HAL
open science

Generalization of the Minimum Branch Vertices Spanning Tree Problem

Massinissa Merabet, Miklós Molnár

To cite this version:

Massinissa Merabet, Miklós Molnár. Generalization of the Minimum Branch Vertices Spanning Tree Problem. [Research Report] Nanyang Technological University, Singapore. 2016. hal-01403864

HAL Id: hal-01403864

https://hal.science/hal-01403864v1

Submitted on 28 Nov 2016

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.

Generalization of the Minimum Branch Vertices Spanning Tree Problem

Massinissa Merabet [1]

ERIAN
Nanyang Technological University
Singapore

Miklos Molnar [2]

LIRMM
University of Montpellier
Montpellier, France

Abstract

Given a connected graph G, a vertex v of G is said to be a branch vertex if its degree is strictly greater than 2. The Minimum Branch Vertices Spanning Tree problem (MBVST) consists in finding a spanning tree of G with the minimum number of branch vertices. This problem has been well studied in the literature and has applications specially for routing in optical networks. In this paper we propose a generalization of this problem. We introduce the notion of k-branch vertex which is a vertex with degree strictly greater than $k+2$. The parameter k can be seen as the limit of the capacity of optical splitters to divide the light signal. In order to respect as far as possible this limit, we propose to search a spanning tree of G with the minimum number of k-branch vertices (k-MBVST problem). We propose a proof of NP-hardness of this new problem whatever the value of k. We also propose an ILP formulation of the k-MBVST by generalising the MBVST one. Experimental tests on random graphs show that the number of k-branch vertices increases with graph size but decreases with k as well as with the density. They also show that when $k \geq 4$, the number of k-branch vertices is close to zero whatever the size and the density of the tested graph.

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

1 Introduction

Wavelength-Division Multiplexing (WDM) is an effective technique to exploit the large bandwidth of optical fibers to meet the explosive growth of bandwidth demand in the Internet [HGCT02].
Multicast consists in simultaneously transmit information from one source to multiple destinations in a bandwidth efficient way (it duplicates the information only when necessary). From the computational point of view, multicast routing protocols in WDM networks is mainly based on light-trees [SM99]. It requires the intermediate nodes to have the ability to split the input signal to multiple outputs if needed. A light-splitting switch is needed in the optical device to perform such a task. A node which has the ability to replicating any input signal on any wavelength to any subset of output fibers is referred to as a Multicast-Capable (MC) node [MZQ98]. On the other hand, a node which has the ability to tap into the signal and forward it to only one output is called a Multicast-Incapable (MI) node. The light-splitters switchs are rather expensive devices therefore the number of MC nodes should be minimum in the light-tree. L.Gargano et al. expressed in [GHSV02] this aspect by introducing The Minimum Branch Vertices Spanning Tree problem (MBVST) which consists in finding a spanning tree of a graph with the minimum number of branch vertices (vertices with degree strictly grater than 2). This NP-hard and no-APX problem [GHSV02] is extensively studied in the literature. In [CGI09], Cerrulli et al. give the first ILP formulation of this problem based on a single commodity flow to guaranty the connectivity. In [CCGG13] F.Carabbs et al. give two more ILP formulations based respectively on Multi Commodity Flow and Miller-Tucker-Zemlin formulation. They also provide both lower and upper bound for the MBVST using Lagrangian relaxation. In [Mar15], A.Marin presents a branch-and-cut algorithm based on an enforced Integer Programming formulation for the MBVST problem. In [CCR14], C.Cerrone et al. present a unified memetic algorithm for the MBVST, the problem of minimize the degree sum of branch vertices (MDST), and the well known Minimum Leaves Problem. M.Merabet et al. prove in [MDM13b] that the set of optimal solutions for MBVST and the set of optimal solutions for MDST are disjoint. They also propose two variants of them, taking into account the position of MC nodes in the optical network. In [MDM13a], they consider the case where the application do not explicitly impose a sub-graph as solution. A more flexible structure called hierarchy is proposed. Hierarchy, which can

[1] Email: mmerabet@ntu.edu.sg
[2] Email: molnar@lirmm.fr

be seen as a generalization of trees, is defined as a homomorphism of a tree in a graph [Mol08]. Since minimizing the number of branch vertices in a hierarchy does not make sense, they propose to search the minimum cost spanning hierarchy such that the number of branch vertices is less than or equal to an integer R.

