

Article

Degree-Constrained Steiner Problem in Graphs with Capacity Constraints

Miklos Molnar 

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

Abstract: The degree-constrained Steiner problem in graphs is well known in the literature. In an undirected graph, positive integer degree bounds are associated with nodes and positive costs with the edges. The goal is to find the minimum cost tree spanning a given node set while respecting the degree bounds. As it is known, finding a tree satisfying the constraints is not always possible. The problem differs when the nodes can participate multiple times in the coverage and the constraints represent a limited degree (a capacity) for each occurrence of the nodes. The optimum corresponds to a graph-related structure, i.e., to a hierarchy. Finding the solution to this particular Steiner problem is NP-hard. We investigate the conditions of its existence and its exact computation. The gain of the hierarchies is demonstrated by solving ILPs to compute hierarchies and trees. The advantages of the spanning hierarchies are conclusive: (1) spanning hierarchies can be found in some cases where spanning trees matching the degree constraints do not exist; (2) the cost of the hierarchy can be lower even if the Steiner tree satisfying the constraints exists.

Keywords: graph theory; degree-constrained Steiner tree; Steiner hierarchy; homomorphism; NP-hard problem; exact computation; approximation

MSC: 05C05; 05C07; 05C60; 90-10



Citation: Molnar, M. Degree-Constrained Steiner Problem in Graphs with Capacity Constraints. *Mathematics* **2024**, *12*, 3521. <https://doi.org/10.3390/math12223521>

Academic Editor: Andrea Scozzari

Received: 4 October 2024

Revised: 4 November 2024

Accepted: 7 November 2024

Published: 11 November 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In the Steiner problem in graphs, a given subset of nodes must be spanned at a minimum cost. Without any constraint, the optimum is a partial minimum spanning tree (PMST) or a Steiner tree. In some applications, additional constraints must be satisfied. Various constrained spanning problems and Steiner problems have been analyzed in graphs (cf. examples in [1–3]). We are interested in the degree-constrained Steiner 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 the node in any spanning structure (for example, in the spanning trees).

Often, *budget*-type degree constraints are assumed. For instance, when the bounds are uniform and equal to two, only paths can be used to span the nodes (the problem corresponds to the well-known traveling salesman problem, and the solution does not always exist). Having different bounds, it is not always possible to span the nodes using trees that meet the degree constraints. Degree-constrained spanning tree and Steiner tree problems are NP-hard. Moreover, they can not be approximated by a constant factor if the degrees correspond to budgets [4]. In the case of a limited budget of the nodes, the minimum cost solution is a tree: it contains a node at most once.

The motivation of the recent study is the definition and the analysis of the exact solution of the minimum cost degree-constrained partial spanning in the cases of limited node capacities. *Supposing capacity-like constraints on the node degrees, the optimum may be different from a tree* (we suppose that a node can participate multiple times in the coverage and the degree of each occurrence is limited by the given capacity). In these cases, a spanning hierarchy corresponds to the optimum. The next section provides formal definitions. The

main advantages of the hierarchies in the discussed Steiner problems are: (1) In some cases, Steiner hierarchies exist but Steiner trees that meet the constraints do not exist; (2) Even if Steiner trees respecting the degree constraints exist, in some cases, a hierarchy of least cost can be constructed.

Capacity-like constraints can be found in transparent optical networks where the duplication capacity of the optical switches is limited. If there is no node/switch capable of duplicating or splitting the incoming light, only light paths can be used for optical routing. The optical multicast route using the cross-connect capacity of the switches can visit a switch several times and can be different from a partial spanning tree [5]. The reader can learn more about the specifics of optical networks in [6].

To the best of our knowledge, our paper is the first to analyze the solution for the minimum Steiner problem under bounded degree capacities in the nodes.

In the paper, we propose a quick presentation of the well-known degree-constrained Steiner tree problems (cf. Section 2). The related definitions and the used notations can be found in Section 3. The necessary and sufficient conditions for a solution to exist are formulated in Section 4. An ILP-based exact computation (cf. Section 5) is also proposed. The tree-based and hierarchy-based solutions are compared in Section 7 followed by the conclusions.

