Degree-Constrained Minimum Spanning Hierarchies in Graphs

Miklós Molnár

algorithms

Article

Degree-Constrained Minimum Spanning Hierarchies in Graphs

Miklos Molnar

LIRMM, University of Montpellier, CNRS, 161 rue Ada, CEDEX 5, 34095 Montpellier, France; molnar@lirmm.fr

Abstract: The minimum spanning tree problem in graphs under budget-type degree constraints (DCMST) is a well-known NP-hard problem. Spanning trees do not always exist, and the optimum can not be approximated within a constant factor. Recently, solutions have been proposed to solve degree-constrained spanning problems in the case of limited momentary capacities of the nodes. For a given node, the constraint represents a limited degree of the node for each visit. Finding the solution with minimum cost is NP-hard and the related algorithms are not trivial. This paper focuses on this new spanning problem with heterogeneous capacity-like degree bounds. The minimum cost solution corresponds to a graph-related structure, i.e., a hierarchy. We study the conditions of its existence, and we propose its exact computation, a heuristic algorithm, and its approximation.

Keywords: graph theory; degree constrained spanning trees; spanning hierarchies; homomorphism; NP-hard problem; exact computation; heuristic algorithm; approximation

1. Introduction

It is important to cover node sets in graphs with minimum cost in several domains, for instance, for efficient broadcast/multicast communications in networks. The subgraph that spans the set of nodes with minimum cost is a minimum spanning tree (MST), and several polynomial-time algorithms to compute it are known. The partial spanning with minimum cost (the Steiner problem in graphs) is NP-hard. Some applications must meet additional constraints. We are interested in the degree-constrained spanning problem. Each node $v \in V$ of the graph $G = (V, E)$ is assigned a positive integer value $D(v)$ which represents the maximum degree of v in any spanning structure. This degree is potentially different from the degree $d(v)$ of $v \in V$ ($D(v) \leq d(v)$). In the literature, proposals can be found to span the node set of a graph adhering to *budget* type degree bounds [1,2]. In these problems, nodes have limited budgets and cannot exceed the given limits in the spanning structures (in the spanning trees). Degree Constrained Minimum Spanning Tree (DCMST) problems are hard to solve, and unfortunately, constant factor approximations do not exist for them [3].

Degree bound does not always correspond to a definitive budget but to a momentary limited *capacity*. For example, in optical networks, the capacity of duplication of the switches may be limited for each incoming light, but a wavelength can pass through a switch many times (cf. [4]). As a result, the optical broadcast/multicast route may be different from a spanning tree. In [5], a special walk returning to some nodes is proposed when spanning trees are impossible due to the absence of nodes having a degree bound greater than two. If the constraints can be greater than two, branching nodes are possible, and the spanning structure could be a hierarchy crossing certain nodes several times. For the case of homogeneous capacity constraints, a hierarchy-based optimum and its approximation have been proposed in [6]. The analysis of this NP-hard problem shows that this kind of solution still exists and can be approximated. The most important results of the related work can be found in Section 3.

Here, we propose the analysis of cases presenting heterogeneous momentary capacity constraints.

check for updates

Citation: Molnar, M. Degree-Constrained Minimum Spanning Hierarchies in Graphs. *Algorithms* **2024**, *17*, 467. https://doi.org/10.3390/a17100467

Academic Editor: Roberto Montemanni

Received: 16 August 2024
Revised: 16 October 2024
Accepted: 18 October 2024
Published: 21 October 2024

We are studying the following questions: In which cases does the solution exist? How can the optimum be calculated? Can it be approximated?

As in the case of homogeneous capacity bounds, the optimal solution to the problem is a *hierarchy*. As a new result, we formulate necessary and sufficient conditions for its existence. The problem is NP-hard, and we highlight that, in some cases, the optimal solution can be approximated. We propose a special heuristic algorithm for this. In the following section, we present the definition of hierarchies. Section 3 offers a quick overview of DCMST issues, followed by the formulation of capacity-constrained spanning problems. The necessary and sufficient conditions for a solution to exist are formulated in Section 4. An ILP-based exact computation (cf. Section 5.1) and a heuristic algorithm (cf. Section 5.2) are also proposed. Section 6 presents some approximations. Performance is compared in Section 7.

2. Notations and Definitions

The most important notations used in the paper are in Table 1.

Table 1. Notations.

$G = (V, E)$	: undirected graph given by node set V and edge set E
$\{n, m\} \in E$	: edge between nodes n and m
$c(n, m)$	: cost associated with the edge $\{n, m\}$
$d(v)$	: degree of the node $v \in V$ in G
$D(v)$	: degree bound of $v \in V$ (positive integer)
M	: the highest degree bound in V
$V_m \subseteq V$	: the subset of nodes with degree bound m
$T = (P, F)$	: tree with node set P and edge set F
$H = (T, h, G)$	: hierarchy defined by a homomorphic function h between tree T and graph G
$\Theta(v)$	: the set of nodes in P (in T) corresponding to $v \in V$
v^i	: i-th occurrence of $v \in V$ in a hierarchy (in $\Theta(v)$)
$G_A = (V, A)$	: directed graph, equivalent to G, two arcs $(a, b) \in A$ and $(b, a) \in A$ correspond to the edge $\{a, b\} \in E$ of G
$\Gamma^-(a)$	: the set of predecessor nodes at the node a in G_A
$\Gamma^+(a)$	: the set of successor nodes at the node a in G_A

A brief review of the hierarchy concept corresponding to the minimum cost solutions follows.

Hierarchies

The generalization of trees is based on graph homomorphism. The homomorphic mapping of nodes between a tree T and a graph G can be used to define an eventually non-elementary tree in G. To simplify, let us suppose undirected graphs but the definition in digraphs is trivial.

Definition 1 (Hierarchy)**.** *Let $G = (V, E)$ be an arbitrary graph and let $T = (P, F)$ be a tree. Let $h : P \to V$ be a homomorphic function that associates a node $v \in V$ to each node $p \in P$. The application (T, h, G) defines a hierarchy in G.*

Definition 2 (Image of a hierarchy)**.** *The sub-graph defined by the set of nodes and edges in G associated by h to T is the image of the hierarchy.*

Since a node $v \in V$ can be associated with several nodes in P, a hierarchy can contain multiple occurrences of a node and an edge. A hierarchy can be considered as a "non-elementary tree" in G. Certain nodes in the hierarchy can eventually be branching nodes; they are the node occurrences corresponding to the branching nodes of the tree T. Figure 1 illustrates with discontinuous lines a hierarchy in a graph and the associated tree. The labels of the nodes in T are the labels of the corresponding nodes in G. Dotted lines with arrows indicate the associations between these nodes. Some nodes of G (namely the nodes c and d) participate on different levels of the hierarchy as shown in the labeled tree T.

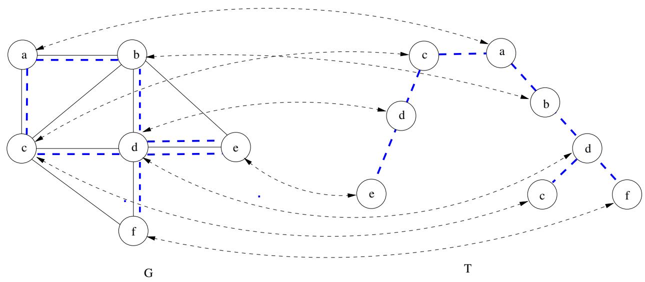


Figure 1. Illustration of a hierarchy with discontinuous lines in an undirected graph G and the corresponding labeled tree T.