The light power in optical networks should be controlled because of the power loss. Indeed, if a light signal is splitted into k copies, the signal power of one copy will be reduced with, at least, a factor of $1/k$ of the original signal power [MJ00]. The more k is high, the more informations are lost. If k is too high then the information can not be read by the destinations. Therefore k can be seen as a tolerance parameter. Let a k-branch vertex be a vertex with degree strictly grater than $k+2$, 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. Notice that the MBVST is a special case with $k = 0$. If the light-tree contains k-branch vertices, an optical amplifier should be installed near each one to guaranty the efficiency of the broadcast/multicast.

In this paper we introduce the k-MBVST problem which, given a positive integer k and a graph $G = (V, E)$, aims to find a minimum spanning tree of G with minimum number of k-branch vertices. I proof of the NP-hardness of the k-MBVST whatever k is given in Section 2. An ILP formulation based on single commodity flow is given in Section 3, and the experimentations on random graphs are done in Section 4.

2 Problem formulation and NP-hardness

The MBVST problem is defined in [GHSV02] as follows :

Definition 2.1 Let $G = (V, E)$ be a graph. The MBVST problem consists in finding a spanning tree T of G such that the number of branch vertices in T is minimum.

Let us define the k-branch vertex :

Definition 2.2 A k-branch vertex is a vertex with degree strictly grater than $k+2$.

The k-MBVST problem which can be seen as the generalization of the MBVST problem is defined as follows :

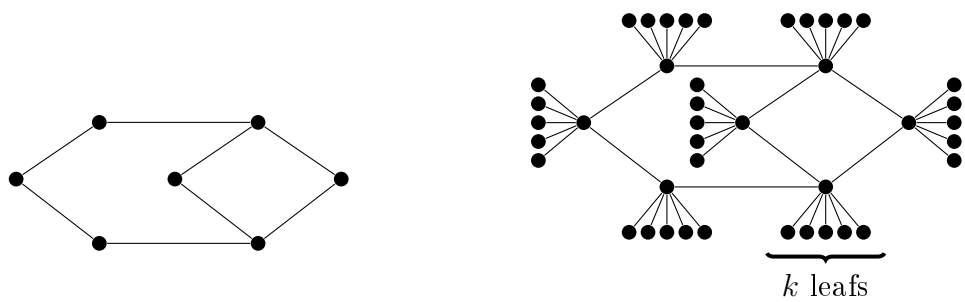
Definition 2.3 Let $G = (V, E)$ be a graph and k be a positive integer. The k-MBVST problem consists in finding a spanning tree T of G such that the number of k-branch vertices in T is minimum. We denote by $s_k(G)$ the smallest number of k-branch vertices in any spanning tree of G.

In a Hamiltonian graph, finding a 0-MBVST is equivalent to finding a Hamiltonian path in G. Thus, The k-MBVST is NP-complete in this case. Furthermore, the classical MBVST problem is NP-complete [GHSV02] and it is a particular case of the k-MBVST problem. Therefore k-MBVST is at least as difficult as the MBVST. We proof in the following that the k-MBVST problem still hard even when k is distant from zero.