2. Previous Work

The minimum Steiner tree computation is largely studied in the literature. Several hundred papers can be found on the approximations and methods for solving this NP-hard problem and its variants. The 11th DIMACS Implementation Challenge as well as the PACE 2018 ChallengeA provide important summaries of the evolution of the treatments (methods and new computational results) of the Steiner problem. A large survey is presented in [7]. A recent paper [8] analyzes integer programming and mixed-integer programming formulations for the optimization problems that are based on induced connectivity. It provides some conditions for LP-relaxations of both the maximum weight connected sub-graph and the Steiner tree problem to be tight. Local search algorithms are used for improving the computed Steiner trees by replacing sub-trees with advantageous structures in [9]. The proposed methods outperform the known local search algorithms. The same authors propose a method based on supervised learning to produce a quasi-optimal Steiner tree in [10]. Artificial Intelligence is a good candidate for finding efficient patterns in the construction of spanning structures. Another learning-based solution can be found in [11] to solve the clustered Steiner tree problem (which is a recently introduced NP-hard problem for efficient multicast routing in a decentralized clustering network architecture).

The degree-constrained spanning tree problem in graphs is one of the NP-hard problems cited in [12]. While the degree-constrained spanning problem has been intensively studied, a few papers talk about the degree-constrained Steiner problem. An extensive study can be found in [13]. In [14], the authors argue the particular interest of this problem with the limited capability of switches in high-speed networks to realize multicast routing. It is not always possible to span the nodes using trees while respecting the degree constraints (cf. [14]). The general case in which the bounds are different is NP-hard using a reduction to the traveling salesman problem (cf. [2]).

The degree-constrained minimum cost sub-graph problems are NP-hard. Polynomial-time algorithms computing good approximations are needed. The first approximation scheme is proposed in [15]. In [4], the problem is presented as a generic bi-criteria optimization problem (A, B, S) where the first objective A corresponds to the respect of the budget constraints, the second (B) is the minimization of the cost, and S describes the class of the solution. The investigated classes (as usual) are spanning trees, Steiner trees, and generalized Steiner trees. The authors prove the hardness of the proposed optimizations. They demonstrate that the partial spanning problem with non-uniform bounds cannot be approximated with $(2 - \epsilon, \rho)$ for any $\epsilon > 0$ and $\rho > 1$ [4]. Paper [16] shows a quasi-polynomial time approximation algorithm computing a Steiner tree in a graph with n

nodes. The tree is of maximum degree $O(B + \log n)$, and its cost is within a constant factor of a minimum Steiner tree whose maximum degree is bounded by B .

For fault-tolerant networking, ref. [17] presents a variant, the minimum degree bounded Steiner network problem, where the minimum-cost Steiner network that satisfies all the degree bounds is required. A Steiner network is a graph with at least $r(u, v)$ edge-disjoint paths between nodes in all pairs of nodes u, v ($r(u, v)$ is the connectivity requirement for the network). The paper proposes a polynomial time approximation algorithm that returns a Steiner forest or a Steiner network of cost at most twice the optimal cost solution when the degree violation at each node is bounded. Lau et al. also present algorithmic and hardness results on the survivable network design problem with degree constraints; among others, there is a constant factor approximation algorithm applicable for many degree-constrained network design problems, including the minimum bounded degree Steiner forest problem [18]. An improvement can be found in [19]. An approximation algorithm for some degree-bounded directed network design problems, such as the minimum cost connected sub-graph problem and upper bounds on in-degrees and out-degrees of the vertices, is proposed in [20]. In [21], two degree-constrained sub-graph problems (the maximum d -degree-bounded connected sub-graph and the minimum sub-graph of minimum degree $\geq d$) and their approximations are discussed.

The network design problems with degree constraints are also analyzed in [22]. After the minimum-cost degree-constrained two-node-connected sub-graph problem and the problem of the arborescence with at least k terminals with minimum maximum out-degree, the terminal Steiner tree problem (where a degree-bound one must be respected for a set of nodes) is analyzed. The problem description is in [23], and a 2ρ -approximation algorithm for the terminal Steiner problem can be found in [24].

A special case of the degree-constrained Steiner tree, the binary Steiner tree, is analyzed in [25]. An exact solution and an approximation ratio are presented (in these trees, which are important in computational biology, the maximal degree bound is 3 for all nodes). Instead of individual degree bounds, a new norm of the node degree vector is proposed in [26]. This norm tries to find a compromise between the maximum degree and the number of edges in the sub-graph and keeps degrees small. A direct application of the degree-constrained Steiner arborescence problem in a symbol graph to the symbolic regression can be found in [27] (the symbolic regression is a task for discovering a symbolic expression that fits a given data set from the space of mathematical expressions).

In the case of degree bounds that correspond to the *capacities* of the nodes, a new degree-constrained Steiner problem can be formulated. The objective is to cover a given node set with a minimum-cost, connected structure that satisfies the imposed capacity-like degree constraints. Assuming that a node can be used several times and the capacity of its occurrences is the same, good solutions can be obtained. The solutions are hierarchies [28] presented hereafter. They can be considered as “non-elementary trees” in G . Recently, a study on the minimum spanning hierarchies in graphs was published [29]. The minimum spanning hierarchies are particular cases of the minimum Steiner hierarchies, where the node set to be covered is the node set in the graph. The Steiner problem with capacity constraints can be solved even if trees satisfying the constraints do not exist. The model is useful especially for the optical multicast routing, where the light can cross a switch several times. Optimal (minimum-cost) solutions correspond to light hierarchies [5].