Hierarchies generalize trees. Trees are special hierarchies without repetition of nodes (when the mapping h is injective), and therefore, inherit the properties of hierarchies. Note that the image of a hierarchy can contain cycles in G but the hierarchy itself preserves the structure of a tree. Hierarchies allow the exact definition of the optimum in the case of capacity-like constraints.

3. Degree-Constrained Spanning Problems

We propose a clear distinction between two categories of degree constraints:

- *budget*-like constraints: in this case, the total number of adjacent nodes of the constrained node is limited,
- *capacity*-like constraints: the number of adjacent nodes is limited *for each occurrence* of the node.

3.1. Minimum Spanning Trees Under Budget-Like Degree Constraints

Any minimum cost solution of spanning problems under budget-like degree constraints, if it exists, is a tree. Trivially, cycles are useless in any optimum. Deleting an edge from a cycle, the node set remains covered. The basic problem was formulated as follows.

Definition 3 (Degree-constrained minimum spanning tree (DCMST) problem)**.** *In an undirected graph G with non-negative costs $c(e)$ on the edges and with positive integer degree bounds $D(v)$ the problem consists in finding a spanning tree in which the degree of any node v is at most $D(v)$ and the total cost is minimized.*

The degree bounds can be uniform or not in the node set.

The DCMST problem is NP-hard [7], and there is not always a solution. As observed in [8], a spanning tree can exist if and only if $\sum_{v \in V} D(v) \geq 2 \cdot |V| - 2$. The hardness of the problem has also been demonstrated using a reduction from the traveling salesman problem. In [3], the authors present the problem as a generic bi-criteria optimization where the first objective corresponds to the respect of the budget (degree) constraints and the second to the cost minimization. The paper also specifies the sub-graphs to solve the problem. The investigated classes are spanning trees, Steiner trees, and generalized Steiner trees. The authors give some negative results on the approximations. There is no polynomial-time ρ-approximation algorithm for the DCMST problem minimizing the cost when respecting the degree constraint, even when the same upper bound is supposed for all nodes. Heuristic solutions were proposed in several works (cf. [2,3,9–13]). In unweighted graphs constrained by a uniform degree bound MB, an $(1, MB + 1)$-approximation algorithm has been proposed by Fürer and Raghavachari [14]. The results are generalized to weighted graphs in [13].

An interesting idea to compute the solution can be based on uniformly generated spanning trees (cf. [15]) followed by the selection of the tree satisfying the constraints. In [16] the DCMST problem in uncertain networks is discussed. The nondeterministic model mainly concerns the edge costs. Note that uncertainties can characterize the degree bounds.

In the mentioned papers, the constraints are *budget* constraints, and spanning trees are the solutions. In the case of degree bounds corresponding to *capacities* of the nodes, we suppose that the solution can contain several times any node, and the capacity of each occurrence of a node is the same. The model has been presented in [6] for uniform degree bounds.

3.2. Minimum Spanning Problem Under Capacity-Like Degree Constraints

Definition 4 (Node capacity constrained minimum spanning problems)**.** *In a graph G, the degree bounds $D(v)$ associated with nodes $v \in V$ represent the maximum degree capacities. The problem consists of finding the minimum cost solution spanning the node set V such that the degree constraints are respected for each occurrence of the nodes.*

Property 1. *The minimum cost solution spanning the set of nodes of a connected graph under limited node capacities is a hierarchy.*

Proof. Let us assume the solution exists (the reader can find the conditions in Section 4). Since homomorphism preserves adjacency, any solution can be given by a homomorphism-based triplet (S, h, G). For a connected solution, S must be a connected graph. Suppose there is an optimal solution respecting the degree constraints, which is not a hierarchy. In this case, S is not a tree but another connected graph. Therefore, it contains cycles. Trivially, some edges can be removed from the cycles in S without removing nodes thus preserving the node coverage in G with a lower cost. In the minimum cost solution, S is a tree, and the triplet defines a hierarchy. □

A degree-constrained minimum spanning hierarchy (DCMSH) is illustrated in Figure 2. The cost of any edge in G is unitary, and the capacity constraints are indicated. Trivially, no DCMST exists but there is a spanning hierarchy satisfying the constraints. It contains the node b twice, and each occurrence of this node respects the degree constraint as shown by the labeled tree T of the solution. The image of the solution is the graph G itself.

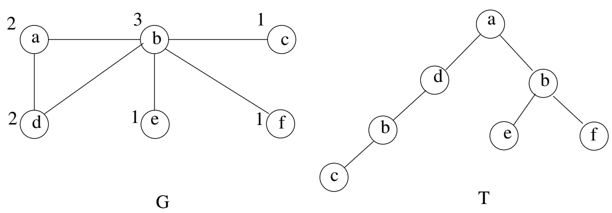


Figure 2. A DCMSH represented by its labeled tree T in the graph G.

The case of uniform capacity bounds and its approximation have been presented in [6]. Spanning hierarchies satisfying homogeneous degree constraints always exist. An ILP-based exact computation of the optimum can be found in [17]. The approximation has been improved in [18].

Since the problem of uniform degree constraints is NP-hard (cf. [17]), the more general problem of non-uniform constraints is NP-hard.

The following section discusses the conditions for the existence of a feasible solution in the case of non-uniform bounds.

4. Necessary and Sufficient Conditions for the Existence of a Solution

Degree-constrained spanning hierarchies do not always exist in the case of heterogeneous degree bounds. In this section, we formulate conditions for their existence.

Remember, we suppose a connected graph and $V_m \subseteq V$ indicates the subset of nodes with degree bound m. Trivially, the nodes in V_1 must be leaves in any solution.

Lemma 1. *If the specified leaf node set V_1 is empty, hierarchies spanning V and satisfying the degree constraints exist.*

Proof. A simple solution can be constructed as follows. Let W be a walk following an MST in G. In W each node occurrence has a degree of at most two. The degree constraints are satisfied and the node set is covered. □

Trivially, Lemma 1 gives a sufficient condition. The complementary case is when V_1 is not empty. The necessary and sufficient conditions for this case are discussed next.

Lemma 2. *If V_1 contains a separator (removing a separator makes a connected graph disconnected), there is no hierarchy spanning V and satisfying the degree constraints.*

Proof. Let A and B be two non-empty subsets of nodes separated by a separator $S \subseteq V_1$. Any path from a node in A to a node in B must pass through S but nodes in S can not be internal nodes in any path. Trivially, no paths exist between nodes in A and nodes in B satisfying the degree constraints in V_1. □

A necessary condition can be formulated: V_1 must be free of any separator. If V_1 is not a separator but a subset of V_1 is a separator, this later isolates some nodes in V_1. Consequently, a spanning hierarchy satisfying the degree constraints does not exist. Figure 3 illustrates this case. Here, $V_1 = \{b, c, d\}$ is not a separator but the subset $\{b, d\}$ is.

