



HAL
open science

ILP formulation of the degree-constrained minimum spanning hierarchy problem

Massinissa Merabet, Miklós Molnár, Sylvain Durand

► **To cite this version:**

Massinissa Merabet, Miklós Molnár, Sylvain Durand. ILP formulation of the degree-constrained minimum spanning hierarchy problem. *Journal of Combinatorial Optimization*, 2018, 36 (3), pp.789-811. 10.1007/s10878-017-0159-4 . lirmm-01584405

HAL Id: lirmm-01584405

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01584405>

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.

ILP formulation of the Degree-Constrained Minimum Spanning Hierarchy Problem

Massinissa Merabet^a, Miklos Molnar^b, Sylvain Durand^c

^a*ERIAN, Nanyang Technological University, Singapore
(e-mail: mmerabet@ntu.edu.sg).*

^b*LIRMM, University of Montpellier, France
(e-mail: miklos.molnar@lirmm.fr)*

^c*LIRMM, Montpellier III University, France
(e-mail: sylvain.durand@lirmm.fr)*

Abstract

Given a connected edge-weighted graph G and a positive integer B , the Degree Constrained Minimum Spanning Tree problem (DCMST) consists in finding a minimum cost spanning tree of G such that the degree of each vertex in the tree is less than or equal to B . This problem, which has been extensively studied over the last few decades, has several practical applications, mainly in networks. However, some applications do not especially impose a subgraph as solution. For this purpose, a more flexible so-called hierarchy structure has been proposed. Hierarchy, which can be seen as a generalization of trees, is defined as a homomorphism of a tree in a graph. In this paper, we discuss the Degree Constrained Minimum Spanning Hierarchy (DCMSH) problem which is NP-hard. An Integer Linear Program (ILP) formulation of this new problem is given. Properties of the solution are analysed, which allows us to add valid inequalities to the ILP. To evaluate the difference of cost between trees and hierarchies, the exact solution of DCMST and DCMSH problems are compared. It appears that, in sparse random graphs, the average percentage of improvement of the cost varied from 20% to 36% when the maximal authorized degree of vertices B is equal to 2, and from 11% to 31% when B is equal to 3. The improvement increases as the graph size increases.

Keywords: Degree-Constrained Spanning Problem, Spanning Hierarchy, Spanning Tree, Integer Linear Programming, ILP, DCMSH, DCMST.

1. Introduction

Several network design problems can be modeled as finding a network corresponding to certain connectivity specifications. For instance, the network may require to connect with minimum cost all nodes in a topology graph (spanning tree problem), a specified subset of the nodes (Steiner tree problem) or interconnect a set of set of nodes (generalized Steiner forest problem). Finding a minimum spanning tree of a graph is polynomial [1]. Conversely, finding a spanning tree whose nodes do not exceed a given degree with minimum total edge length is NP-hard [2] and known as the Degree Constrained Minimum Spanning Tree (DCMST) problem. The DCMST problem has many practical applications. For example, to support broadcasts in optical networks where branching nodes (nodes with degree strictly greater than 2) should be equipped with optical splitters, which split the light signal into several copies. To model the limited node splitting capacity, it is necessary to upper bound vertex degrees in the topology graph. So the splitting capacity is represented by a degree constraint. The DCMST problem also arises in many other areas such as in the design of integrated circuits, energy networks, transportation, logistics, sewage networks and plumbing for maximum network reliability [3].

Most of the existing research on degree-constrained spanning problems is based on spanning trees or k -connected spanning subgraphs. In some applications, however, the solution does not need to correspond to

a subgraph. For instance, special optical routes as light-trails can return several times to a switch/link [4]. Further, the degree constraint does not always express the fact that the vertex has a global "budget" to connect neighbouring vertices. This budget approach can be found in [5]. Indeed, the degree constraint can also express the fact that the vertex can perform a given action (a branching) for each of its visit only for a limited number of neighbour vertices. In our paper we suppose that the degree bound expresses this later limited capacity of the vertex for each visit. Moreover, we assume that the limit is the same constant value, valid for all vertices in the graph.

A flexible structure to solve constrained spanning problems is proposed in [6]. In contrast with trees, this structure (called a *hierarchy*) is not a subgraph but rather a homomorphism of a tree in a graph. As we will show in this paper, the advantages of the hierarchy concept for solving degree constrained spanning problems are obvious. Even if there is no spanning tree satisfying the degree constraints (which correspond to a limited capacity of a vertex), the coverage is always possible with a hierarchy. Moreover, the spanning hierarchy always achieves the best solution. In fact, a spanning tree is a special hierarchy.

Note that for some strongly NP-hard problems, there exist relaxations which allow to span vertices more than once. Indeed, in [7], Christofides et al. relax the Capacitated Vehicle Routing Problem (CVRP) by introducing the notion of «*q*-route». A *q*-route is a walk that starts at the depot vertex, traverses a sequence of client vertices with total demand at most C , and returns to the depot. Some clients may be visited more than once. Minimum cost *q*-routes can be found in pseudo-polynomial time by dynamic programming and that allows to obtain a decisive information about the optimal solution since the set of valid CVRP routes is strictly contained in the set of *q*-routes. A similar approach is achieved on the Capacitated Minimum Spanning Tree problem (CMST) in [8], where Uchoa et al. introduce the concept of «*q*-arbs», an arborescence-like structure directed from the root, having degree 1 at the root and with total demand at most C , but allowing some vertices (and even arcs) to appear more than once. The Travelling Salesman Problem (TSP) is also usually relaxed to the shortest Complete Circuit Problem (CCP), i.e. the shortest circuit going through each vertex at least once [9, 10]. The interest of the relaxed problem is to compute lower bounds for the original one, and it can also be considered as a basis of empirically successful heuristics for computing upper bounds and sophisticated reduction techniques, culminating in an exact algorithm which achieves impressive empirical results. Although a hierarchy can be seen as a relaxation (generalization) of tree, it has a real practical interest and it is not only a «tool» to solve the constrained spanning tree problems.

In our study, we investigate the degree constrained minimum spanning hierarchy (DCMSH) problem. This NP-hard problem involves finding a minimum spanning hierarchy in a graph such that the degree of each vertex in the hierarchy is less than or equal to a given integer B . The DCMSH problem is in APX while the DCMST is not [11]. We proposed an approximation offering a guarantee of $\frac{B}{B-1}$ in [12]. The DCMSH is increasingly studied in the literature [13]. In our study, we are focus on the exact resolution of this problem. The first Integer Linear Program (ILP) formulation of this problem is given. This ILP formulation of the DCMSH is not trivial since the hierarchy structure is more complex than trees. Indeed, a hierarchy can return several times to a vertex and the usual cut on flow based approach can not be applied. Properties of the solution are analysed, which allows us to add valid inequalities to the ILP and strongly reduce the combinatory of this problem. This allows the DCMSH problem to be competitive regarding the DCMST one in term of both time resolution and the size of the solved instances. To evaluate the difference of cost between trees and hierarchies, the exact solutions of DCMST and DCMSH problems are compared for a concrete case (NSF network) as well as in random graphs.

The rest of the paper is organized as follows. First, a concise related work is given in Section 2. Then, the degree constrained minimum spanning hierarchy problem is formulated in Section 3. Some useful properties of the optimal solution are proved in Section 4. In Section 5, ILP formulation is developed to compute the optimal hierarchy. Computational results are done in Section 6 to compare optimal hierarchies and optimal trees. Finally the paper is concluded in Section 7.

2. Previous works

The Degree-Constrained Minimum Spanning Tree (DCMST) problem was firstly introduced and investigated in [2] (it is also briefly mentioned in [14]).