Theorem 2.4 *Let r be a fixed non-negative integer. It is NP-complete to decide whether a given graph G satisfies $s_k(G) \leq r$ whatever the value of k.*

Proof

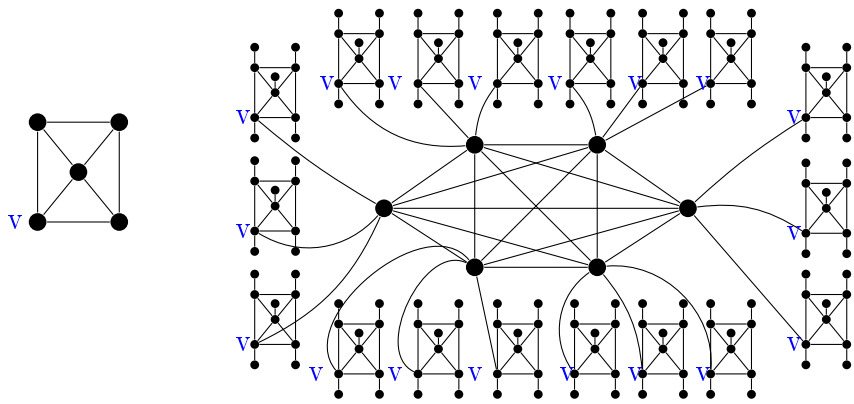
- For $r = 0$: Let $G = (V, E)$ be a given graph. Construct a new graph G' by linking k leafs to each vertex $v \in V$. Decide whatever G' contains a spanning tree with no k-branch vertex is equivalent to decide if G is Hamiltonian.



(a) Graph G (b) Graph G'

Figure 1. Reduction from Hamiltonien problem to 0-MBVST ($k = 5$)

- For $r \geq 1$: Let $G = (V, E)$ be a given graph. We construct a graph G' by duplicating $r \cdot k_k$ times the graph G and adding a complete graph K_r. We choose an arbitrary vertex $v \in V$ and we link every vertex of the graph K to r distinct duplications of G from their vertex v. We link k leafs to each vertex of each duplication of G. In any spanning tree of G', the k vertices of the complete graph are necessary a k-branch vertices. Thus, the graph G' contains a spanning tree with $s_k(G') = r$ if and only if G admits a Hamiltonian path starting from v.



(a) **Graph** G (b) **Graph** G'

Figure 2. Construction of the graph G' for $k = 1$ and $r = 6$.

□

3 ILP formulation of the k-MBVST problem

In this section, we propose an ILP formulation of the k-MBVST problem based on the single flow formulation proposed in [CGI09]. In order to define a spanning tree T of G, we can send from a source vertex $s \in V$ one flow unit to every other vertices $v \in V \setminus \{s\}$. The flow is directed. Thus, the initial graph has to be transformed in a symmetric oriented graph $G^d = (V, E^d)$ where for each edge $\{u, v\}$ in E corresponds two arcs (u, v) and (v, u) in E^d. For each arc $e = (u, v) \in E^d$, we define a variable $f_{(u,v)}$ representing the flow going from u to v. We have a binary decision variable x_e. It is equal to 1 if $f_{(u,v)}$ or $f_{(v,u)}$ carry out a non-zero flow, and zero otherwise. Finally, for each $v \in V$, we have a decision variable y_v that is equal to 1 if v is a k-branch vertex, and 0 otherwise. In the following the linear program is presented.

The objective of our problem is to minimize the number of k-branch vertices belonging to the spanning tree of G. Hence the general objective function can be expressed as follows:

$$Minimize : \sum_{v \in V} y_v \tag{1}$$

This objective function is subject to a set of constraints.

Spanning tree constraints:

$$\sum_{(u,v)\in E^d} x_{(u,v)} = 1 \qquad \forall v \in V \setminus \{s\} \tag{2}$$

$$\sum_{(u,v)\in E^d} x_{(u,v)} = n - 1 \tag{3}$$

Since a vertex with more than one parent creates a cycle, the constraint (2) ensure that each vertex except the source have one and only one predecessor. The number of edges of any tree must be equal to the number of its vertices minus one. The constraints (3) makes sure that exactly $n-1$ arcs are selected in the solution. These two constraints are necessary but not sufficient to have a tree since the connectivity must be ensured. In this purpose, flow based constraints are added.

