance constraints is derived in Section 3, and finally Section 4 records some preliminary computational results along with associated insights and conclusions.

## 2 Proofs of NP-hardness and negative approximability

In a Hamiltonian graph, it is evident that finding a 0-MBVST is equivalent to finding a Hamiltonian path in  $\mathcal{G}$ . Thus, the  $k$ -MBVST is NP-complete in this case. Furthermore, the classical MBVST problem is NP-complete, even on non-Hamiltonian graphs [GHSV02], and moreover, it is a particular case of the  $k$ -MBVST problem corresponding to  $k = 0$ . Therefore, the 0-MBVST is at least as difficult as the MBVST even in this case.

In the following discussion, we prove that the  $k$ -MBVST problem is NP-hard for any generic  $k > 0$ . Towards this end, denote by  $s_k(\mathcal{G})$  the smallest number of  $k$ -branch vertices in any spanning tree of  $\mathcal{G}$ .

**Theorem 1.** *Let  $r$  be a fixed non-negative integer. It is NP-complete to decide whether a given graph  $\mathcal{G}$  satisfies  $s_k(\mathcal{G}) \leq r$  whatever the value of  $k$ .*

*Proof.* – Case 1 :  $r = 0$ , let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  be a given connected graph. Construct a new graph  $\tilde{\mathcal{G}}$  by linking  $k$  leaves to each vertex  $v \in \mathcal{V}$ . Deciding whether  $\tilde{\mathcal{G}}$  contains a spanning tree with no  $k$ -branch vertex is equivalent to determining whether  $\mathcal{G}$  is Hamiltonian or not.

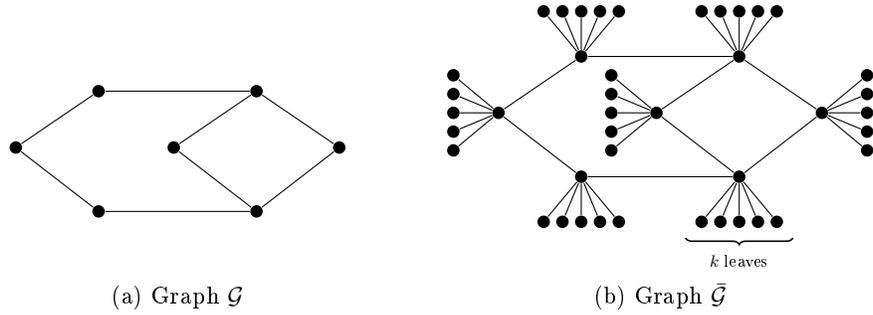


Fig. 1. Reduction from the Hamiltonian problem to the 0-MBVST ( $k = 5$ )

- Case 2 :  $r \geq 1$ , let  $\mathcal{G} = (\mathcal{V}, \mathcal{E})$  be a given connected graph. Construct a graph  $\bar{\mathcal{G}}$  by replicating  $r \cdot (k + 1)$  times the graph  $\mathcal{G}$  and add a chain  $C$  of size  $r + 2$ . Choose an arbitrary vertex  $v \in \mathcal{V}$  and link every *internal vertex* of  $C$  to  $k + 1$  distinct replications of  $\mathcal{G}$  from their corresponding vertices (duplicates)  $v$ . Moreover, link  $k$  leaves to each vertex of each duplication of  $\mathcal{G}$ . In any spanning tree of  $\bar{\mathcal{G}}$ , the  $k$  vertices of the complete graph are necessarily  $k$ -branch vertices. Thus, the graph  $\bar{\mathcal{G}}$  will contain a spanning tree with  $s_k(\bar{\mathcal{G}}) = r$  if and only if  $\mathcal{G}$  admits a Hamiltonian path starting from  $v$ .