Definition 2.1. *Let $G = (V_G, E_G)$ be an undirected connected, edge-weighted graph such that V_G is the set of vertices and E_G the set of edges. Let $C : E_G \rightarrow \mathbb{B}_+^*$ be a function which associates a strictly positive real cost $C(e)$ to each edge $e \in E_G$ and B a positive integer. The DCMST problem consists of finding a minimum cost spanning tree of G such that the degree of each vertex in the tree is less than or equal to B .*

This problem is NP-hard. Solving the DCMST problem with the degree bound B equal to two is equivalent to solving the minimum Hamiltonian path problem. Moreover, by reducing the DCMST problem to an equivalent symmetric traveling salesman problem (TSP), Garey and Johnson [15] proved that this problem is NP-hard for any fixed constant $2 \leq B \leq |V - 1|$. Ravi showed that approximating the DCMST problem within a constant factor of the cost of the optimal tree is NP-hard [11]. In unweighted graphs, Furer and Raghavachari [16] gave an elegant algorithm that returns a spanning tree in which the degree of each vertex is at most $B + 1$, or returns a witness certifying that the degree bounds are infeasible. Goemans proved in [17] that this result can be generalized to weighted graphs. In polynomial time, we can find a spanning tree of maximum degree at most $B + 1$ whose cost is no more than the cost of a minimum cost tree with maximum degree at most B . Note that these results are formulated in the case where the degree bound is homogeneous. When the degree bounds depend on the vertices, Goemans proved that one can find in polynomial time a spanning tree of maximum degree at most $B + 2$ whose cost is no more than the cost of a minimum cost tree with maximum degree at most B . The best result was presented by Singh and Lau in [18]. Their algorithm computes a spanning tree of minimum cost which violates the degree upper-bounds by at most one. Since it is not possible to obtain any approximation algorithm for the original problem, insisting on the satisfaction of all the degree upper bounds, this result is the best possible.

The DCMST problem may not always have a feasible solution (in some cases there is no spanning tree which meets the degree constraint). For instance, with $B = 2$, the DCMST problem can be solved if and only if the graph G has a Hamiltonian path. The instance G on the left of Figure 1 cannot be spanned by a tree with degree bound $B = 3$ because all spanning trees of G may have the degree of vertex b or e equal to 4. More generally, whatever the value of the degree constraint B , there are instances for which there is no solution. Indeed, whatever the value of the degree constraint B , we can consider a star with $B+1$ leaves.

The reality of some routing technologies in networks explicitly imposes neither a tree nor a subgraph as solution. In some cases, cycles may be present in the subgraph used for routing. To solve the routing in the case of optical networks with any splitter, Maher and Deogun [19] proposed to find a trail that starts from the source and visits all destination nodes. This trail corresponds to a walk visiting vertices several times. Molnar [6] introduced a more flexible so-called hierarchy structure, which can correspond to the minimum cost connected spanning structure. The degree constrained minimum spanning hierarchy problem is formulated using this structure in the following.

3. Problem formulation

3.1. Hierarchies

A hierarchy is not necessary a subgraph. It is a graph related structure obtained by a homomorphism of a tree in a graph. Remember that in graphs, a homomorphism can be defined as follows.

Definition 3.1. *Let $Q = (W, F)$ and $G = (V, E)$ be two (undirected) graphs. An application $h : V \rightarrow W$ mapping a vertex in V to each vertex in W is a homomorphism if the mapping preserves the adjacency: $(u, v) \in F$ implies $(h(u), h(v)) \in E$ [20].*

Definition 3.2. If Q is a connected graph which has no vertex with degree greater than two, then (Q, h, G) defines a walk in G . If h is also injective, then the walk is an elementary walk (a path) in G . If several vertices in W correspond to the same vertex in V then (Q, h, G) gives a non-elementary walk in G .

Definition 3.3. If Q is a connected graph without a cycle (a tree) then the triple (Q, h, G) defines a hierarchy in G . Furthermore, if all vertices of G are spanned by Q then the hierarchy H is called spanning hierarchy.

Figure 1 shows an example of a hierarchy. Each vertex of the tree Q is associated with a vertex of the graph G . In reverse direction, some vertices of G are mapped to several vertices in Q . A vertex in Q can be labelled by the vertex in G with which it is associated. To distinguish occurrences related to the same vertex v in G , we use the labels v^1, v^2, \dots, v^k if needed.

If the application h is injective, then the hierarchy corresponds to a tree in G . Using the analogy with elementary and non-elementary walks, a hierarchy without vertex repetition is a tree and a hierarchy that contains several occurrences of some vertices can be considered as a "non-elementary tree".

The degree of a vertex occurrence in a hierarchy has been defined in [21]:

Definition 3.4. Let u be a vertex in Q such that the mapping associates $v \in V_G$ ($h(u) = v$) to this vertex in the hierarchy. The degree $d_H(v^i)$ of the vertex occurrence v^i in H is $d_H(v^i) = d_Q(u)$.

We extend the definition of the function C to associate a cost to each edge belonging to Q . The cost of a hierarchy corresponds to cost sum of the edges belonging to Q .

3.2. Degree Constrained Minimum Spanning Hierarchy problem

When the degree constraint is an instantaneous capacity constraint and the spanning structure can return vertices several times, the spanning problem can be reformulated using hierarchies [21]. The degree constrained minimum spanning hierarchy problem is defined as follows.

Definition 3.5. The DCMSH problem consists of finding a minimum cost spanning hierarchy of G such that the degree of each vertex occurrence in the hierarchy is less than or equal to B . (The cost of a hierarchy $H = (Q, h, G)$ is the sum of the costs of the edges used in H : $C(H) = \sum_{e' \in Q} C(e)$, where $e \in E$ where $e \in E$ is the edge associated with e' in Q).

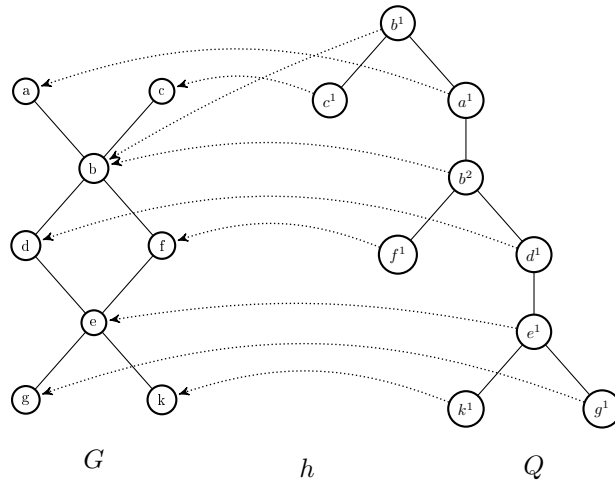


Figure 1: Mapping of vertices for a hierarchy

Figure 1 illustrates how a hierarchy can satisfy the degree constraints. In all connected spanning trees of G , at least one of the vertices b or e have a degree equal to 4. However, the maximum degree of vertices

in $H(Q, h, G)$ is equal to 3 because the vertex b can be visited twice. This case illustrates that there is no feasible spanning tree of G for the degree bounded spanning problem with $B = 3$ but there is at least one feasible spanning hierarchy of G .

Lemma 3.1. *For any degree bound $B \geq 2$, the DCMSH problem always has a solution.*

Proof. A traversal is a particular spanning hierarchy, in which the degree of each vertex occurrence is at most 2. Since a connected graph always has traversals, there are always hierarchies spanning the graph and respecting any degree constraint $B \geq 2$. \square

The problem of the degree constrained minimum spanning hierarchy is NP-hard as it is demonstrated in the following.

Lemma 3.2. *If among all the Minimum Spanning Trees (MST) of a graph G there exists one satisfying the degree constraint, it is an optimal solution for the DCMSH problem and all the optimal solutions are trees in G .*

Proof. Obvious. The minimum cost spanning structure to connect all the vertices without any constraint is the MST, which is connected and does not contain any redundancy. So if one of the MSTs, for instance a tree T^* respects the degree constraint, it is optimal for the spanning problem and also for the DCMSH problem.

Now suppose that an optimal hierarchy $H = (T, h, G)$ exists and it is not a tree in G . Because the MST T^* is an optimal solution of our problem, the cost $c(H)$ of the optimal hierarchy must be the same as the cost $c(T^*)$ of the MST solution. Trivially, the cost of a hierarchy is greater than or equal to the cost of its image in G : $c(I) \leq c(H)$, where I is the image (the subgraph generated by H in G). If H is not a tree, it contains at least a cycle in G (a duplicated edge can also be considered as a cycle). I covers the vertex set V . Two possibilities can arise.