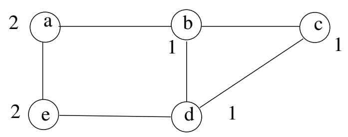


Figure 3. V_1 is not a separator, but contains a separator.

Suppose there is no separator in V_1. Lemmas 3 and 5 indicate that a set V_1 without a separator is necessary but not sufficient for the existence of a spanning hierarchy satisfying the degree constraints. Further sufficient conditions are given by Lemmas 4, 6 and 7.

Lemma 3. *If there is no node with a degree bound greater than 2 and there are more than two nodes with a degree bound 1 ($|V_1| > 2$), there is no hierarchy spanning V and satisfying the constraints.*

Proof. Each node in V_1 must be a leaf in the spanning hierarchy (T, h, G) (and trivially in the corresponding tree T). Since $|V_1| > 2$, the tree T and the hierarchy must have more than two leaves. Because $V_d = \emptyset, \forall d > 2$, the degree of internal nodes in T is equal to 2. T is a path and as such it can not have more than two leaves. □

Lemma 4. *If $|V_1| \leq 2$, hierarchies spanning V and satisfying the constraints exist.*

Proof. Suppose there are two nodes a and b with degree bound 1. A complete graph MC can be constructed in which the edges represent the shortest paths in G that contain neither a nor b as an internal node. These paths exist because V_1 does not contain any separators.

In MC, Hamiltonian paths exist whose endpoints coincide with a and b. Each Hamiltonian path corresponds to a Hamiltonian walk (a Hamiltonian walk covers the node set and can return several times to certain nodes) in the original graph covering the node set and respecting the degree constraints. These walks are solutions. The same demonstration is trivial with only one node in V_1. In this case, one end of the Hamiltonian paths can be freely chosen in MC. □

Lemma 5. *Let us suppose that* $|V \setminus V_1| = 1$ *and let* v_b *the node in* $V \setminus V_1$*. If the degree bound* $D(v_b) < |V_1|$*, there is no hierarchy spanning V and respecting the constraints.*

Proof. Since there is no separator in V_1, the nodes in V_1 are connected to v_b. (suppose that a node $a \in V_1$ is not connected to v_b but to another node $b \in V_1$. b separates a from the other nodes) The nodes in V_1 must be leaves in any spanning hierarchy. Since there is no return from leaves, there is only one occurrence of v_b in any hierarchy. This occurrence of v_b can have at most $D(v_b)$ neighbors, but $|V_1| > D(v_b)$. □

Lemma 6. *If* $|V \setminus V_1| = 1$*, and* $D(v_b) \geq |V_1|$ *of the node* $v_b \in V \setminus V_1$*, a spanning hierarchy respecting the degree constraints exists.*

Proof. The graph G contains a star with a central node $D(v_b) \geq |V_1|$ and the nodes in V_1 are connected to it (there is no separator in V_1). Since $D(v_b) \geq |V_1|$, and the star satisfies the degree constraints. □

Lemmas 3 and 5 indicate cases where there are several nodes in V_1 and there is no spanning hierarchy respecting the degree constraints. The following lemma formulates a sufficient condition.

Lemma 7. *If* $|V \setminus V_1| \geq 2$ *(there are at least two nodes with a degree bound greater than* 1*) and* $|V \setminus \{V_2 \cup V_1\}| \geq 1$ *(there is at least one node with a degree bound greater than* 2*), a spanning hierarchy respecting the degree constraints exists.*

Proof. Let b be a node with $D(b) \geq 2$, and another c with $D(c) > 2$. A degree-constrained spanning hierarchy can be constructed as follows. Because there is no separator in V_1, each node in $V \setminus (\{b\} \cup \{c\})$ can be connected to c with a path satisfying the degree constraints. The set of paths can be partitioned such that there are at most $D(c) - 2$ paths in every group. Since one endpoint of the paths is the node c, each group corresponds to a "spider" respecting the degree constraints and having an occurrence of c as the central node. The different occurrences of c can be chained by a walk visiting the node b (having a degree bound at least 2) such that the degree of each central node is at most $D(c)$. The obtained structure is a degree-constrained hierarchy spanning the node-set. □

Figure 4 illustrates a spanning hierarchy corresponding to Lemma 7. The occurrences of the node c are the central nodes of spiders which are connected by paths passing through the node b. Notice that it is not the "best" spanning hierarchy. The reader can easily find hierarchies containing fewer edges while respecting the constraints.

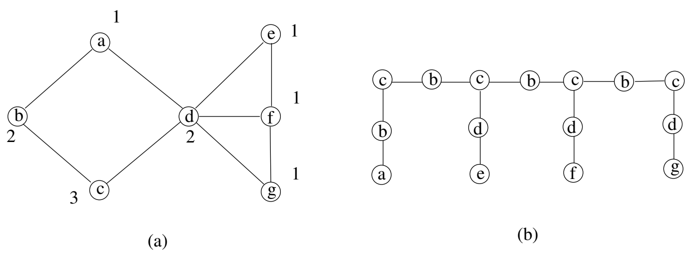


Figure 4. Illustration of Lemma 7. (**a**) indicates the graph and (**b**) its spanning hierarchy.

The following theorem formulates necessary and sufficient conditions for the existence of degree-constrained spanning hierarchies.

Theorem 1. *Let the set of conditions be as follows:*
A: V_1 *is empty*
B: there is no separator in V_1
C: $|V_1| \leq 2$ *(there are at most two nodes with degree bound* 1*)*

D: $|V_1| > 2$ and the nodes in V_1 are neighbor nodes of a node v s.t. $v \in V_m, m \geq |V_1|$
E: $|V \setminus V_1| \geq 2$ (there are at least two nodes with degree bound greater than 1) and $|V \setminus (V_2 \cup V_1)| \geq 1$ (there is at least one node with a degree bound greater than 2).

A hierarchy spanning all the nodes of a connected graph and respecting non-uniform capacity degree constraints can be found, if $A \vee (B \wedge (C \vee D \vee E))$

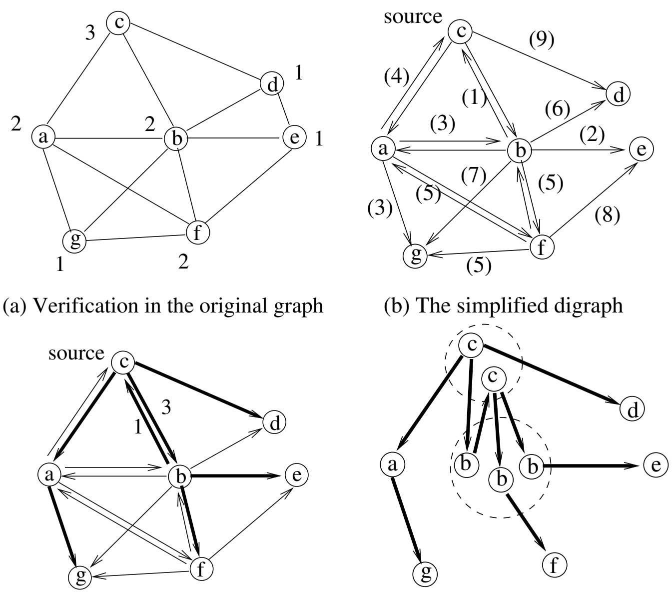
Proof. Following Lemma 1, condition A is sufficient.