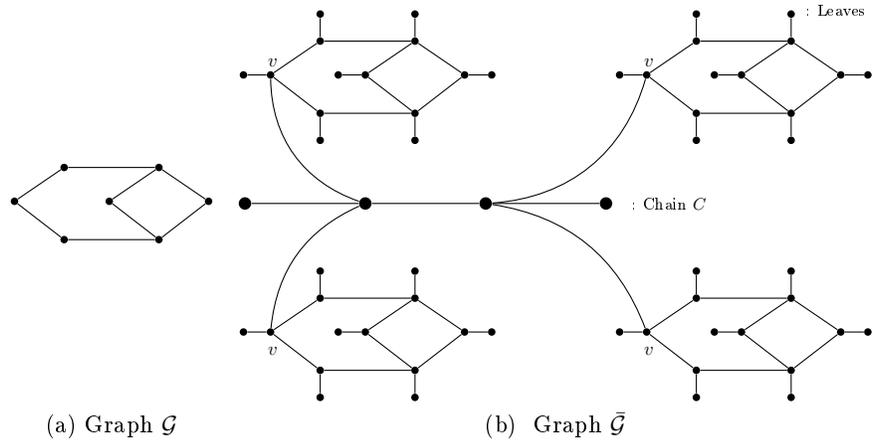


Fig. 2. Construction of the graph  $\bar{\mathcal{G}}$  for  $k = 1$  and  $r = 2$ .

Hence, proved.

In the following discussion, we show that the  $k$ -MBVST problem is not in the APX class for any generic  $k > 0$ . We prove this result by applying an AP-reduction  $(f, g, \rho)$  from the *Minimum Set Cover problem* (**MSC**) to the  $k$ -MBVST problem. Consider an instance of the MSC problem given by a ground set  $\mathcal{U} = \{x_1, x_2, \dots, x_n\}$ , and a collection of  $m$  subsets  $\mathcal{S} = \{\mathcal{S}_i\}_{i=1}^m$ , such that  $\bigcup_{i=1}^m \mathcal{S}_i \equiv \mathcal{U}$ . A solution to the MSC problem aims to find a minimum number of subsets whose union contains each element of  $\mathcal{U}$ . This MSC problem is NP-complete and is not in the APX class [GPMS<sup>+</sup>99].

**Theorem 2.** *The  $k$ -MBVST problem is not in APX for any value of  $k$ .*

*Proof.* Let  $\mathcal{X} = (\mathcal{U}, \mathcal{S})$  be a given instance of the MSC problem. We now construct the graph  $\mathcal{G}$  underlying an instance of the  $k$ -MBVST problem such that a feasible solution for the  $k$ -MBVST problem exists if and only if the instance  $\mathcal{X}$  contains a feasible solution for the MSC problem. This construction procedure is described next. Define  $\mathcal{G} = f(x, r) = f(x)$  by adding a vertex  $v_i$  corresponding to each element  $x_i \in \mathcal{U}$ . Similarly, add a vertex  $s_j$  for each subset  $\mathcal{S}_j \in \mathcal{U}$ . If  $x_i \in \mathcal{S}_j$ , then connect  $s_j$  and  $v_i$  by an edge. Add a vertex  $z$  and link it to each vertex  $x_i$ . Moreover, link  $k - 2$  leaves to each vertex  $v_i$  and link  $k - 1$  leaves to  $z$  (Figure 3).

This construction ensures that if an optimal solution for the MSC problem in  $\mathcal{X}$  contains  $n$  subsets then the optimal solution for the  $k$ -MBVST in  $\mathcal{G}$  contains  $n + 1$   $k$ -branch vertices. Moreover, if an optimal solution for the  $k$ -MBVST in  $\mathcal{G}$  contains  $n$   $k$ -branch vertices then an optimal solution for the MSC problem in  $\mathcal{X}$  contains exactly  $n - 1$  subsets. An additional but necessary criterion of the AP-reduction process is that the graph construction must be done in a polynomial time. Clearly, the polynomial time computation of the graph  $\mathcal{G}$  is trivial.