- I is a tree and its cost is lower bounded by the cost of the MST: $c(T^*) \leq c(I)$. In this case, there is at least one duplicated edge in H (remember that H is not a simple tree) and $c(I) < c(H)$. Finally: $c(T^*) < c(H)$ and consequently H can not be optimal.
- I is not a tree. By eliminating some redundancies with non-zero length, a tree T' spanning V is obtained. Trivially, $c(T') < c(I)$ and $c(I) < c(H)$. Finally: $c(T^*) \leq c(T') < c(H)$.

\square

Theorem 3.3. *The problem DCMSH is NP-hard whatever $B \geq 2$.*

Proof. without loss of generality let us propose that $c(e) = 1, \forall e \in E_G$. Let $G' = (V'_G, E'_G)$ be the graph obtained connecting $B - 2$ new vertices of degree 1 to each vertex $v \in V_G$ (Figure 2). We then have $|V'_G| = |V_G| + |V_G| \cdot (B - 2) = (B - 1) \cdot |V_G|$. Every spanning tree of G' has a cost equal to $(B - 1) \cdot |V_G| - 1$. There is a spanning hierarchy of the graph G' for which every vertex respects the degree bound B and of cost $(B - 1) \cdot |V_G| - 1$ if the graph G is Hamiltonian.

Assume that there is a spanning hierarchy H of G' of cost $(B - 1) \cdot |V_G| - 1$. Given its cost, the hierarchy H is necessary a tree. Indeed, if there was at least one duplicated edge in H then the cost of H would be at least equal to $(B - 1) \cdot |V_G|$. If one gets from H the $(B - 2) \cdot |V_G|$ vertices of $V'_G \setminus V_G$, we therefore obtain a connected subgraph in which all vertices have a degree less than or equal to two, which is a Hamiltonian path of G .

Reciprocally, by linking $B - 2$ leaves to each vertex of a Hamiltonian path of G , we get a tree that respects the degree bound B and which is, by considering the lemma 3.2, an optimal solution for DCMSH problem in G' .

\square

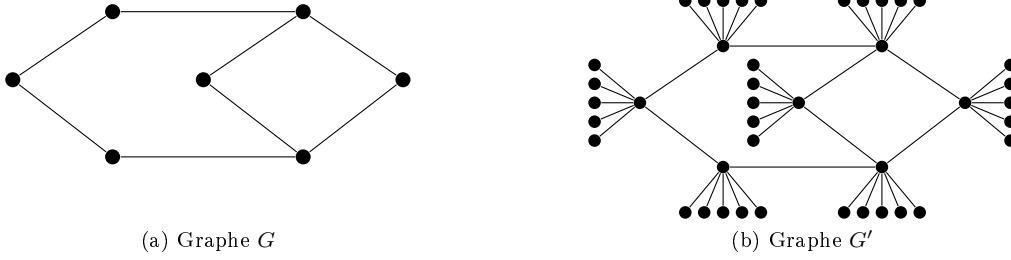


Figure 2: Reduction of DCMSH problem to Hamiltonian problem ($B = 7$)

4. Properties of degree bounded hierarchies

In order to efficiently construct the optimal hierarchy, it is important to study its relevant properties. In this section, we present an upper bound of the vertex occurrences and edge occurrences in the DCMSH (Theorems 4.5 and 4.6 respectively). These properties are directly used in our ILP to compute the solution.

Notations. For any graph $G = (V_G, E_G)$, we denote by $\lambda_m(G)$ the number of vertices of degree m in G . Let $H = (Q_H, h_H, G)$ be a hierarchy in G , we denote by $\theta(v)$ the set of occurrences of v in H . An optimal hierarchy for the problem DCMSH in G is denoted by $H^* = (Q_{H^*}, h_{H^*}, G)$. At first, we study the number of leaves in an arbitrary tree T .

Proposition 4.1. *For any tree $T = (V_T, E_T)$ with $|V_T| \geq 2$, $\lambda_m(T) \geq \alpha \Rightarrow \lambda_1(T) > \alpha * (m - 2) + 2$.*

Proof. By induction.

1. The base case: The proposition is trivially true for $\alpha = 0$ and $|V_T| \geq 2$ since any tree with at least 2 vertices has at least 2 leaves.
2. Suppose that the proposition is true for any $\alpha' < \alpha$. Let $T = (V_T, E_T)$ be a tree with $\lambda_m(T) = \alpha$. Let $v \in V_T$ be a vertex such that $d_T(v) = m$ with $\{s_1, s_2, s_3, \dots, s_m\} \in V_T$ its neighbours. We denote by $F = (T_1, \dots, T_m)$ the forest obtained by deleting v from T and adding a vertex s'_i and an edge $\{s_i, s'_i\}$ to each connected component C_i . Let α_i be the number of vertices of degree m of each connected component C_i .

$$\begin{aligned}
\lambda_1(T) &= \sum_i^m \lambda_1(T_i) - m \\
&> \sum_i^m (\alpha_i * (m - 2) + 2) - m \\
&> \sum_i^m (\alpha_i) * (m - 2) + m \\
&> (\alpha - 1) * (m - 2) + m \quad (\text{because } \alpha - 1 = \sum_i^m (\alpha_i)) \\
&> \alpha * (m - 2) + 2
\end{aligned}$$

□

Proposition 4.2. *For any optimal hierarchy $H^* = (Q_{H^*}, h_{H^*}, G)$ for the DCMSH problem : $\forall v \in V_G$, $\sum_{i=1}^{|\theta(v)|} d_{H^*}(v^i) > B * (|\theta(v)| - 1) + 1$.*

Proof. Proceed by contradiction. Suppose that there is an optimal hierarchy $H^* = (Q_{H^*}, h_{H^*}, G)$ and a vertex $v \in V_G$ such that $\sum_{i=1}^{|\theta(v)|} d_{H^*}(v^i) \leq B * (|\theta(v)| - 1) + 1$. In this case, the number of edges that can be added to the $|\theta(v)|$ occurrences without exceeding the degree constraint is less than or equal to $B - 1$. We direct the hierarchy H^* . For this, we choose an arbitrary root $r \notin \theta(v)$ and we direct H^* from r to the leaves. We can remove one occurrence v^i of v and link at most $B - 1$ successors of v^i to the $|\theta(v)| - 1$ other occurrences of v without exceeding the degree constraint B . When we remove the orientation of arcs, we obtain a feasible hierarchy H' with a cost strictly lower than H^* (H' has at least one less edge than H^*). This is absurd. \square

Proposition 4.3. *There is an optimal hierarchy $H^* = (Q_{H^*}, h_{H^*}, G)$ for the DCMSH problem such that for all $v \in V_G$ with $d_G(v) > B$ and $|\theta(v)| > 1$, there are at least $|\theta(v)| - 1$ occurrences of v with degree equal to B in H^* .*

Proof. We prove that it is possible to transform any optimal hierarchy H^* for the DCMSH problem into an optimal hierarchy H^{**} which respects the conditions of the proposition. Let $H^* = (Q_{H^*}, h_{H^*}, G)$ be an optimal hierarchy for the DCMSH, and let v^i be an arbitrary occurrence of the vertex v . the proposition 4.2 implies that :

$$\begin{aligned} B * (|\theta(v)| - 1) + 1 &< \sum_{j=1, j \neq i}^{|\theta(v)|} d_{H^*}(v^j) + d_{H^*}(v^i) \\ B * (|\theta(v)| - 1) - \sum_{j=1, j \neq i}^{|\theta(v)|} d_{H^*}(v^j) &< d_{H^{**}}(v^i) - 1 \\ &\leq d_{H^*}(v^i) - 2 \end{aligned}$$

Choose an arbitrary root $r \notin \theta(v)$. Direct H^* from r to the leaves. Since H^* is optimal, v^i cannot be a leaf (otherwise it could be deleted, which would decrease the cost), thus $d_{H^*}(v^i) \geq 2$. We can disconnect $B * (|\theta(v)| - 1) - \sum_{j=1, j \neq i}^{|\theta(v)|} d_{H^*}(v^j)$ successors of v^i and connect them to the other occurrences of v without exceeding the degree constraint. The hierarchy H^* thus obtained satisfies the conditions of the proposition and has the same cost as the optimal hierarchy H^{**} . \square