If A is not true, the second part of the expression, i.e., $X = B \wedge (C \vee D \vee E)$ gives a sufficient condition for the existence of a solution as it can be proven easily. In any case, V_1 cannot contain a separator (condition B is necessary, as a consequence of Lemma 2), and one of the cases C, D or E must be true. Following Lemmas 4, 6 and 7 these conditions together with B are sufficient. If V_1 is not empty ($\neg A$) and there is no separator in it (B), the expression $(C \vee D \vee E)$ gives also a necessary condition, that is one of them should be true for the existence. We prove that if none of them are true, there is no spanning hierarchy respecting the constraints. Let us suppose that none of them are true: there are more than two nodes in V_1 ($\neg C$), and they are not neighbors of a node with a sufficiently large degree bound ($\neg D$), and there is no node with a degree bound greater than 2 or another with degree bound 2 ($\neg E$). We start our proof by $\neg E$. If there is a single node with a degree bound greater than 2, it must be the neighbor node of the nodes in V_1, but following $\neg D$, its bound is not sufficiently large. If there is no node with a degree bound greater than 2, there are only nodes in V_1 and in V_2. Following $\neg C$, there are more than two nodes in V_1. Therefore, the three conditions cannot be false together (and another fourth condition cannot guarantee the spanning hierarchy) when $\neg A \wedge B$. □

5. Algorithm Design

5.1. Exact Solution

We propose an ILP-based computation in several steps. It is an improved version of the procedure in [17] using a flow model in a directed graph. Since an arc can be used multiple times by a directed DCMSH, the flow computation must be adapted to this. We will show that the result of the ILP is not the optimal hierarchy but its image. Each arc in the image is associated with an integer value indicating how many times the arc is used. The computed image permits the reconstruction of the DCMSH. The procedure is illustrated by Figure 5 and detailed as follows.



(a) Verification in the original graph (b) The simplified digraph

(c) The directed image of the solution (d) The directed DCMSH

Figure 5. The steps of the construction of a DCMSH.

5.1.1. Verification for the Existence

Theorem 1 gives the necessary and sufficient conditions for the existence of any solution. These conditions must be verified.

(1) At first, it is checked that there is no node separated in V_1 itself. If there is at least one node in V_1 s.t. all its neighbors are also in V_1, it is isolated.
(2) V_1 cannot correspond to any separator in G. After removing V_1 from G, the number of connected components is calculated. If it is greater than 1, there is a separator and the solution does not exist.
(3) If there is a single node $v \in V_m, m > 1$ and $m < |V_1|$, there is no solution.
(4) If $|V_1| > 2$ and the nodes in $V \setminus V_1$ are in V_2, there is no solution.

Figure 5a shows an undirected graph and the degree constraints on the nodes. The graph does not contain any separators.

5.1.2. Creation of a Digraph

If the solution exists, a directed graph G'_A is created for the computation of the image of the optimum.

(1) At first, each edge of G is replaced by two arcs, one in each direction and having the same cost as the edge.
(2) A node s is selected as the source of flows. (in the computation a set of flows will be used, one flow to each destination from s)
(3) Potential simplifications of the digraph are possible. If $s \in V_1$, it can have only one successor and its incoming arcs can be deleted. The other nodes in V_1 can have only incoming arcs. Their outgoing arcs can be deleted.

The simplified digraph of the example is illustrated in Figure 5b. The costs of the arcs are also indicated.

5.1.3. Computation of the Image of the Solution

In the next step, a valued, directed image of the DCMSH is computed in the digraph. Integer values are associated with the arcs indicating the number of transited flows. The set of arcs with non-zero flows determines the image. Note that the number of flows is not equal to the number of occurrences of the arc in the directed DCMSH.

ILP variables:

$x(m,n)$: Integer variable. It is equal to the number of occurrences of arc (m,n) in the resulted hierarchy.

$F(m,n)$: Commodity flow variable. It denotes the quantity of flow transiting on arc (m,n).

Objective:

The cost of the image must be minimized.

$$Minimize \sum_{m \in V} \sum_{n \in \Gamma_m^+} x(m,n) \cdot C(m,n). \tag{1}$$

The values associated with the arcs indicate the number of incoming/outgoing arcs in the DCMSH. In an optimal hierarchy, there is a strictly minimal number of occurrences of any node. In our directed model, the number of occurrences of a non-source node is equal to the number of its predecessors. If there are multiple occurrences of the source node, the first one has no predecessor. Respecting the constraints, the number of occurrences must be sufficient to connect the successor nodes by the outgoing arcs. The degree constraints can be formulated as follows. At the arbitrarily selected source s, there are returns if and only if the degree $D(s)$ is not sufficient to be the start point of the outgoing arcs:

$$\sum_{n \in \Gamma_s^+} x(s,n) \leq D(s) + (D(s)-1) \cdot \sum_{n \in \Gamma_s^-} x(n,s). \tag{2}$$

Especially, if $D(s) = 1$, Γ_s^- is empty and there is only one outgoing arc.

For an arbitrary non-source node m, each occurrence of m can be the start point of at most $D(m) - 1$ outgoing messages:

$$\sum_{n \in \Gamma_m^+} x(m,n) \leq (D(m) - 1) \cdot \sum_{n \in \Gamma_m^-} x(n,m) \qquad \forall m \in V \setminus \{s\}. \tag{3}$$

Trivially, for nodes in V_1, no outgoing arc is used.

The following lower bounds are not necessary but practical.

At the source:

$$\sum_{n \in \Gamma_s^+} x(s,n) > D(s) + (D(s) - 1) \cdot (\sum_{n \in \Gamma_s^-} x(n,s) - 1). \tag{4}$$

In a non-source node m with degree bound $D(m) > 1$:

$$\sum_{n \in \Gamma_m^+} x(m,n) > (D(m) - 1) \cdot (\sum_{n \in \Gamma_m^-} x(n,m) - 1) \qquad \forall m \in V \setminus \{s\}, D(m) \neq 1. \tag{5}$$

Every node except the source has a predecessor in the hierarchy. This condition guarantees the full coverage as follows.

$$\sum_{n \in \Gamma_m^-} x(n,m) \geq 1 \qquad \forall m \in V \setminus \{s\}. \tag{6}$$

Full coverage does not mean full connectivity. Flows can be used to guarantee the connectivity. The arbitrary source is the start point of one flow per destination. (the destinations are the other $|V| - 1$ nodes). We have

$$\sum_{n \in \Gamma_s^+} F(s,n) = |V| - 1. \tag{7}$$

Each non-source node is the destination of one and only one flow. We require

$$\sum_{n \in \Gamma_m^+} F(m,n) = \sum_{n \in \Gamma_m^-} F(n,m) - 1 \qquad \forall m \in V \setminus \{s\}. \tag{8}$$

The conditions of flow transit by occurrences of arcs are as follows.

$$F(m,n) = 0 \qquad \forall (m,n) \in A \text{ if } x(m,n) = 0, \tag{9}$$

$$F(m,n) \geq x(m,n) \qquad \forall (m,n) \in A \text{ if } x(m,n) > 0. \tag{10}$$

If an arc does not belong to the solution, it does not carry any flow. Moreover, each arc should carry a non-zero flow if it belongs to the DCMSH, and the value of this flow must not be less than the number of occurrences of the arc.