Let  $c^*(f(x, r))$  be the value of the optimum solution of the instance  $f(x, r)$  and let  $c(f(x, r), y)$  denote the value of a solution  $y$ . Let  $c^*(x)$  be the value of the optimum solution for an instance  $x$ . Finally, let  $c(x, g(x, y, r))$  be the value of a solution  $g(x, y, r)$ . Then, suppose that

$$r \geq \frac{c(f(x, r), y)}{c^*(f(x, r))}. \quad (1)$$

Our mapping yields that  $c^*(x) = c^*(f(x, r)) - 1$  and that  $c(x, g(x, y, r)) = c(f(x, r), y) - 1$ . Hence, for some  $r > 1$  and fixed  $\rho = 2$ , it is sufficient for us to show that

$$1 + 2(r - 1) \geq \frac{c(f(x, r), y) - 1}{c(f(x, r)) - 1}. \quad (2)$$

Using inequality (1), it is enough to prove that

$$(r - 1)(c^*(f(x, r)) - 2) \geq 0. \quad (3)$$

Since every spanning tree of  $\mathcal{G}$  has at least two  $k$ -branch vertices ( $z$  and at least one of  $s_i$ ), the inequality (3) is trivially true for  $r > 1$ .

Thus, the above defined  $(f, g, 2)$  is an AP-reduction from the MSC problem to the  $k$ -MBVST problem. Hence proved.

**Theorem 3.** *If an optimization problem  $\mathcal{P}_1$  is AP-reducible to an optimization problem  $\mathcal{P}_2$  and  $\mathcal{P}_2 \notin \text{APX}$ , then  $\mathcal{P}_1 \notin \text{APX}$  [GPMS<sup>+</sup> 99].*

It follows from theorems 2 and 3 that the  $k$ -MBVST problem does not belong to the APX class.

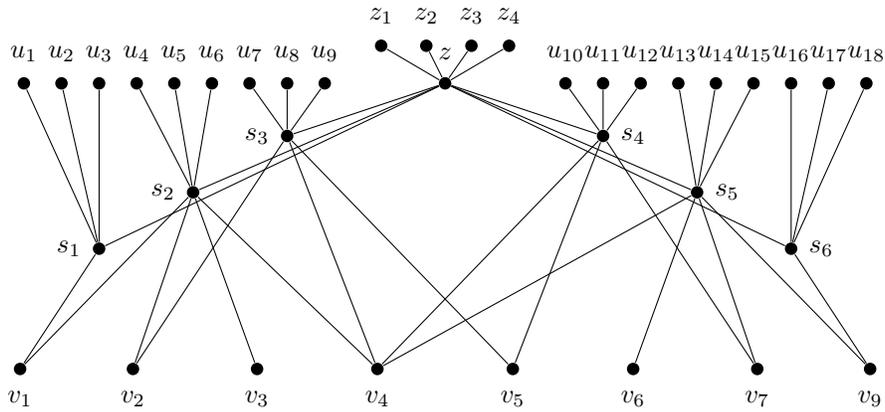


Fig. 3. Reduction of the Minimum Set Cover problem to the  $k$ -MBVST problem with  $k = 5$ ,  $\mathcal{U} = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ , and  $\mathcal{S} = \{\{v_1\}, \{v_1, v_2, v_3, v_4\}, \{v_2, v_4, v_5\}, \{v_4, v_5, v_7\}, \{v_4, v_6, v_7, v_9\}, \{v_9\}\}$ .

Having proved the NP-completeness and non-inclusion in the APX class of the  $k$ -MBVST problem, in the following section, we turn our attention to deriving an integer linear programming formulation of this problem.

### 3 An ILP formulation of the $k$ -MBVST problem

The formulation of the  $k$ -MBVST problem as an *integer linear program (ILP)* derived in this paper is predicated on the single balance commodity flow formulation proposed in [CGI09]. The basic construct underlying this formulation is as follows. In order to define a spanning tree  $T$  of  $\mathcal{G}$ , we can send from a source vertex  $s \in \mathcal{V}$ , one unit of flow to every other vertex  $v \in \mathcal{V} \setminus \{s\}$ . As this flow needs to be directed, the given graph  $\mathcal{G}$  has to be transformed into a symmetrically oriented graph  $\mathcal{G}^d = (\mathcal{V}, \mathcal{E}^d)$ , where each edge  $\{u, v\} \in \mathcal{E}$  now corresponds to two

arcs  $(u, v)$  and  $(v, u)$  in  $\mathcal{E}^d$ . For each arc  $(u, v) \in \mathcal{E}^d$ , we define an integer variable  $f_{(u,v)}$  representing the (directed) flow going from  $u$  to  $v$ . Furthermore, define a binary decision variable  $x_{(u,v)}$ , which equals 1 if  $f_{(u,v)}$  or  $f_{(v,u)}$  carry a non-zero flow, and 0 otherwise. Finally, for each  $v \in \mathcal{V}$ , we have a decision variable  $y_v$  that equals to 1, if  $v$  is a  $k$ -branch vertex in the spanning tree, and 0 otherwise. We are now ready to formally present the  $k$ -MBVST problem as follows.