Theorem 4.4. *For any optimal hierarchy $H^* = (Q_{H^*}, h_{H^*}, G)$ with $B > 2$ and for any vertex $v \in V_G$, the number of occurrences of v in H^* is less than or equal to $\frac{|V_G| - 2}{B - 2}$.*

Proof. Arguing by contradiction. Suppose that in the optimal hierarchy $H^* = (Q_{H^*}, h_{H^*}, G)$ there is a vertex $v \in V_G$ for which the number of its occurrences is strictly greater than $\frac{|V_G| - 2}{B - 2}$. By following the same reasoning as in the proof of proposition 4.3, H^* can be transformed into a hierarchy $H^{**} = (Q_{H^{**}}, h_{H^{**}}, G)$ containing the same vertex occurrences but such that there is at least $|\theta(v) - 1|$ occurrences of v with a degree equal to B . We have :

$$\begin{aligned} \lambda_B(H^{**}) &\geq |\theta(v)| - 1 \\ &> \frac{|V_G| - 2}{B - 2} - 1 \\ &\geq \frac{|V_G| - 2}{B - 2} \end{aligned}$$

Using proposition 4.1, we obtain:

$$\begin{aligned}\lambda_1(H^{**}) &\geq \left(\frac{|V_G| - 2}{B - 2}\right) * (B - 2) + 2 \\ &\geq |V_G|\end{aligned}$$

Which is absurd. □

Theorem 4.5. *In any optimal hierarchy $H^* = (Q_{H^*}, h_{H^*}, G)$ for the DCMSH problem with $B > 2$: $|V_Q| < \frac{|V_G| - 2}{B - 2} + |V_G| - 1$.*

Proof. Let $H^* = (Q_{H^*}, h_{H^*}, G)$ be an optimal hierarchy for the DCMSH problem. By following the same reasoning as in the proof of proposition 4.3, H^* can be transformed into a hierarchy $H^{**} = (Q_{H^{**}}, h_{H^{**}}, G)$ containing the same vertex occurrences but such that there is at least $|\theta(v) - 1|$ occurrences of v with a degree equal to B . So :

$$\begin{aligned}\lambda_B(H^{**}) &\geq \sum_{v \in V_G} (|\theta(v)| - 1) + 1 \\ &\geq \sum_{v \in V_G} |\theta(v)| - |V_G| + 1 \\ |V_G| \geq \lambda_1(H^{**}) &\geq \left(\sum_{v \in V_G} |\theta(v)| - |V_G| + 1\right)(B - 2) + 2 \\ \frac{|V_G| - 2}{B - 2} &> \sum_{v \in V_G} |\theta(v)| - |V_G| + 1 \\ \frac{|V_G| - 2}{B - 2} + |V_G| - 1 &> \sum_{v \in V_G} |\theta(v)|\end{aligned}$$

□

The following theorem gives an upper bound for the usage of any edge in the well known minimum Hamiltonian walk problem.

Theorem 4.6. *In an optimal hierarchy for the DCMSH problem with $B = 2$, the number of occurrences of any edge is bounded by 2.*

Proof. Let $H^* = (Q_{H^*}, h_{H^*}, G)$ be an optimal hierarchy for a DCMSH with $B = 2$. We suppose that there is an edge $\{a, b\}$ with at least 3 occurrences in H . Since $B = 2$, H^* is a walk in G . Let us orient H^* from one extremity (randomly chosen) to the other. Let (a^1, b^1) , (a^2, b^2) , and (a^3, b^3) be the 3 first occurrences of the arc (a, b) in this directed walk. We suppose without loose of generality that the edge $\{a^1, b^1\}$ is directed from a^1 to b^1 . Four cases are possible :

1. The arcs (a^1, b^1) , (a^2, b^2) , and (a^3, b^3) are all oriented from a^i to b^i . The oriented walk can be described by $W_1(a^1, b^1)W_2(a^2, b^2)W_3(a^3, b^3)W_4$, where W_i are directed potentially empty sub-walks. Let $H' = W_1(a^1, b^1)W_2\overline{W_3}W_4$ where $\overline{W_3}$ is the directed walk W_3 oriented in the opposite way.
2. The arcs (a^1, b^1) and (a^2, b^2) are oriented from a^2 to b^2 and (a^3, b^3) is oriented from b^3 to a^3 . The oriented walk can be described by $W_1(a^1, b^1)W_2(a^2, b^2)W_3(b^3, a^3)W_4$. Let $H' = W_1(a^1, b^1)W_3W_2W_4$.
3. The arcs (a^1, b^1) , (a^3, b^3) and (a^2, b^2) are oriented from b^i to a^i . The oriented walk can be described by $W_1(a^1, b^1)W_2(b^2, a^2)W_3(a^3, b^3)W_4$. Let $H' = W_1W_3(a^1, b^1)W_2W_4$.
4. The arcs (a^2, b^2) and (a^3, b^3) are oriented from a^i to b^i and (a^1, b^1) is oriented from b^1 to a^1 . The oriented walk can be described by $W_1(a^1, b^1)W_2(b^2, a^2)W_3(a^3, b^3)W_4$. Let $H' = W_1(b^1, a^1)W_2\overline{W_3}W_4$.

In the four cases, we can build a spanning walk H' with a strictly smaller weight than H^* . This contradicts the optimality of H^* . \square

5. Exact solution of the DCMSH problem

As proved in [2], it is NP-hard to find a degree constrained spanning tree. Many integer linear programs (ILP) have been proposed using different models [22, 2]. In [23], the ILP method was successfully applied to search a light-hierarchy structure with optimal cost for all optical multicast routing problems.

In this section, we propose the computation of the DCMSH in two steps: at first, we give the flow based ILP formulation of an equivalent DCMSH in a corresponding multi-graph problem and we compute the optimal flow. Based on this optimal flow, we construct thereafter an optimal hierarchy of the DCMSH problem in the original graph. As well as being optimal for the DCMSH problem, the hierarchy also respects Proposition 4.3.

In our model, the connectivity is preserved by flow formulation. Thus, the initial graph has to be transformed to a symmetric directed multi-graph. We propose to duplicate each edge using arcs in both directions (Figure 4). The number of duplications is bounded regarding Theorem 4.6 if B is equal to 2 and by Theorem 4.4 otherwise. As it is mentioned in Section 2, a hierarchy can be considered as a "non-elementary tree". A flow belonging to a hierarchy can transit more than once between two adjacent vertices in G . In order to properly compute the flow, each of its passage must be done on a distinct arc occurrence of the initial edge. If the flow transits on n occurrences related to an edge, then this edge must appear n times in the final "non-elementary tree". Thus, we introduce a directed multi-graph G_k corresponding to the graph G such that each edge of G is replaced by k arcs in both directions in G_k (k represents the number of duplications of each arc). Then, we formulate an Integer Linear Program (ILP) whose variables corresponding to the vertices and the arcs of the multi-graph and which computes a minimum cost flow in G_k with respect of the degree constraint. This ILP contains valid inequalities corresponding to the properties of the DCMSH. The computed flow induces several optimal hierarchies in G_k . Finally, from the set of solutions obtained in G_k , using a polynomial transformation, an optimal solution for the DCMSH respecting the proprieties of the proposition 4.3 is achieved (we proved that such a solution always exists).

Figure 3 shows the three necessary steps to achieve the exact resolution of the DCMSH problem.