The number of occurrences of an edge in a DCMSH (and the number of occurrences of an arc in the directed model) is limited:

$$x(n,m) \leq EB \qquad \forall m,n \in V. \tag{11}$$

The upper bound EB can be calculated in advance.

Remember, solving our linear program does not give a DCMSH but its image. Figure 5c illustrates the valued image of the solution in our example. A last phase is needed to construct an optimal "non-elementary tree" from its image.

5.1.4. Reconstruction of the Optimal Hierarchy

The variable $x(n,m)$ indicates how many times the arc (n,m) is used in the directed solution. From the image, a directed DCMSH can be built. A directed, labeled tree can be first reconstructed, and then an undirected version can be obtained. Several directed trees can correspond to the image if the number of occurrences of some arcs is different from one. Let $i(n) = \sum_{m \in \Gamma^-(n)} x(n,m)$ and $o(n) = \sum_{m \in \Gamma^+(n)} x(n,m)$ be the numbers of incoming and outgoing arcs, respectively. Each node occurrence has one and only one predecessor (except the source). A node n must be duplicated $i(n)$ times in the labeled tree (i.e., in the DCMSH). For each occurrence of n, there is one and only one incoming arc. For the source, one occurrence has no predecessor and the number of occurrences is $i(n) + 1$.

The successors can be arbitrarily associated with the occurrences of n, but to respect the degree constraints, each occurrence of n can have at most $D(n) - 1$ successors. Property 4.3 in [17] indicates that for a node n, an optimal hierarchy exists in which all occurrences of n except one have the maximal degree $D(n)$. Following this property, the successors can be distributed between the occurrences of n such that $D(n) - 1$ successors are connected to each first $o(n) - 1$ occurrences of n. A depth-first search type recursive algorithm can be used for the construction of a directed, labeled tree T_d. Figure 5d shows the directed DCMSH built in the given graph. The labels of the nodes in the tree are the labels of the nodes in the image.

Omitting the direction of arcs from T_d, an undirected, labeled tree T_u is obtained.

Theorem 2. *The labeled tree T_u corresponds to a DCMSH.*

Proof. Since the directed tree T_d (and consequently T_u) refers to the nodes in G, and the arcs (the edges) in T_d (in T_u) correspond to arcs (to edges) in G_d (in G), neighbor nodes in T_d (in T_u) are neighbors in G_d (in G). As a result $H_d = (T_d, h, G_d)$ and $H = (T_u, h, G)$ are hierarchies.

The computed image covers the whole graph G_d. Each node in V is associated with at least one node of T_d (and with at least one node of T_u). The hierarchy H spans the node set V. By the construction, the nodes in the tree T_u respect the degree constraints (a node occurrence of the node $n \in V$ has at most $D(n)$ neighbors). So, H is a degree-constrained hierarchy spanning V. It is with minimum cost. Its cost is the sum of the cost of edge occurrences. Because an edge $\{m,n\}$ is present $x(m,n) + x(n,m)$ times in H:

$$c(DCMSH) = \sum_{\{m,n\} \in E} c(m,n) \cdot (x(m,n) + x(n,m)).$$

This sum is minimized by the ILP. □

5.2. A Heuristic

An approximation of the DCMSH under homogeneous capacity degree bounds has been presented in [6] and improved in [18]. The proposed approximation scheme can not directly be used for our case with in-homogeneous degree bounds. In the present paper, possible approximations under non-uniform degree bounds are analyzed, and a heuristic is proposed. First of all, a lower bound of the cost of any optimal solution is presented.

5.2.1. A Lower Bound

Consider the following spanning tree.

Definition 5 (Minimum spanning tree with specified leaves—MSTSL)**.** *Let $V_1 \subset V$ be the subset of desired leaves. The MSTSL is a minimum cost spanning tree covering the node set such that the nodes in V_1 are leaves in the tree.*

The degrees of the nodes in $V \setminus V_1$ are not limited in the tree. An MSTSL in G can be computed in polynomial time as follows:

1. Delete the nodes in V_1.
2. Compute an MST in the remaining graph.
3. Reconnect the nodes in V_1 to the MST using the best cost adjacent edges.

A more complex study can be found in [19], which gives sufficient conditions for a graph to have a spanning tree with a specified set of leaves.

Theorem 3. *Suppose that spanning hierarchies corresponding to the given degree constraints exist in a graph G. Then, an MSTSL exists and its cost is a lower bound for the cost of any DCMSH.*

Proof. Satisfying the conditions in Theorem 1 guarantees that there is no separator in V_1 and the MSTSL exists. In a DCMSH H, each node in V_1 is present only once (nodes in V_1 must be leaves in H). By deleting the leaf nodes in V_1 and the set E_1 of their adjacent edges from H, a hierarchy H' is obtained. The image I' of H' in G may contain cycles. Since a node of $v \in V \setminus V_1$ can be associated with several node occurrences in H and in H', it can have more than $D(v)$ neighbors in the image. I' does not necessarily respect the degree constraint. From each possible cycle in I', an edge can be deleted and a tree T' in G covering the node set $V \setminus V_1$ is obtained (it is possible that T' does not respect the degree constraints). By adding the deleted nodes in V_1 using the edges in E_1 to T' a spanning tree T_1 is created. It respects the constraints for the specified leaf nodes in V_1. Since T_1 contains the edges that are present in H only once and some edges are eventually deleted from the image I': $c(T_1) \leq c(DCMSH)$. Since an MSTSL T_L exists and T_1 is also a spanning tree respecting the constraints given by V_1, $c(T_L) \leq c(T_1)$. Finally, $c(T_L) \leq c(T_1) \leq c(DCMSH)$. □

The main features of the proposed heuristic are as follows.

1. An MSTSL is created.
2. The MSTSL is decomposed into a set of edge-disjoint stars.
3. A degree-constrained spanning hierarchy with low cost is computed in each star.
4. Finally, these spanning hierarchies are connected to form a unique hierarchy spanning the MSTSL. (cf. Figure 6).

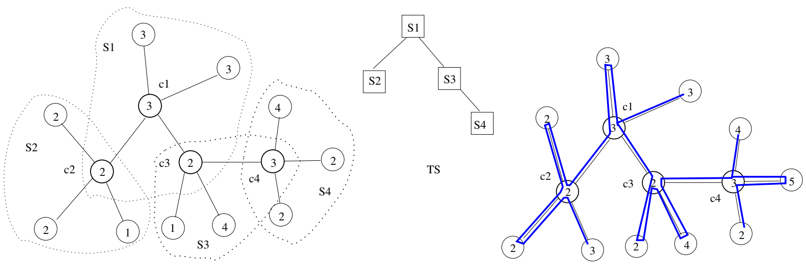


Figure 6. The decomposition of an MSTSL, the tree of stars and its spanning hierarchy.

5.2.2. Decomposition of the MSTSL into a Set of Edge-Disjoint Stars

Let us suppose there are more than three nodes in the tree; otherwise, the tree is a path and decomposition is useless. The decomposition starts with the unique neighbor node of an arbitrary leaf of the MSTSL (e.g., node c_1 in Figure 6). This node is considered the first star's central node, and the nodes related to this central node are leaves in the star ($S1$ in the figure). If a leaf in the star is not a leaf in the MSTSL, it is considered the central node of a following star. In our example, the nodes c_2 and c_3 are such nodes in $S1$. The decomposition continues in the newly detected stars until there are no more leaves eligible for new centers.