The objective of our problem is to minimize the number of  $k$ -branch vertices belonging to the spanning tree of  $\mathcal{G}$ , which can be expressed as follows.

**Objective function:**

$$\text{Minimize } \sum_{v \in \mathcal{V}} y_v \quad (4a)$$

**Spanning tree constraints:**

$$\sum_{\substack{u \in \mathcal{V}: \\ (u,v) \in \mathcal{E}^d}} x_{(u,v)} = 1, \quad \forall v \in \mathcal{V} \setminus \{s\} \quad (4b)$$

$$\sum_{(u,v) \in \mathcal{E}^d} x_{(u,v)} = |\mathcal{V}| - 1. \quad (4c)$$

As a vertex with more than one parent creates a cycle, constraint (4b) ensures that each vertex except the source has exactly one predecessor. Next, as the number of edges in any spanning tree must be equal to  $|\mathcal{V}| - 1$ , constraint (4c) ensures that exactly  $|\mathcal{V}| - 1$  arcs are selected in the optimal solution. Note that these two constraints are necessary but are not sufficient to generate a spanning tree as tree connectivity is not yet guaranteed. To obtain connectedness, additional flow balance-based constraints are incorporated, and these are described next.

**Connectivity constraints:**

$$\sum_{(s,v) \in \mathcal{E}^d} f_{(s,v)} - \sum_{(v,s) \in \mathcal{E}^d} f_{(v,s)} = |\mathcal{V}| - 1 \quad (4d)$$

$$\sum_{\substack{u \in \mathcal{V}: \\ (u,v) \in \mathcal{E}^d}} f_{(v,u)} - \sum_{\substack{u \in \mathcal{V}: \\ (u,v) \in \mathcal{E}^d}} f_{(u,v)} = -1, \quad \forall v \in \mathcal{V} \setminus \{s\} \quad (4e)$$

$$x_{(u,v)} \leq f_{(u,v)} \leq (|\mathcal{V}| - 1) \cdot x_{(u,v)}, \quad \forall (u, v) \in \mathcal{E}^d. \quad (4f)$$

Constraint (4d) states that the flow emitted by the source is equal to  $|\mathcal{V}| - 1$  and constraint (4e) ensures that each vertex except the source "consumes" one and only one unit of flow, which in turn also guarantees that each vertex is reachable from the source  $s$ . Constraint (4f) allows each arc to carry non-zero flow if and only if it is used in the optimal spanning tree, and the value of this flow cannot exceed the total flow emitted from the source vertex.

**Degree constraints:**

$$\sum_{\substack{u \in \mathcal{V}: \\ (v,u) \in \mathcal{E}^d}} x_{(v,u)} + \sum_{\substack{u \in \mathcal{V}: \\ (u,v) \in \mathcal{E}^d}} x_{(u,v)} - (k+2) \leq d(v) \cdot y_v, \quad \forall v \in \mathcal{V} \quad (4g)$$

Constraint (4g) imposes vertex  $v$  to be a  $k$ -branch vertex if and only if its degree is strictly greater than  $k+2$  in the spanning tree. Note that while the above constraint merely sets the value of  $y_v$  to be greater than or equal to zero if  $d(v) \leq k+2$ , nevertheless, the objective function (which minimizes the sum of the  $y_v$ -variables) drives the value of  $y_v \equiv 0$  at optimality.