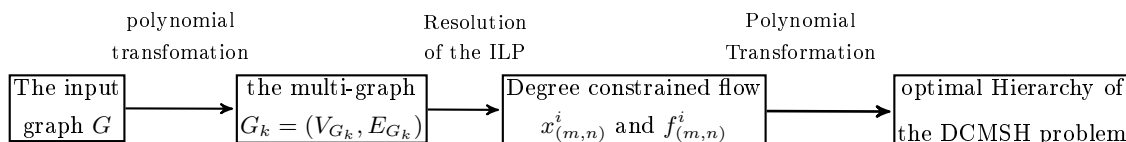
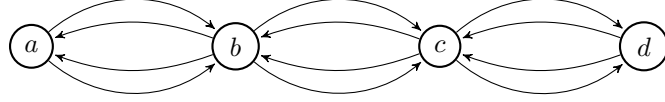
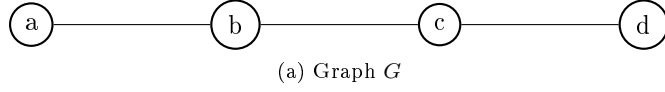


Figure 3: Exact resolution schema of DCMSH problem

5.1. The first step : Construction of the k -graph

Let $G = (V_G, E_G)$ be a connected weighted graph, we construct a directed multi-graph $G_k = (V_{G_k}, E_{G_k})$ such that $V_{G_k} = V_G$ and E_{G_k} contains k arcs in each direction corresponding to each edge of E_G . The number of duplications of each edge is bounded by the result of Theorem 4.6 if B is equal to 2 and by the result of Theorem 4.5 otherwise (Figure 4). Each arc has the same cost than the corresponding edge in G . The interest of this formulation is that a flow which uses an arc only once and respects the degree constraint can be formulated in the multi-graph.



(b) Input instance of our ILP with $B = 2$ obtained by the modification of G

Figure 4: The construction of the k-graph G_k from the graph G

5.2. Second step : Formulation of the ILP

Our model is based on a flow formulation. To ensure the connectivity, one unit of flow is sent from a source chosen arbitrarily to each of the $|V_{G_k}| - 1$ other vertices of G_k . The flow can pass more than once between two adjacent vertices u and v , but every passage of the flow must be done on a distinct occurrence of an arc linking u and v . If the flow transits on n arc occurrences then the n occurrences of the corresponding edge in G must appear in the hierarchy. This model allows to respect the degree constraint. Indeed, each vertex receiving a flow coming from a distinct predecessor is allowed to have at most $B - 1$ successors. Thus, the number of successors of each vertex can at most be equal to the number of its predecessors multiplied by $B - 1$.

In the following the linear program computing the optimal flow respecting the degree constraints is presented.

Network parameters:

- $\Gamma_{(m)}^-$: The set of vertices which has an outgoing arc leading to node m .
- $\Gamma_{(m)}^+$: The set of vertices which can be reached from m .
- $C(m, n)$: The cost of the arc from node m to node n . All duplications of this arc have the same cost and $C(m, n) = C(n, m)$.
- β : Global upper bound on the number of duplications of each arc (Theorem 4.6 if $B = 2$ and theorem 4.5 otherwise).

ILP variables:

- $x_{(m,n)}^i$: Binary variable. Equal to 1 if the occurrence i of the arc (m, n) is in the output solution, 0 otherwise.
- $f_{(m,n)}^i$: Commodity flow variable. Denotes the quantity of flow transiting on the occurrence i of the arc (m, n) .

The objective of our problem is to minimize the total cost of arcs belonging to the hierarchy. Hence the general objective function can be expressed as follows:

$$\text{Minimize : } \sum_{m \in V} \sum_{n \in \Gamma_{(m)}^+} \sum_{i=1}^{\beta} x_{(m,n)}^i * C(m, n) \quad (1)$$

This objective function is subject to a set of constraints, which are listed below.

Degree constraints:

$$\sum_{n \in \Gamma_{(s)}^+} \sum_{i=1}^{\beta} x_{(s,n)}^i \leq B + (B-1) * \sum_{n \in \Gamma^-(s)} \sum_{i=1}^{\beta} x_{(n,s)}^i \quad (2)$$

$$\sum_{n \in \Gamma_{(m)}^+} \sum_{i=1}^{\beta} x_{(m,n)}^i \leq (B-1) * \sum_{n \in \Gamma_{(m)}^-} \sum_{i=1}^{\beta} x_{(n,m)}^i \quad \forall m \in V \setminus \{s\} \quad (3)$$

Constraints (2) and (3) ensure that for each vertex except the source, the number of authorized successors is at most equal to the number of predecessors multiplied by the degree constraint $B - 1$ (each incoming flow can be divided into $B - 1$ outgoing flows). Concerning the arbitrary source, the number of authorized successors is equal to the number of predecessors multiplied by the degree constraint minus 1 plus B because the first occurrence of the source can have at most B successors despite the fact that it has no predecessors.

$$(B-1) * \sum_{n \in \Gamma_{(m)}^-} \sum_{i=1}^{\beta} x_{(n,m)}^i - \sum_{n \in \Gamma_{(m)}^+} \sum_{i=1}^{\beta} x_{(m,n)}^i \leq B-1 \quad \forall m \in V \quad (4)$$

Constraint (4) ensures that each vertex can have k predecessors if and only if this vertex has at least $B * (k - 1)$ successors. This allows construction of an optimal hierarchy such that each vertex v has at most one occurrence with degree strictly smaller than B (Proposition 4.3). This constraint is not indispensable but reduces the branch-and-bound time by eliminating redundant solutions.

$$\sum_{n \in \Gamma_{(m)}^-} x_{(n,m)}^1 \geq 1 \quad \forall m \in V \setminus \{s\} \quad (5)$$

Constraint (5) guarantees that each vertex except the source (which certainly has successors) has at least one predecessor. This ensures that there are no isolated vertices on the output graph (graph generated by the flow).

Connectivity constraints:

In order to guarantee the connectivity of the output graph, in our ILP we have introduced some flow constraints adapted to the specificities of degree constrained hierarchies.

$$\sum_{i=1}^{\beta} \sum_{n \in \Gamma^+(s)} f_{(s,n)}^i = \sum_{i=1}^{\beta} \sum_{n \in \Gamma^-(s)} f_{(n,s)}^i + |V| - 1 \quad (6)$$

The source, like the other vertices of the input graph, can be duplicated in the optimal hierarchy, but only the first occurrence is really a source from where commodity flow is emitted. The other duplications are only relays. For this reason, we permit some occurrences of the "source" vertex to have predecessors. Naturally, the sum of commodity flow outgoing from the source minus the flow incoming to the source is equal to $|V| - 1$, which corresponds to the flow originally emitted from the source.

$$\sum_{i=1}^{\beta} \sum_{n \in \Gamma_{(m)}^-} f_{(n,m)}^i = \sum_{i=1}^{\beta} \sum_{n \in \Gamma_{(m)}^+} f_{(m,n)}^i + 1 \quad \forall m \in V \setminus \{s\} \quad (7)$$

Equation (7) ensures that each vertex except the source consumes one and only one flow. This constraint also guarantees that each vertex is reachable from the source s .

$$f_{(n,m)}^i \geq x_{(m,n)}^i \quad i \in 1..\beta, \forall m \in V \quad (8)$$

$$f_{(m,n)}^i \leq (|V| - 1) * x_{(m,n)}^i \quad i \in 1..\beta, \forall m \in V \quad (9)$$

Constraints (8) and (9) show that each arc should carry non-zero flow if it is used in the output graph, and the value of this flow should not be beyond the total flow emitted by the source.

Valid inequalities:

These constraints do not change the solution of our ILP but improve the solving time for our problem. Indeed, adding these constraints reduces the solving time by 20% to 30% on average.

$$x_{(m,n)}^{i-1} \geq x_{(m,n)}^i \quad i \in 1..\beta, \forall m \in V \quad (10)$$

Constraint (10) is an asymmetry-breaking constraint. It ensures that the occurrence i of the arc (m, n) can be selected in the output graph if and only if the occurrence $i - 1$ of this arc is already selected.

$$\sum_{m \in V} \sum_{n \in \Gamma_{(m)}^-} x_{(n,m)}^1 \geq |V| - 1 \quad \forall m \in V \quad (11)$$

This constraint ensures that the total number of arcs of occurrence 1 is greater than or equal to $|V| - 1$ (trivially, the number of arcs of index 1 is a lower bound of the number of arcs used).

$$\sum_{m \in V} \sum_{n \in \Gamma_{(m)}^+} \sum_{i=1}^{\beta} x_{(m,n)}^i < \frac{|V_G| - 2}{B - 2} + |V_G| \quad (12)$$

According to Proposition 4.5, this constraint ensures that the total number of arcs must be less than $\frac{|V_G| - 2}{B - 2} + |V_G|$.