Connectivity constraints:

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

$$\sum_{(v,u)\in E^d} f_{(v,u)} - \sum_{(u,v)\in E^d} f_{(u,v)} = -1 \qquad \forall v \in V \setminus \{s\} \tag{5}$$

$$x_{(u,v)} \leq f_{(u,v)} \leq (|V| - 1) \cdot x_{(u,v)} \qquad \forall (u,v) \in E^d \tag{6}$$

Constraints (5) ensures that each vertex except the source "consumes" one and only one unit of flow. This constraint also guarantees that each vertex is reachable from the source s. Constraint (4) ensure that the flow emitted by the source is equal to $|V| - 1$. Constraints (6) allows each arc to carry non-zero flow if and only if it is used in the output graph. The value of this flow should not exceed the total flow emitted by the source.

Degree constraints:

$$\sum_{(v,u)\in E^d} x_{(v,u)} + \sum_{(u,v)\in E^d} x_{(u,v)} - k - 2 \leq d(v) \cdot y_v \qquad \forall v \in V \tag{7}$$

Constraints (7) impose vertex v to be a k-branch vertex if its degree is strictly greater than $k+2$ in the tree. Note that the value of y_v is unconstrained if $d(v) \leq k+2$, however in this case it will be set to zero by the objective function.

4 Experimentation

In this section, we describe the computational results that we obtained by applying the proposed single commodity flow formulation (SC) for the k-MBVST to a set of instances generated according to parameters originally proposed in [CCGG13]. In order to obtain a significant number of branch vertices, these instances are considered as spars. We consider 9 different values for the number of vertices of random graph: $|V| = \{50, 100, 200, 300, 400, 500, 600, 700, 800\}$. The number of edges is generated according to the following formula: $\lfloor (|V|-1)+i\times 1.5\times \lceil \sqrt{|V|} \rceil \rfloor$ with $i \in \{1, 2, 3\}$. For each value of the parameter $k \in \{0, 1, 2, 3, 4, 5\}$, we randomly generated 30 instances for each choice of $|V|$ and i. The SC formulation has been coded in C. The IBM ILOG CPLEX 12 solver was used to solve the mathematical formulations, considering a time limit of 1 hour for each instance. All tests have been executed on an Intel i7 6820HQ 2.7Ghz (with 8 Cores) workstation with 16 gigabytes of RAM.