## 4 Computational Results

In this section, we describe the computational results obtained by applying the proposed single commodity flow formulation for the  $k$ -MBVST problem on a set of instances, synthetically generated based on the parameters originally proposed in [CCGG13]. We considered nine different values for the number of vertices given by:  $|\mathcal{V}| = \{50, 100, 200, 300, 400, 500, 600, 700, 800\}$ . The number of edges is generated according to the following formula:

$$\lfloor (|\mathcal{V}| - 1) + i \times 1.5 \times \lceil \sqrt{|\mathcal{V}|} \rceil \rfloor, \quad \text{with } i \in \{1, 2, 3\}. \quad (5)$$

For each value of the parameter  $k \in \{0, 1, 2, 3, 4, 5\}$ , we randomly generated 30 instances for each choice of  $|\mathcal{V}|$  and  $i$ . In order to obtain a significant number of branch vertices, the instances generated using (5) are typically sparse graphs. The time limit was set to be 3600 seconds for each instance. All of our computations were performed on an Intel i7 6820HQ 2.7Ghz (with 8 Cores) Windows workstation with 16GB RAM, using C++ as the modeling environment and Cplex 12.7 as the underlying ILP solver.

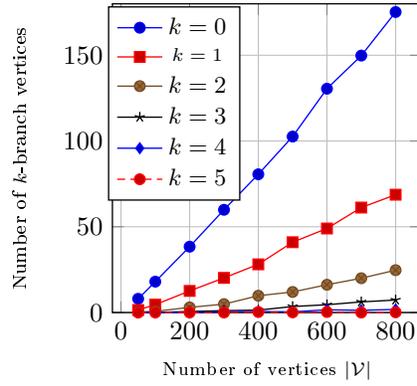
Table 1 displays the optimum number of  $k$ -branch vertices (averaged over all solved instances), the CPU time (seconds), and the number of instances of each type that were successfully solved to optimality (within the specified time limit). From the numerical results recorded in Table 1, it can be observed that the computational time increases along with an increase in the size of the graph and with higher graph density, as the number of decision variables in the ILP are directly correlated to the size and density of the graph. Moreover, as the size of the instance increases, constraint (4f) no longer remains tight, and furthermore, the number of branch vertices reduce in higher density instances ( $i = 2$  and  $i = 3$ ) because the propensity of the graph to be Hamiltonian is also reduced in such cases. Finally, the number of branch vertices increases (almost linearly) with instance size, and this phenomenon is amplified by the fact that the edge generation scheme used in our work makes the density decrease with instance size.

Figure 4 displays the number of  $k$ -branch vertices as a function of the parameters  $k$  and  $|\mathcal{V}|$  for each value of  $i$ . As noted before, the number of  $k$ -branch vertices

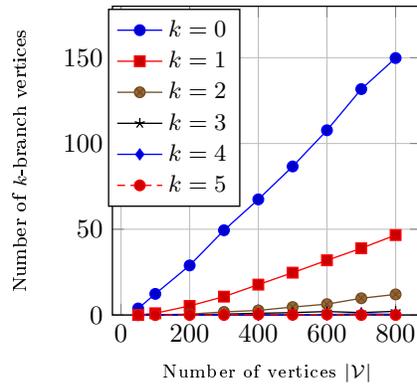
increases as the size of the vertex set increases, but decreases with  $k$  as well as with  $i$ . When  $k \geq 4$ , the number of  $k$ -branch vertices is close to zero regardless of the value of  $|\mathcal{V}|$  and  $i$ .