Notice that in the case where a vertex has more than $B - 1$ successors and more than one predecessor, the successors can be attributed to these occurrences in different ways. In the following, we construct an optimal hierarchy respecting the proposition 4.3.

5.3. Third step : The construction of an optimal hierarchy

We denote by $G_s = (V_{G_s}, E_{G_s})$ the directed graph corresponding to the minimum cost flow. Since all vertices are covered $V_{G_s} = V_{G_k} = V_G$, E_{G_s} contains only the arcs of E_{G_k} traversed by nonzero flow.

Proposition 5.1. *The graph G_s corresponds to the image of the directed DCMSH H_k^* in G_k .*

Proof. Without loss of generality, let us suppose that there is only one DCMSH. Trivially, the image covers V_G . One can prove that E_{G_s} correspond to the arcs in the image of the DCMSH. Let us suppose that there are an arc $e \in E_{G_k}$ such that e has not a correspondence in the optimum. By deleting e , a new graph G'_s is obtained such that $c(G'_s) < c(G_s)$. It is in contradiction with the fact that the computed flow is of minimum cost. Moreover, the deletion of e can implicate the non preservation of the flow. \square

Our third step construct the DCMSH from its image. Notice that in some cases, several DCMSH with the same cost can be built.

In order to construct a directed DCMSH H_k^* in G_k , it is necessary to distinguish between the different vertex duplications. This is possible thanks to the properties of the ILP: indeed, the constraints 2 and 3 ensure that each vertex u with p predecessors is allowed to have at most $p \cdot (B - 1)$ successors. Thus, the vertex u can be duplicated p times and link each duplication to one predecessor among the p existing ones and to $B - 1$ successors among the $p \cdot (B - 1)$ existing ones. Note that the $p - 1$ first duplications of a vertex u have $(B - 1)$ successors and the last duplication has at least one, and at most $(B - 1)$ (Proposition 4.3). Thus, all the vertices of the optimal hierarchy respect the degree bound B . This hierarchy is also connected as the ILP ensure the reception by each vertex of the flow sending from the source.

Figure 5 shows an example for the construction of a directed DCMSH in G_k with $B = 3$. The vertex u of the graph G_k has 4 predecessors, therefore, it is duplicated 4 times in the optimal hierarchy H_k . the 3 first duplications have $B - 1$ successors, and the last one has only one. The successors can be arbitrary attributed to the occurrences of u .

For each vertex $u \in V_{G_s}$, we denote by $d_{G_s}^-(u) = \sum_{v \in V_{G_s}} \sum_{i=1}^{\beta} x_{(v,u)}^i$ the number of predecessors and by $d_{G_s}^+(u) = \sum_{v \in V_{G_s}} \sum_{i=1}^{\beta} x_{(u,v)}^i$ the number of successors. The construction of the optimal solution H^* from the graph G_s is as follows :

1. Each vertex $m \in V_{G_s} - \{s\}$, is duplicated $d_{G_s}^-(u)$ times in H_k^* . The vertex considered as the source in the ILP is duplicated $d_{G_s}^-(s) + 1$ times as the first occurrence has no predecessor.
2. A Breadth First Search from the source s to the leaves is done. To each vertex u of G_s corresponds $d_{G_s}^-(u)$ vertices in H_k^* . Let $\Gamma_{G_s}^+(u)$ be the set of successors of u in G_s , we construct $\Gamma_{H_k^*}^+(u)$ the set of successors of $u^1, u^2, \dots, u^{d_{G_s}^-(u)}$ in H_k^* such that if there are p edges between u and a successors of u then the vertices v^1, v^2, \dots, v^p are added to $\Gamma_{H_k^*}^+(u)$. We divided the set $\Gamma_{H_k^*}^+(u)$ in $d_{G_s}^-(u) - 1$ disjoint sub-sets of size $B - 1$ and one last not empty sub-set of size at most $B - 1$. We link each occurrence of u in H_k^* to all the elements (vertices) of a distinct sub-set. Thus, each vertex of the optimal hierarchy H_k^* has at most $B - 1$ successors and only one predecessor. The degree constraint is then satisfied for all the vertices.

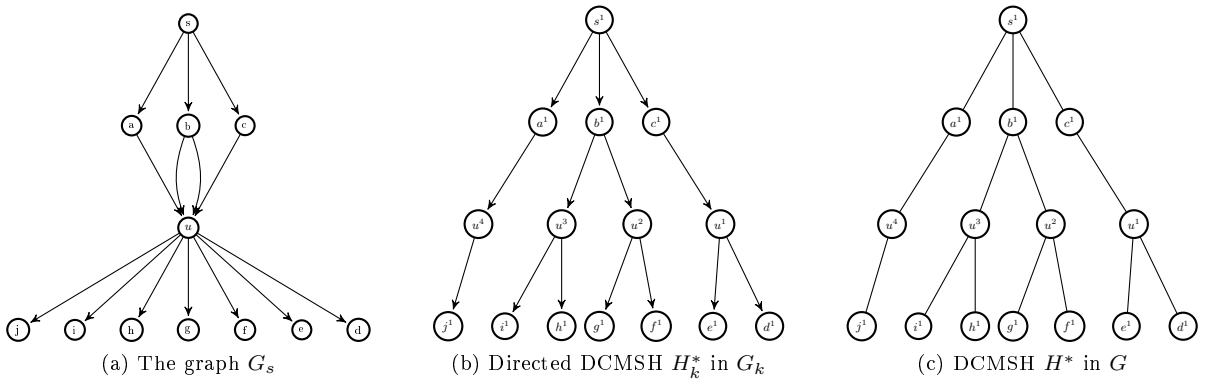


Figure 5: Construction of the optimal hierarchy for the DCMSH problem with $B = 3$ from the graph G_s

Proposition 5.2. *The directed DCMSH H_k^* in G_k corresponds to the DCMSH H^* in G .*

Proof. The directed graph G_k is constructed by the duplication of the edges of G . Each arc in G_k has the same length as the corresponding edge in G . Remember that $G_k = (T_k, h^*, G_k)$. Let T be the tree obtained from T_k by deleting the orientation of its arcs. Since $V_G = V_{G_k}$, the mapping h^* defines a hierarchy (T, h^*, G) . This hierarchy is a DCMSH H^* in G and $C(H_k^*) = C(H^*)$.

Let us suppose that H^* is not a minimal hierarchy respecting the degree constraint but an other hierarchy $H' = (T', h', G)$ is the minimum. Let T'_k an arborescence obtained from T' by the orientation of the arcs from an arbitrary chosen vertex. Using the mapping h' a directed hierarchy $H'_k = (T'_k, h', G_k)$ is obtained in G_k such that $C(H'_k) < C(H_k^*)$, which is in contradiction with the fact that H_k^* is minimal. \square

6. Experimentation

This section aims at illustrating the advantages of the DCMSH compared to the DCMST. Remember that the DCMSH corresponds to the solution of the problem only when the degree constraint is an instantaneous capacity constraint.

6.1. Simulation setup

In order to demonstrate the advantage of the proposed hierarchy structure, a set of computations is conducted to compare it with the spanning tree structure. ILP formulations are implemented in C with the GLPK package [24] by using the 14-node NSF network which is considered to be a realistic case. Moreover, random graphs are generated with NetGen [25] to illustrate general cases. Figure 6a shows the modelling of the NSF network by a weighted graph. The optimal hierarchy in Figure 6c achieves a total edge cost lower than the optimal spanning tree (Figure 6b).