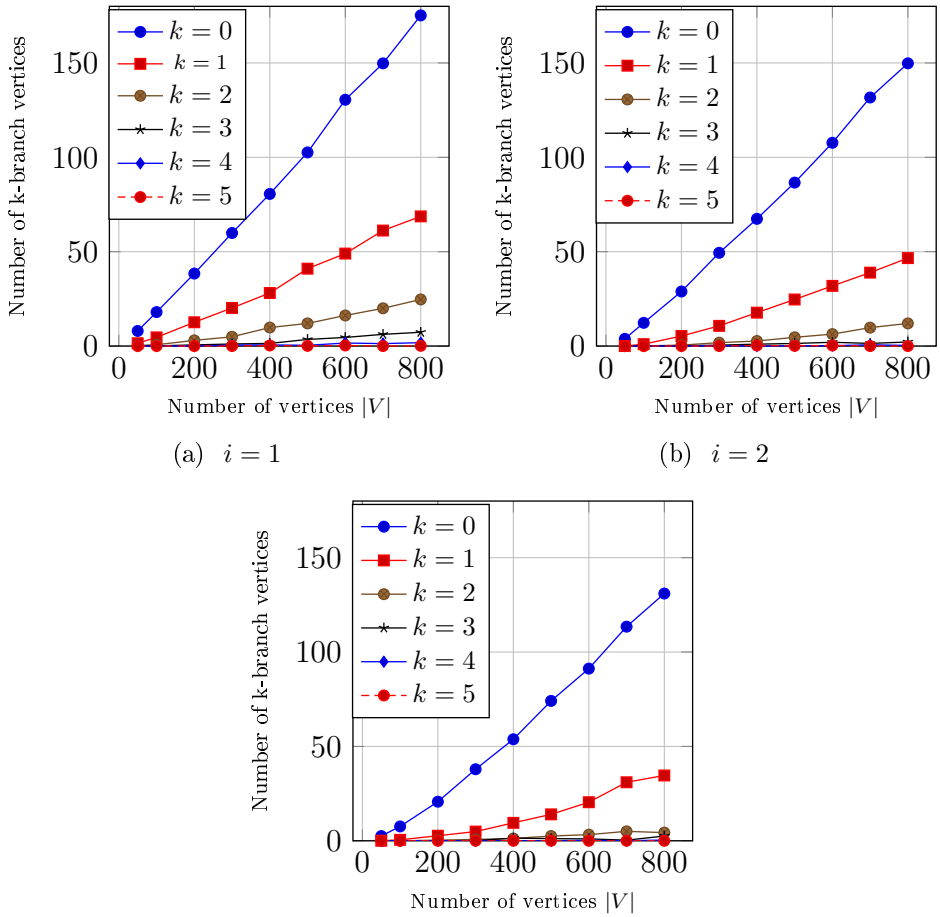
The numerical results are presented in Table 1. Trivially, the computational time increases both with the size of the graph and with de density. Indeed, greater and denser is the instance, higher is the number of decision variables in the ILP. Furthermore, greater is the instance, less the constraint 6 is tight in average. The number of branch vertices decrease in a more dense instances ($i = 2$ and $i = 3$) because more an instance is dense and more farther it is from the Hamiltonicity form. Obviously, the number of branch vertices mechanically increase with the instance size. This is amplified by the fact that our density formula makes the density decrease with the instance size.

Figure 3 shows the number of k-branch vertices regarding the variation of the parameter k and $|V|$ for each value of i. The number of k-branch vertices mechanically increases with $|V|$ but decreases with k as well as with i. When $k \geq 4$, the number of k-branch vertices is close to zero whatever the value of $|V|$ and the value of i. This affirmation can not be confirmed for the unsolved instances but even when the percentage of solved instances is equal to 100% this affirmation still true.

Instances	$i=1$			$i=2$			$i=3$		
$\|V\|$	Sol	Time	# Inst	Sol	Time	# Inst	Sol	Time	# Inst
k=0									
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	09.12	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 $|V|$, k and i.



(a) $i = 1$

(b) $i = 2$

(c) $i = 3$

Figure 3. Variation of $s_k(G)$ regarding $|V|$, k and i.

5 Conclusion

In this paper, we propose a generalization of the well known MBVST problem by introducing the notion of 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 G with the minimum number of k-branch vertices. Let r be a non-negative integer, we proved that it is NP-complete to decide whatever a graph can be spans by a tree with at most r k-branch vertices, whatever the value of k. We also proposed an ILP based on a single flow formulation. Tests on spars random graphs allowed us to evaluate the number of k-branch vertices in the optimal solution and the running time regarding the value of the parameters k, the graph size, and the graph density. The results show that the number of k-branch vertices increases with graph size but decreases with k as well as with the density. They also show that when $k \geq 4$, the number of k-branch vertices is close to zero whatever the size and the density of the tested graph.

References

[CCGG13] 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(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. *Comput. Optim. Appl*, 42:353–370, April 2009.

[GHSV02] L. Gargano, P. Hell, L. Stacho, and U. Vaccaro. Spanning Trees with Bounded Number of Branch Vertices. ICALP '02, pages 355–365, London, UK, 2002. Springer-Verlag.

[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. In *IEEE Journal of Selected areas in communication*, 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. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3531, pages 209–220, October 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, Feb 1999.