Instances $ \mathcal{V} $	$k=0$								
	$i=1$			$i=2$			$i=3$		
	Sol	Time	# Inst	Sol	Time	# Inst	Sol	Time	# Inst
50	8.00	0.22	30	3.80	0.60	30	2.50	0.62	30
100	18.00	0.39	30	12.30	1.11	30	7.60	5.44	30
200	38.40	1.14	30	28.90	5.21	30	20.70	18.31	30
300	59.90	1.91	30	49.40	9.73	30	37.90	42.76	30
400	80.60	4.06	30	67.40	15.15	30	53.80	50.98	30
500	102.60	4.32	30	86.60	19.69	30	74.10	446.47	30
600	130.50	5.93	30	107.70	21.81	30	91.22	289.46	29
700	149.80	6.68	30	131.70	40.04	30	113.44	275.39	29
800	175.20	8.73	30	149.80	43.79	30	131.00	188.82	30
$k=1$									
50	1.50	0.10	30	0.00	0.21	30	0.00	0.25	30
100	4.60	0.31	30	1.00	0.46	30	0.50	0.82	30
200	12.60	0.58	30	5.30	1.80	30	2.67	3.73	27
300	20.20	1.72	30	10.70	8.32	30	4.88	9.35	24
400	28.10	3.21	30	17.70	12.19	30	9.50	21.20	25
500	41.00	4.78	30	24.67	16.92	27	14.00	42.85	27
600	49.00	6.16	30	31.89	22.37	27	20.43	108.02	21
700	61.20	8.11	30	38.90	23.84	30	31.00	77.04	6
800	68.70	8.99	30	46.60	40.74	30	34.60	150.90	12
$k=2$									
50	0.30	0.07	30	0.00	0.12	30	0.00	0.17	30
100	0.70	0.16	30	0.10	0.31	30	0.00	0.49	30
200	3.00	0.49	30	0.50	1.45	30	0.33	1.68	27
300	4.90	0.94	30	1.80	3.56	30	0.71	6.19	21
400	9.80	2.09	30	2.67	7.36	27	1.22	15.07	27
500	12.00	2.29	30	4.63	12.04	24	2.50	25.04	30
600	16.20	4.23	30	6.33	24.68	18	3.25	32.69	12
700	20.00	5.56	30	9.75	19.19	12	5.00	118.11	6
800	24.70	5.62	30	12.00	23.90	9	4.33	57.01	9
$k=3$									
50	0.10	0.07	30	0.00	0.10	30	0.00	0.13	30
100	0.10	0.15	30	0.10	0.17	30	0.00	0.27	30
200	0.60	0.32	30	0.10	0.93	30	0.00	1.56	30
300	1.10	0.79	30	0.50	1.63	30	0.10	3.98	30
400	1.40	1.32	30	1.00	4.42	24	0.40	10.80	30
500	3.50	1.62	30	1.38	9.50	24	0.13	22.14	24
600	4.56	3.00	27	2.00	5.59	7	1.00	30.55	21
700	6.20	4.18	30	1.40	53.08	15	0.33	69.49	9
800	7.30	3.04	30	2.14	21.57	20	2.50	57.01	6
$k=4$									
50	0.00	0.05	30	0.00	0.09	30	0.00	0.12	30
100	0.00	0.08	30	0.00	0.13	30	0.00	0.23	30
200	0.00	0.18	30	0.00	0.54	30	0.00	0.35	30
300	0.50	0.32	30	0.14	0.98	21	0.00	1.88	27
400	0.60	0.54	30	0.00	0.91	21	0.00	2.41	24
500	0.40	0.51	30	0.13	5.00	24	0.00	24.68	18
600	1.60	1.15	30	0.20	8.65	15	0.25	50.22	12
700	1.30	2.63	30	0.67	15.77	9	0.00	86.80	9
800	1.80	3.59	30	0.33	16.36	18	0.40	14.20	15
$k=5$									
50	0.00	0.05	30	0.00	0.08	30	0.00	0.10	30
100	0.00	0.09	30	0.00	0.15	30	0.00	0.14	30
200	0.00	0.11	30	0.00	0.42	30	0.00	0.50	30
300	0.00	0.20	30	0.00	1.12	24	0.00	1.08	30
400	0.30	0.34	30	0.00	21.68	24	0.00	4.76	24
500	0.10	0.29	30	0.00	80.04	21	0.00	45.24	18
600	0.30	0.64	30	0.00	6.34	15	0.00	40.58	15
700	0.00	1.83	21	0.00	16.95	8	0.00	42.75	8
800	0.11	2.39	27	0.00	10.39	21	0.00	49.27	6

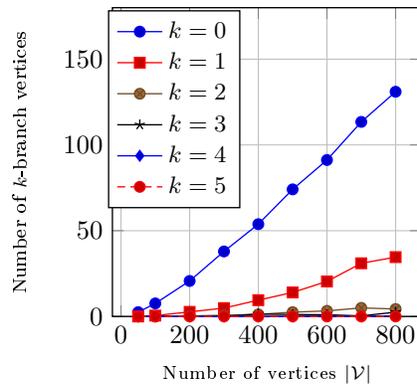
Table 1. Solution value, running time and number of solved instances regarding  $|\mathcal{V}|$ ,  $k$  and  $i$ .