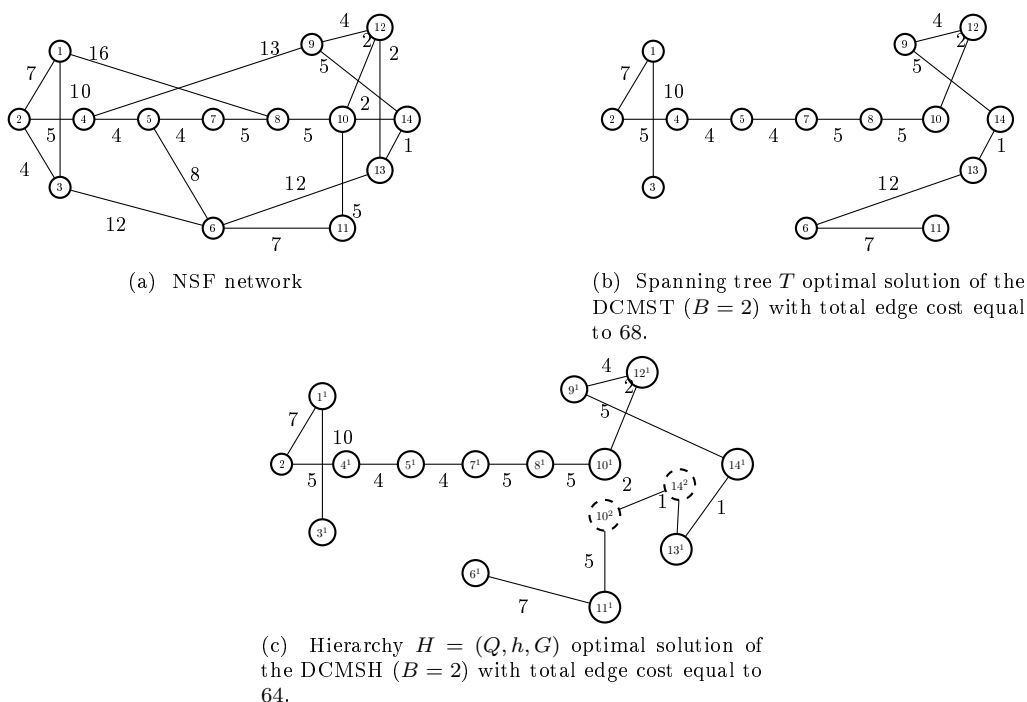


Figure 6: Optimal tree versus optimal hierarchy of the NSF network

We consider six different values for the number of vertices to generate random graphs: $|V_G| \in \{20, 30, 40, 50, 60, 70, 80, 90, 100\}$. We consider a single density value (ratio between the number of edges and the number of vertices) fixed at 2. Graphs with this density are considered as sparse graphs. We suppose that sparse graphs are more interesting for real applications (e.g. networks). In order to have a set of significant tests,

one hundred feasible instances of the DCMST with random edge cost are generated for each value of $|V_G|$. Then ILP formulations are used to compute the optimal degree constrained spanning tree and the optimal degree constrained spanning hierarchy with $B = 2$ and $B = 3$.

6.2. Simulation results

To analyse the results in a meaningful way, it is essential to consider the percentage of infeasible instances for a given scenario with a specified bound B . Note that any instance necessarily admits a feasible solution for the DCMSH problem for any $B \geq 2$. The proportions of infeasible instances for the DCMST problem with degree bounds $B = 2$ and $B = 3$ are presented in Table 1.

$ V_G $	20	30	40	50	60	70	80	90	100
$B = 2$	52.03%	78.81%	89.62%	91.67%	92.05%	94.28%	96.01%	96.89%	98.11%
$B = 3$	9.20%	33.44%	46.82%	55.12%	60.67%	63.07%	65.41%	66.81%	69.03%

Table 1: Proportion of infeasible instances for DCMST

A feasible graph for DCMST with $B = 2$ implies that this graph admits a Hamiltonian path. However, our random graphs are sparse, so the probability of generating a Hamiltonian graph is low. Consequently, the proportions of infeasible instances for the DCMST problem are high and increase with the graph size. All the generated random graphs are feasible for the DCMSH. Thus, a first advantage of hierarchies over trees is clear : contrary to the DCMST, whatever the topology of a connected graph G , this graph admits necessarily a feasible solution for the DCMSH.

$B = 2$									
$ V_G $	20	30	40	50	60	70	80	90	100
Average cost of DCMST	7487	12706	18985	22536	25195	29026	32380	35926	39148
Average cost of DCMSH	5961	9556	13825	15874	17281	19221	21053	23341	24729
Improvement	20.37%	24.79%	27.18%	29.56%	31.41%	33.78%	34.98%	35.03%	36.83%
$B = 3$									
$ V_G $	20	30	40	50	60	70	80	90	100
Average cost of DCMST	6477	10623	14327	18101	21358	23269	25905	28504	30197
Average cost of DCMSH	5671	8725	11408	14006	16967	17282	18641	19924	20696
Improvement	11.80%	17.20%	20.37%	22.62%	24.11%	25.73%	28.04%	30.10%	31.60%

Table 2: Average cost of hierarchies versus trees

The numerical cost results are presented in Table 2. Since an optimal tree for DCMST is a feasible solution for DCMSH, the optimal hierarchy cost is always less or equal to the tree cost (we mentioned the reason). The average percentage of improvement of the cost varied between 20% and 36% when $B = 2$ and between 11% and 31% when $B = 3$. The improvement increases with the graph size. When the graph size is high, there is a high possibility of duplication of vertices in hierarchies, which increases the probability of improvement of the cost. But the improvement decreases as B increases. This is because, when B increases, the probability that the minimum cost spanning tree respects the degree constraint increases. Knowing that the minimum cost spanning tree (MST) without degree constraint can be considered as a lower bound cost for both DCMST and DCMSH, it is normal that the average improvement decreases.

The cost improvement with hierarchies is clearer in Figures 7a and 7b, which represent the average costs of trees and hierarchies for $B = 2$ and $B = 3$, respectively. As shown in these figures, the average hierarchy cost is lower than tree cost in any situation. In both figures, the standard deviation is not very meaningful. Figures 8a and 8b allow more precision to our results. These figures show, for $B = 2$ and $B = 3$, the number of instances for which there is an improvement included in a specific interval concerning each value of $|V|$. When $B = 2$, for $|V| \geq 70$, hierarchies obviously improve the cost by more than 30% for more than 50 instances among 100. The improvement increases as the graph size increases. Indeed, for $|V| = 90$ the hierarchies improve the cost by more than 40% for 75 instances among 100. When $B = 3$, the improvement is less marked. However Figure 8b shows that, for $|V| \geq 70$, hierarchies improve the cost by more than 20% for more than 65 instances among 100. Since all generated graphs have a constant ratio between the number of edges and number of vertices, the number of edges increases with the graph size. The increased

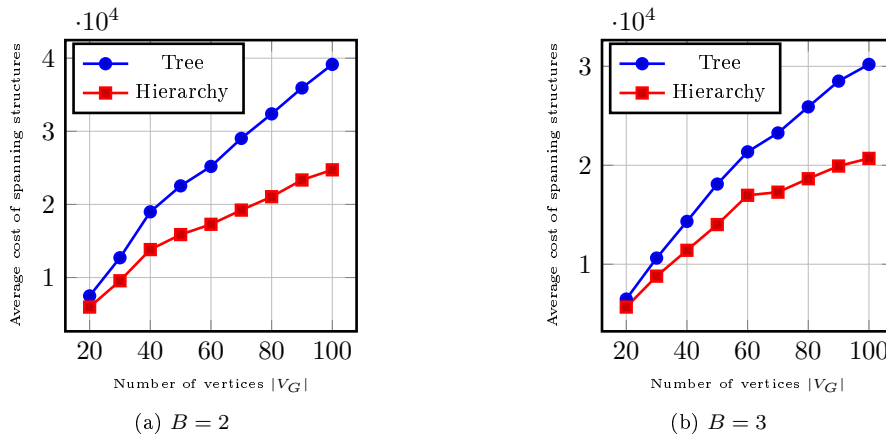


Figure 7: Optimal spanning tree versus optimal spanning hierarchy

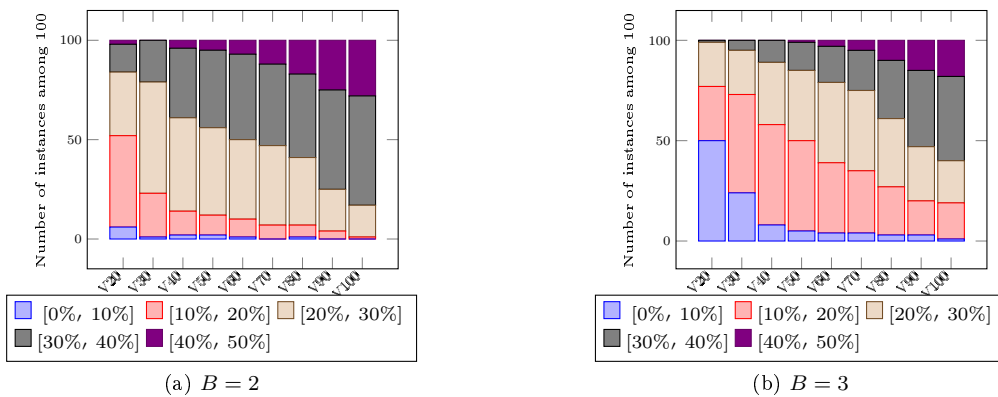


Figure 8: Number of instances regarding intervals of improvement of the cost by hierarchies