3. Notations and Definitions

3.1. Notations

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

Table 1. Notations used in the paper.

$G = (V, E)$: undirected graph given by node set V and edge set E
$\{v, w\} \in E$: edge between nodes v and w
$c(v, w)$: cost associated with the edge $\{v, w\}$
$d(v)$: degree of the node v
$D(v)$: degree bound of v (positive integer)
$M \subseteq V$: the set of nodes to cover
$V_m \subseteq V$: the subset of nodes with degree bound m
v^i	: i^{th} occurrence of $v \in V$ in a hierarchy
$G_A = (V, A)$: directed graph, equivalent to G , two arcs $(v, x) \in A$ and $(w, v) \in A$ correspond to the edge $\{v, w\} \in E$ of G
$T = (P, F)$: 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 $v \in V$
$P_m \subseteq P$: the subset of nodes with degree m in the tree
$AdjN(v)$: the set of adjacent nodes of the node v
$AdjA^+(v)$: the set of incoming arcs at the node v in G_A
$AdjA^-(v)$: the set of outgoing arcs at the node v in G_A
$\Gamma^-(v)$: the set of predecessor nodes at the node v in G_A
$\Gamma^+(v)$: the set of successor nodes at the node v in G_A

3.2. Problem Formulation

In this paper, we suppose a simple graph $G = (V, E)$ with positive costs. A positive integer value $D(v)$ is associated with any node $v \in V$ and considered as the degree bound of v .

Definition 1 (Budget-type degree bound). *The number of the neighboring nodes of v in any coverage can not exceed the degree bound $D(v)$.*

Definition 2 (Capacity-type degree bound). *The number of the neighboring nodes of each occurrence of v in any coverage can not exceed the degree bound $D(v)$.*

Definition 3 (Spanning structure). *It is a structure spanning the node set $M \subseteq V$ that contains at least a walk between between the nodes of an arbitrary node pair of M .*

A generalization of the tree concept can be based on the graph homomorphism. The homomorphic mapping of the nodes of a tree T to nodes of a graph G defines a non-elementary tree in G .

Definition 4 (Hierarchy). *We let $T = (P, F)$ be a tree. We let $h : P \rightarrow V$ be a homomorphic function that associates node $v \in V$ to each node $p \in P$. Application (T, h, G) defines a hierarchy in G .*

Definition 5 (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.*

The definitions in digraphs are also trivial.

Problem 1 (Budget-type degree-constrained Steiner tree problem). *The integer value $D(v)$ associated with any node $v \in V$ is a budget. Given a subset $M \subseteq V$ of nodes, we find a partial spanning tree in which the degree of any node v is at most $D(v)$ and the tree cost is minimum.*

Problem 2 (Capacity-type degree-constrained Steiner problem). *The integer value $D(v)$ associated with any node $v \in V$ represents a capacity. Given a subset $M \subseteq V$ of nodes, we find a*

spanning structure such that the degree of any occurrence of v is at most $D(v)$ in the structure and its cost is minimum.

Property 1. A spanning structure of $M \subseteq V$ in G can be given by a homomorphism-based triplet (S, h, G) such that all nodes in M are associated by h with at least one node of S and S is a connected graph.

Proof. All nodes in M must be covered, and so they must be associated with at least one node in S . In a spanning structure, every pair of nodes of M must be connected by at least a walk. If S is not connected, there are node pairs without connection between them. Therefore, S must be connected. \square

Property 2. The minimum cost solution of the capacity-type degree-constrained Steiner problem is a hierarchy.

Proof. Suppose that the solution exists (the conditions for the existence are presented in Section 4). Any solution is a spanning structure and it can be given by a homomorphism-based triplet (S, h, G) . Suppose that 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. Some edges can be removed from the cycles in S without removing nodes, thus preserving the node coverage in G and satisfying the constraints with lesser cost. Consequently, in the minimum cost solution, S is a tree, and the triplet defines a hierarchy. \square

A degree-constrained minimum Steiner hierarchy (DCMStH) is illustrated in Figure 1. The cost of any edge in G is unitary, $M = \{a, f, h\}$, and the capacity constraints are indicated. Trivially, a DCMStT does not exist, but a Steiner hierarchy satisfying the constraints exists. It contains the node e twice, and each occurrence of this node respects the degree constraint.