The decomposition can be seen as a rooted tree TS of stars in which each star (except the first) has a "parent". Each star is connected to its "parent" via its central node (which is a leaf in the "parent" star).

5.2.3. Computation of a Degree-Constrained Hierarchy Spanning a Star

A polynomial-time algorithm computes a spanning hierarchy H_c in a star S with a central node c satisfying the degree constraints. The outline is as follows:

1. If $D(c) \geq d(c)$, the star itself satisfies the degree constraint; there is nothing to do.

2. If $D(c) < d(c)$ and there are more than two leaves with degree bound one in S and there is no node with a degree bound greater than two, there is no solution (cf. Theorem 1).
3. Otherwise, we propose to cover S by a set $SC = \{Sc_i, i = 1 \dots n_S\}$ of connected, degree constrained small stars.

In the last case, multiple occurrences of c are needed, and some edges are duplicated. Trivially, edges going to nodes in V_1 are not duplicated. For the duplication, the best cost edges are selected in ascending order. Each duplication of an edge $\{c, w\}$ links two occurrences of c. At most $D(w)$ adjacent edges of w are possible. Each small star can cover at most $D(c) - 1$ leaves different from w. Some adjacent edges of the occurrences of c must be reserved for the connection to the parent star. An example of these small stars in a star is illustrated in Figure 7. The dotted lines indicate the connections to the parent star.

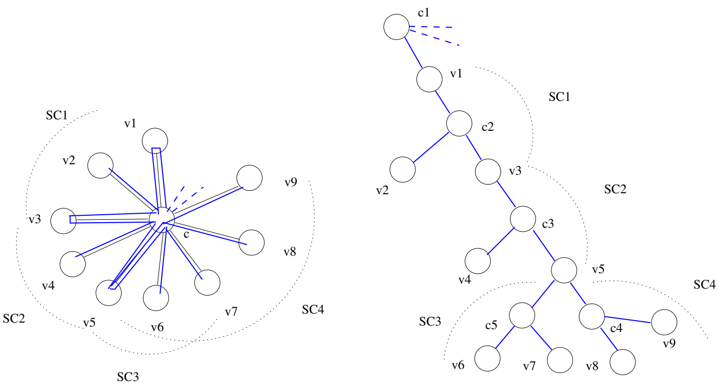


Figure 7. Degree-constrained spanning hierarchy H_c of a star.

5.2.4. The Connection of the Hierarchies

Let us suppose, that the spanning hierarchy H_c of S with the central node c must be connected to the hierarchy H_p spanning the parent star S' with the central node p. The edge $\{p, c\}$ is used $dp(c) \le D(c)$ times in S'. It is the number of adjacent edges of c in H_p. Moreover, c has $d_c(c) > 0$ adjacent edges in H_c. Finally, $d_p(c) + d_c(c)$ neighbor edges must be connected to c. If $d_p(c) + d_c(c) \le D(c)$, a unique occurrence of c is sufficient for the connections. If $d_p(c) < D(c)$, the first occurrence of c ensures the $d_p(c)$ connections to the parent node p and $D(c) - d_p(c) - 1$ edges in H_c. The remaining edges in H_c must be connected to the next occurrences of c in the defined order. If $d_p(c) = D(c)$, the first occurrence of c can have only $D(c) - 1$ times occurrences of $\{p, c\}$ as adjacent edges and the last occurrence of $\{p, c\}$ is connected to the second occurrence of c, etc. In the example of Figure 7, $D(c) = 3$ and $d_p(c) = 2$. The first occurrence of c is connected twice to S_p (these connections are indicated with dashed lines).

As stated before, the heuristic does not work in cases where the conditions in Theorem 1 are not met in a star. This means that the proposed heuristic does not work in these cases, even if an eventual degree-constrained spanning hierarchy exists in the MSTSL.

6. Approximations

If there are only a small number of nodes in V_1, guaranteed ratios can be found.

6.1. *Case of* $V_1 = \emptyset$

Theorem 4. *Let us suppose there is no node with a degree bound* 1*. The cost of a DCMSH can be approximated within a ratio* $\frac{D}{D-1}$*, where* $D = \min_{v \in V} D(v)$.

Proof. The proposed heuristic gives an approximation. Remember, the MSTSL T is decomposed into a set $S = \{S_i, i = 1, \dots k\}$ of edge-disjoint stars. Let us recall that in each star S_i, a spanning hierarchy H_{Si} satisfying the degree constraint of the (possibly multiplied) central node c_i can be built. Since $V_1 = \emptyset$, possible duplicates concern the best cost edges in H_{Si}. In the worst case, the leaves are in V_2 and only one return is possible from each leaf. The cost of the hierarchy H_{Si} spanning the star S_i is bounded by $\frac{D(c_i)}{D(c_i)-1}c(S_i)$ where $c(S_i)$ is the cost of the star (cf. [6]). The hierarchies $H_{Si}, i = 1, \ldots, k$ spanning the k different stars can be connected, and a hierarchy H spanning the node set V and satisfying the degree constraints is obtained. With $D = \min_{i=1,\ldots,k} D(c_i)$, since $\frac{D(c_i)}{D(c_i)-1} \leq \frac{D}{D-1}, i = 1, \ldots, k$ the cost of the spanning hierarchy is bounded:

$$c(DCMSH) = \sum_{i=1}^{k} c(H_{Si}) \leq \sum_{i=1}^{k} \frac{D(c_i)}{D(c_i)-1} c(S_i) \leq \frac{D}{D-1} c(T).$$

□

In the worst case, the degree bound of all nodes is two, and the ratio of costs is also two. If there are nodes with large degree bounds, the heuristic computes hierarchies close to the optimum.

6.2. Case of $0 < |V_1| \leq 2$

In this case, we suppose that V_1 is not empty but it contains at most two nodes that are not separators. We propose to analyze the case of $|V_1| = 2$, but the result is trivially true if there is only one node in V_1.

Theorem 5. *In the case of at most two specified leaves in* V_1*, the cost of a DCMSH can be approximated within a factor* 2*.*

Proof. Let a and b be the two nodes in V_1 and let T be an MSTSL having nodes a and b as leaves. Following T, a walk can be computed starting from a and ending at b. This walk W contains the edges of T at most twice and corresponds to a spanning hierarchy satisfying the degree constraints (the internal nodes in W have only degree 2). Finally, $c(W) \leq 2 \cdot c(T) \leq 2 \cdot c(DCMSH)$. □

If there are nodes with a degree bound greater than two, the heuristic proposed in Section 5.2 can return hierarchies with lower cost.

6.3. Case of $|V_1| > 2$

Unfortunately, in arbitrary graphs with $|V_1| > 2$, a bounded approximation ratio related to an MSTSL can not be guaranteed.