improvement is normal since, the number of possible spanning hierarchies of a graph is greater than the number of possible spanning trees (because in hierarchies, edges can be used several times).

7. Conclusion and future works

Our experiments show that when the spanning structure should not necessarily be a subgraph (the degree constraint corresponds to an instantaneous capacity of vertices, the spanning structure can covers vertices and edges several times), hierarchies give the optimal solution. Besides the fact that for any instance and any upper bounds B greater than or equal to 2, there always exists a feasible solution for the DCMSH, the cost of the optimal solution (hierarchy) of the DCMSH problem may be lower than the cost of the optimal solution (tree) of the DCMST problem. This observation is obvious since trees are special cases of hierarchies, but these experiments give a more accurate assessment of the cost improvement. We can note that the average percentage cost improvement varied between 18% and 30% when the maximal authorized degree of vertex B is equal to 2, and between 8% and 22% when B is equal to 3. The improvement increases as the graph size increases. Let us give a more precise example. Hierarchies improve the cost by more than 20% for 90 random generated graphs among 100 with size equal to 40. These experimental results provide clear proof that in a lot of cases spanning hierarchies are good alternatives to spanning trees. Furthermore, in [12] we proved that The DCMSH problem is in APX while the DCMST is not and we proposed an approximation offering a guarantee of $\frac{B}{B-1}$. This ratio is the best possible with a heuristic based only on a minimum spanning tree. Future work will consist in improving the ratio by relaxing the ILP to find a better lower

bound then a spanning tree and showing that the problem is APX-complete (or to find a PTAS). Singh and Lau proved that one can find in polynomial time a spanning tree of maximum degree at most $B + 1$ whose cost is no more than the cost of a minimum cost tree with maximum degree at most B [18] (this result is the best possible assuming that $\mathcal{P} \neq \mathcal{NP}$). It could be interesting to find such a result for the DCMSH problem.

References

- [1] J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, *Proceedings of the American Mathematical Society* 7 (1) (1956) pp. 48–50.
- [2] N. Deo, L. Hakimi, THE SHORTEST GENERALIZED HAMILTONIAN TREE, in: 6th Annual Allerton Conference, Illinois, USA, 1968, pp. 879–888.
- [3] M. Krishnamoorthy, A. T. Ernst, Y. M. Sharaiha, Comparison of algorithms for the degree constrained minimum spanning tree, *Journal of Heuristics* 7 (6) (2001) 587–611.
- [4] F. Bauer, A. Varma, Degree-constrained multicasting in point-to-point networks, in: *Proc. IEEE INFOCOM*, 1995, pp. 369–376.
- [5] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, H. B. Hunt III, Approximation algorithms for degree-constrained minimum-cost network-design problems, *Algorithmica* 31 (1) (2001) 58–78. doi:<http://dx.doi.org/10.1007/s00453-001-0038-2>.
- [6] M. Molnar, Hierarchies for Constrained Partial Spanning Problems in Graphs, Technical report PI-1900 (2008).
- [7] N. Christofides, A. Mingozzi, P. Toth, Exact algorithms for the vehicle routing problem, based on spanning tree and shortest path relaxations, *Mathematical Programming* 20 (1) (1981) 255–282. doi:[10.1007/BF01589353](https://doi.org/10.1007/BF01589353).
- [8] E. Uchoa, R. Fukasawa, J. Lysgaard, A. Pessoa, M. P. de Aragão, D. Andrade, Robust branch-cut-and-price for the capacitated minimum spanning tree problem over a large extended formulation, *Mathematical Programming* 112 (2) (2008) 443–472. doi:[10.1007/s10107-006-0043-y](https://doi.org/10.1007/s10107-006-0043-y).
- [9] P. Miliotis, G. Laporte, Y. Nobert, Computational comparison of two methods for finding the shortest complete cycle or circuit in a graph, *RAIRO-Oper. Res.* 15 (3) (1981) 233–239. doi:[10.1051/ro/1981150302331](https://doi.org/10.1051/ro/1981150302331).
- [10] G. Cornuéjols, J. Fonlupt, D. Naddef, The traveling salesman problem on a graph and some related integer polyhedra, *Mathematical Programming* 33 (1) (1985) 1–27. doi:[10.1007/BF01582008](https://doi.org/10.1007/BF01582008).
- [11] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, H. B. Hunt III, Many birds with one stone: multi-objective approximation algorithms, in: *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing, STOC '93*, ACM, New York, NY, USA, 1993, pp. 438–447.
- [12] M. Molnar, S. Durand, M. Merabet, Approximation of the degree-constrained minimum spanning hierarchies, in: M. M. Halldorsson (Ed.), *Structural Information and Communication Complexity: 21st International Colloquium, SIROCCO 2014*, Takayama, Japan. *Proceedings*, Springer International Publishing, 2014, pp. 96–107. doi:[10.1007/978-3-319-09620-9](https://doi.org/10.1007/978-3-319-09620-9).
- [13] M. Singh, R. Zenklusen, *k*-Trails: Recognition, Complexity, and Approximations, Springer International Publishing, Cham, 2016, pp. 114–125. doi:[10.1007/978-3-319-33461-5_10](https://doi.org/10.1007/978-3-319-33461-5_10).
- [14] A. K. Obruca, Spanning tree manipulation and the travelling salesman problem, *The Computer Journal* 10 (4) (1968) 374–377.
- [15] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman & Co, New York, NY, USA, 1979.
- [16] M. Furer, B. Raghavachari, Approximating the minimum degree spanning tree to within one from the optimal degree, in: *Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms, SODA '92*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992, pp. 317–324.
- [17] M. Goemans, Minimum bounded degree spanning trees, in: *Foundations of Computer Science, 2006. FOCS '06. 47th Annual IEEE Symposium on*, 2006, pp. 273–282.
- [18] M. Singh, L. C. Lau, Approximating minimum bounded degree spanning trees to within one of optimal, in: *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, STOC '07*, ACM, New York, NY, USA, 2007, pp. 661–670.
- [19] A. Maher, S. Jitender, Power-Efficient Design of Multicast Wavelength-Routed Networks, in: *IEEE Journal of Selected areas in communication*, 2000, pp. 1852–1862.
- [20] P. Hell, X. Zhu, Homomorphisms to oriented paths, *Discrete Mathematics* 132 (1–3) (1994) 107 – 114.
- [21] M. Molnar, hierarchies to solve constrained spanning problem, Private communication.
- [22] R. Cerulli, M. Gentili, A. Iossa, Bounded-degree spanning tree problems: models and new algorithms, *Comput. Optim. Appl.* 42 (3) (2009) 353–370.
- [23] F. Zhou, M. Molnar, B. Cousin, Light-hierarchy: the optimal structure for multicast routing in wdm mesh networks, in: *ISCC*, 2010, pp. 611–616.
- [24] A. Makhorin, GNU Linear Programming Kit (GLPK) v 4.38, gnu project Edition (May 2009).
- [25] D. Klingman, A. Napier, J. Stutz, A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems, *Management Science* 20 (5) (1974) 814–821.