(a)  $i = 1$



(b)  $i = 2$



(c)  $i = 3$

Fig. 4. Variation of  $s_k(\mathcal{G})$  regarding  $|\mathcal{V}|$ ,  $k$  and  $i$ .

## 5 Conclusion

In this paper, we propose a generalization of the well-known MBVST problem by introducing the notion of a  $k$ -branch vertex, which is a vertex with degree strictly greater than  $k + 2$ . Our new parametrized problem ( $k$ -MBVST) aims to find a spanning tree of  $\mathcal{G}$  with the minimum number of  $k$ -branch vertices. For any non-negative integer  $r$ , we proved that it is NP-complete to decide whether a graph can be spanned by a tree with at most  $r$   $k$ -branch vertices, irrespective of the value of  $k$ . Furthermore, we also established that the  $k$ -MBVST is hard to approximate by proving its non-inclusion in the APX class. We also proposed an integer linear programming formulation based on a single commodity flow balance constraints. Tests on sparse random graphs allowed us to evaluate the number of  $k$ -branch vertices in the optimal solution as well as the computational time required to determine the optimum objective function value with respect to the value of  $k$ , the graph size, and the graph density. Our results indicate that the number of  $k$ -branch vertices increases with graph size but decreases with  $k$  as well as with graph density. It was also observed that when  $k \geq 4$ , the number of  $k$ -branch vertices is close to zero, and is independent of the size and density of the graph.

## References

- [CCGG13] F. Carrabs, R. Cerulli, M. Gaudio, and M. Gentili. Lower and upper bounds for the spanning tree with minimum branch vertices. *Computational Optimization and Applications*, 56(2):405–438, 2013.
- [CCR14] 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(3):442 – 453, 2014.
- [CGI09] R. Cerulli, M. Gentili, and A. Iossa. Bounded-degree spanning tree problems: models and new algorithms. *Computational Optimization and Applications*, 42:353–370, April 2009.
- [GHSV02] L. Gargano, P. Hell, L. Stacho, and U. Vaccaro. *Spanning Trees with Bounded Number of Branch Vertices*, pages 355–365. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002.
- [GPMS<sup>+</sup>99] A. Giorgio, M. Protasi, A. Marchetti-Spaccamela, G. Gambosi, P. Crescenzi, and V. Kann. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1999.
- [HGCT02] J. He, S. H. Gary Chan, and D. H. K. Tsang. Multicasting in wdm networks. *IEEE Communications Surveys Tutorials*, 4(1):2–20, 2002.
- [Mar15] A. Marín. Exact and heuristic solutions for the minimum number of branch vertices spanning tree problem. *European Journal of Operational Research*, 245(3):680 – 689, 2015.
- [MDM13a] M. Merabet, S. Durand, and M. Molnár. Exact solution for branch vertices constrained spanning problems. *Electronic Notes in Discrete Mathematics*, 41(0):527 – 534, 2013.

- [MDM13b] M. Merabet, S. Durand, and M. Molnár. Minimization of branching in the optical trees with constraints on the degree of nodes. In *The Eleventh International Conference on Networks - ICN*, pages 235–240, 2013.
- [MJ00] A. Maher and S. D. Jitender. Power-Efficient Design of Multicast Wavelength-Routed Networks. pages 1852–1862, 2000.
- [Mol08] M. Molnár. Hierarchies for Constrained Partial Spanning Problems in Graphs. Technical report PI-1900, 2008.
- [MZQ98] R. Malli, X. Zhang, and C. Qiao. Benefits of multicasting in all-optical networks. 3531:209–220, 1998.
- [SM99] L. H. Sahasrabuddhe and B. Mukherjee. Light trees: optical multicasting for improved performance in wavelength routed networks. *IEEE Communications Magazine*, 37(2):67–73, 1999.