Remark 1. *Even if a DCMSH exists, it can not always be approximated by a constant factor compared to an MSTSL when* $|V_1| > 2$*.*

To illustrate this remark, let us consider a simple star as follows. Let a be a leaf with a degree bound 3 and b the central node in the star with a degree bound 2. Let k nodes in V_1 be connected to the node b. So there is a feasible solution. Consider the cost of the edges leading to leaves in V_1 being w negligible compared to the cost of the edge $\{a, b\}$. In this graph (cf. Figure 8), the MSTSL corresponds to the graph itself, and its cost is equal to $1 + k \cdot w$. In the DCMSH (indicated in the figure), the possible branching nodes are occurrences of a. The cost of a path from a to an arbitrary leaf is $1 + w$. Since there are k leaves: $c(DCMSH) > k(1 + w)$. The ratio between the costs is as follows:

$$\frac{c(DCMSH)}{c(MSTSL)} > \frac{k(1+w)}{1+k \cdot w}.$$

When w tends to zero, the lower bound tends to k. When k tends to infinity, the lower bound cannot be bounded by a constant.

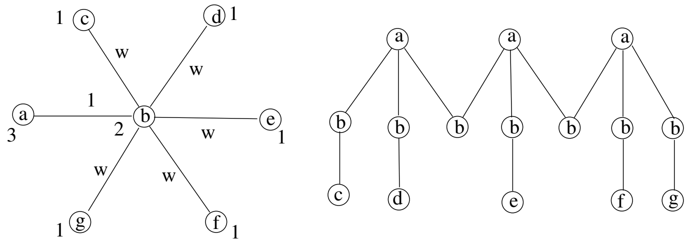


Figure 8. A particular star and its DCMSH.

Note that MSTSLs are not the only possible lower bounds of our problem. Whether or not an approximation related to another sub-graph (tree or spanning hierarchy computable in polynomial time) can be found is an open question.

7. Results, Evaluation of the Solutions

We compare four solutions (the MSTSL, the DCMSH, the DCMST, and the result of the heuristic) if they exist.

7.1. Materials and Methods

For the analysis, random graphs based on the model of Albert-Barabási [20] were generated. The algorithms are implemented in C++ using the Leda library [21]. The DCMSHs and DCMSTs were computed using the CPLEX Ilog server [22]. The heuristic and the computation of the MSTSL are in polynomial time, but the exact solutions need expensive computations. Fortunately, in the tested random graphs the computation of the exact solutions required from a few seconds to a few hours and was feasible.

Parameter Settings

In the experiment, some parameters were variable: the number of nodes in the random graphs, the degree bounds on the nodes, and the edge costs. Medium-size topologies were generated containing 80–170 nodes. Notice that the number of edges in the graphs was not deterministic, since the edges were generated randomly. In each experiment presented hereafter, we indicate the average number of edges. The degree bounds were set randomly from an interval $[D_{min}, D_{max}]$ using a uniform) distribution. The edge costs were also uniformly generated from the interval $[1, C_{max}]$.

7.2. Discussion

7.2.1. Effect of the Degree Bounds

In the first experiment, we show the effect of the maximal degree bound D_{max} varied from 3 to 12 when $D_{min} = 1$. The edge costs were randomly generated with $C_{max} = 5$. Table 2 resumes the graph properties, and also the number of cases in which trees and hierarchies corresponding to the constraints exist. In each run, 100 random graphs with 100 nodes were created.

Table 2. Results by varying D_{max}.

D_{max}	$\lvert V_1 \rvert$	$\lvert A_{G'_A} \rvert$	*nb(H)*	*nb(T)*	*c(MSTSL)*	*c(Heur)*	*c(DCMSH)*	*c(DCMST)*
3	34.06	613.06	86	50	150.116	201.151	178.43	181.8
4	25.14	689.67	96	96	142.188	180.656	153.802	156.979
5	19.29	751.69	100	100	136.61	166.21	143.63	144.66
6	17.1	768.76	100	100	134.19	157.54	139.32	139.9
7	14.01	797.2	99	99	131.778	149.162	135.273	135.657
8	12.9	807.07	100	100	131.05	145.46	133.8	134.12
9	11.54	822.09	100	100	130.39	144.9	132.9	133.09
10	9.92	835.36	99	99	130.162	140.222	132.061	132.172
11	9.12	841.31	100	100	128.51	139.73	130.42	130.54
12	8.32	849.74	100	100	127.17	135.76	128.67	128.76

The average number $|E|$ of randomly generated edges was 463 in each run. Trivially, the average number $|V_1|$ decreases by increasing D_{max}. The column $|A_{G'_A}|$ indicates the average number of arcs in the simplified digraph. The next two columns indicate the number of graphs satisfying the conditions for the existence of the DCMSHs (the number of hierarchies) and the number of DCMSTs, respectively. With a large set V_1, there are fewer possible trees. The last columns contain the average costs of the four structures. It is clear that as D_{max} increases, the costs converge to the same value. Probably, the cost $c(MSTSL)$ converges towards the cost of the MST. The cost $c(DCMSH)$ of the optimal hierarchies is always better or equal to the cost $c(DCMST)$ (if these solutions exist). By increasing D_{max}, the probability of degree bounds one (and also of the other small degree bounds imposing real constraints on the degrees) decreases. Consequently, the difference between trees and hierarchies decreases and their costs tend to the cost of the MSTSL. Notice that the heuristic also provides good costs.

7.2.2. Effect of the Edge Costs

In this case, the number of nodes was fixed at 100 and D_{max} was equal to 3. The edge costs were randomly generated from the interval $[1, C_{max}]$, and the value of C_{max} varied as indicated in Table 3.

Table 3. Results by varying the edge costs.

C_{max}	$\|E_{G'_A}\|$	*dupl*	*nb(H)*	*nb(T)*	*c(MSTSL)*	*c(Heur)*	*c(DCMSH)*	*c(DCMST)*
1	100.725	1.242	91	57	99	144.176	101.143	99
2	102.349	1.542	83	57	104.675	141.265	111.458	109.088
3	105.556	2.044	90	63	116.5	154.589	132.644	132.746
4	107.536	2.369	84	53	132.417	174.774	155.643	158.585
5	108.837	2.5	92	55	147.804	194.859	176.652	181.709
6	109.209	2.813	91	61	165.934	220.824	197.593	206.541
7	108.965	3.012	85	63	180.365	240.624	215.153	228.381
8	110.022	3.198	91	61	199.56	264.835	240.495	259.279
9	109.683	3.598	82	54	219.646	290.707	263.268	283.13

Independently from the edge costs, the obtained typologies are relatively stable; $|V_1|$ and $|E_{SG}|$ show a small variation. DCMST does not exist in 50–60% of cases and DCMSH does not exist in 10% of cases. The first lines of the results need explanation. Let us analyze the first line. If the DCMST exists (in 57 cases of 100 runs) the cost is equal to 99, which is the cost of the 99 edges needed to cover the 100 nodes. The average cost of DCMSHs is greater, but it is the average of 91 cases that contain hierarchies returning several times to some nodes (cases not covered by trees). The average of the maximal number of edge duplication is indicated in the column $dupl$.

7.2.3. Varying the Size of the Graphs

A third series allows us to see the effect of the size of Albert-Barabási random graphs on the minimum spanning structures. The experiment is summarized in Table 4. The number of nodes varied from 80 to 170 nodes. D_{max} was equal to 3 and C_{max} to 5. It can be seen that the observed values increase almost linearly with the size of the graphs. For instance, let us compare the lines with 80 and 160 nodes. The average number of edges, the number of desired leaves in V_1, and the cost are almost double. The number of DCMSHs and DCMSTs are mainly the same. Note that trees are 5–10% more expensive than hierarchies.

Table 4. Results by varying the graph size.

M	$\vert E\vert$	$\vert V_1\vert$	$\vert E_{G'_A}\vert$	*nb(H)*	*nb(T)*	*c(MSTSL)*	*c(Heur)*	*c(DCMSH)*	*c(DCMST)*
80	364.5	20.14	543.69	97	97	113.381	146.34	122.124	124.351
90	414.71	22.16	625.29	98	98	125.673	159.673	135.408	137.969
100	463.67	24.98	694.83	97	96	139.897	177.093	150.351	152.802
110	513.32	27.62	768.61	98	98	153.735	195.276	165.847	168.888
120	562.41	29.21	849.71	98	98	167.571	211.327	180.347	183.306
130	612.35	33.19	912.57	97	97	182.34	231.216	196.505	200.412
140	661.74	35.55	990.49	94	94	211.351	265.606	227.915	232.457
160	760.62	40.5	1139.45	98	97	223.867	283.52	241.643	246.649
170	810.81	42.95	1212.8	96	96	236.906	304.094	257.229	262.969

8. Conclusions and Perspectives

In our study, new results on degree-constrained minimum spanning hierarchies in graphs with non-uniform capacity constraints are presented. DCMSHs can be built in several cases, where DCMSTs do not exist. An ILP-based exact solution and a heuristic are proposed for the NP-hard DCMSH problem. Certain approximation ratios are also discussed. As the solution does not always exist, the necessary and sufficient conditions for the existence are formulated.

Our study shows that the set of specified leaves with degree bound 1 is crucial for the properties of the solutions. If this set is large, the solution may not exist and the approximation within a constant is not trivial (we consider, it to be an open question). The DCMSH exists in several cases where the DCMST does not exist. The average cost of DCMSHs is lower than the average cost of DCMSTs. The weaker the constraints, the more the solution converges to the MST (or to the MSTSL). Note that the heuristic provides good results relatively close to the optimum.

In future work, the approximation of spanning hierarchy problems with different degree constraints in various graphs should be analyzed. We conjecture that in some cases, depending on the positions of leaves in V_1, approximations can be found even if $|V_1| > 3$. Another important perspective is the analysis of partial spanning problems like degree-constrained Steiner problems. We believe that an important part of the results presented in this paper can be applied to partial spanning problems. Applications of the degree-constrained spanning hierarchies can be found, for instance in optical broadcast/multicast routing. In these applications, additional constraints (e.g., the uniqueness of the used wavelength in the fibers, limited length of paths, etc) exist, and various constraints must be fully satisfied. Analyzing these issues promises further interesting challenges.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Deo, N.; Hakimi, S. The shortest generalized Hamiltonian tree. In Proceedings of the Sixth Annual Allerton Conference, Monticello, IL, USA, 2–4 October 1968; pp. 879–888.
2. Boldon, B.; Deo, N.; Kumar, N. Minimum-Weight Degree-Constrained Spanning Tree Problem: Heuristics and Implementation on an SIMD Parallel Machine. *Parallel Comput.* **1996**, *22*, 369–382. [CrossRef]
3. Ravi, R.; Marathe, M.V.; Ravi, S.S.; Rosenkrantz, D.J.; Iii, H.B.H. Approximation algorithms for degree-constrained minimum-cost network-design problems. *Algorithmica* **2001**, *31*, 58–78. [CrossRef]
4. Mukherjee, B. *Optical WDM Networks (Optical Networks)*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 26–27.
5. Ali, M.; Deogun, J. Cost-effective implementation of multicasting in wavelength-routed networks. *IEEE J. Light. Technol. Spec. Issue Opt. Netw.* **2000**, *18*, 1628–1638. [CrossRef]
6. Molnár, M.; Durand, S.; Merabet, M. Approximation of the Degree-Constrained Minimum Spanning Hierarchies. In Proceedings of the SIROCCO, Takayama, Japan, 23–25 July 2014; pp. 96–107.

7. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*; W. H. Freeman & Co.: New York, NY, USA, 1979.
8. Cieslik, D. The vertex degrees of minimum spanning trees. *Eur. J. Oper. Res.* **2000**, *125*, 278–282. [CrossRef]
9. Könemann, J.; Ravi, R. A Matter of Degree: Improved Approximation Algorithms for Degree-Bounded Minimum Spanning Trees. *SIAM J. Comput.* **2002**, *31*, 1783–1793. [CrossRef]
10. Könemann, J.; Ravi, R. Primal-Dual Meets Local Search: Approximating MSTs With Nonuniform Degree Bounds. *SIAM J. Comput.* **2005**, *34*, 763–773. [CrossRef]
11. Ravi, R. Matching Based Augmentations for Approximating Connectivity Problems. In Proceedings of the 7th Latin American Symposium on Theoretical Informatics (LATIN'06), Valdivia, Chile, 20–24 March 2006; pp. 13–24.
12. Ravi, R.; Singh, M. Delegate and Conquer: An LP-Based Approximation Algorithm for Minimum Degree MSTs. In Proceedings of the ICALP, Venice, Italy, 10–14 July 2006; pp. 169–180.
13. Singh, M.; Lau, L.C. Approximating minimum bounded degree spanning trees to within one of optimal. In Proceedings of the STOC '07: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, San Diego, CA, USA, 11–13 June 2007; pp. 661–670.
14. Fürer, M.; Raghavachari, B. Approximating the minimum-degree Steiner tree to within one of optimal. *J. Algorithms* **1994**, *17*, 409–423. [CrossRef]
15. Russo, L.M.S.; Teixeira, A.S.; Francisco, A.P. Linking and Cutting Spanning Trees. *Algorithms* **2018**, *11*, 53. [CrossRef]
16. Gao, X.; Jia, L. Degree-constrained minimum spanning tree problem with uncertain edge weights. *Appl. Soft Comput.* **2017**, *56*, 580–588. [CrossRef]
17. Merabet, M.; Molnár, M.; Durand, S. ILP formulation of the degree-constrained minimum spanning hierarchy problem. *J. Comb. Optim.* **2018**, *36*, 789–811. [CrossRef]
18. Singh, M.; Zenklusen, R. k-Trails: Recognition, Complexity, and Approximations. In *Integer Programming and Combinatorial Optimization*; Louveaux, Q., Skutella, M., Eds.; Springer: Liège, Belgium, 1–3 June, 2016; pp. 114–125.
19. Egawa, Y.; Matsuda, H.; Yamashita, T.; Yoshimoto, K. On a Spanning Tree with Specified Leaves. *Graphs Comb.* **2008**, *24*, 13–18. [CrossRef]
20. Albert, R.; Barabási, A.L. Statistical mechanics of complex networks. *Rev. Mod. Phys.* **2002**, *74*, 47–97. [CrossRef]
21. Mehlhorn, K.; Näher, S. *LEDA: A Platform for Combinatorial and Geometric Computing*; Cambridge University Press: Cambridge, MA, USA, 1999.
22. Cplex, I.I. V12. 1: User's Manual for CPLEX. *Int. Bus. Mach. Corp.* **2009**, *46*, 157.