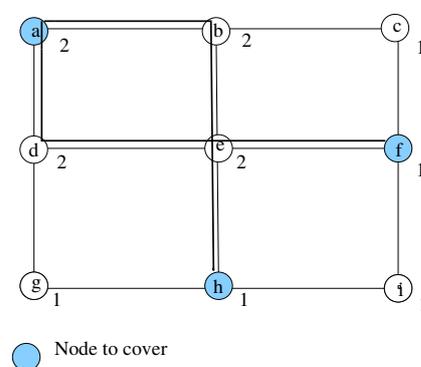


Figure 1. Illustration of a DCMStH.

The capacity-constrained minimum Steiner problem (the computation of the “Steiner hierarchy”) is NP-hard. A particular case of the problem is where the degree constraint $D(v) = d(v)$ for each node $v \in V$. This case is the classic NP-hard Steiner problem. Consequently, our more general problem is NP-hard.

The following section discusses the conditions for the existence of a feasible solution for the Steiner hierarchy problem.

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

To solve the capacity-constrained spanning problems, solutions do not always exist, neither for the minimum spanning hierarchy nor for the minimum Steiner hierarchy problem. The conditions for the existence of a minimum spanning hierarchy are presented in Section 4 in [29]. Here, we offer the conditions for the Steiner hierarchy problem. Despite the differences between the concerned node sets, an adaptation allows us to formulate

similar lemmas and proofs (cf. Lemmas 2 and 4–6). The conditions are not completely identical but close in the two problems.

Remember, $V_m \subseteq V$ indicates the subset of nodes with degree bound m . The nodes in V_1 must be leaves in any solution and cannot be internal nodes in any spanning structure. If the graph G contains nodes in V_1 that are not in M , these nodes cannot belong to any solution and can be deleted. We let $G' = (V', E')$, the graph obtained after the suppression of nodes in $V_1 \setminus M$. $V' = V \setminus (V_1 \setminus M)$, and E' does not contain the adjacent edges of the deleted nodes. Graph G' is not necessarily connected.

Lemma 1. *If G' does not contain a connected component containing all the nodes of the set M , there is no hierarchy spanning M and satisfying the degree constraints.*

Proof. In a spanning hierarchy, there is a walk between any pair of nodes. If there are members of M in different connected components in G' , then some walks are missing for a spanning structure. \square

We let $G'' = (V'', E'')$ be the connected component in G' that contains the node set M . Remember, the subset V_i'' is the subset of nodes with degree bound i in G'' , especially $V_1'' \subseteq M$.

Lemma 2. *If V_1'' contains a separator (removing a separator makes a graph disconnected) such that the nodes in M are separated into two or more sub-graphs, there is no hierarchy spanning M and satisfying the degree constraints.*

Proof. We let A and B be two non-empty subsets of nodes separated by separator $SP \subseteq V_1''$ and containing at least a node of M each. We let $m_A \in A \cap M$ and $m_B \in B \cap M$. Any walk from m_A to m_B must pass through SP , but nodes in SP cannot be internal nodes in any walk. Trivially, walks are not possible between m_A and m_B satisfying the degree constraints. \square

We let $MC = (M, EC)$ be a complete graph on M such that each edge in EC corresponds to an arbitrary walk without nodes in V_1'' as an internal node (the endpoints can be in V_1''). Following Lemma 2, this graph exists.

In the following, we suppose that V_1'' does not contain any separator cutting the set M . Remember that $V_1'' \subseteq M$.

Lemma 3. *If $|V_1''| < 3$, then hierarchies spanning M and satisfying the degree constraints exist.*

Proof. Based on the complete graph MC , simple solutions can be constructed as follows. In MC , Hamiltonian paths can be found. If $|V_1''| = 0$, we select an arbitrary Hamiltonian path. It corresponds to a Hamiltonian walk in G'' trivially satisfying the constraints. If $|V_1''| = 2$, we select a Hamiltonian path P_H with the two nodes with degree-bound values of one as endpoints. Path P_H corresponds to walk W_H in G'' . Since the internal nodes in P_H are not in V_1'' , walk W_H does not contain internal nodes in V_1'' . The degree constraints are satisfied and the node set M is covered. If $|V_1''| = 1$, one endpoint of the selected Hamiltonian path must be the node with degree bound one, and the other endpoint can be selected arbitrarily. As in the previous case, the corresponding walk is a solution. \square

Now, let us suppose that $|V_1''| > 2$ (there are at least three leaves in the spanning hierarchy).

Lemma 4. *Let us suppose that $|V'' \setminus V_1''| = 1$ and we let v_b the node in $V'' \setminus V_1''$. If the degree bound $D(v_b) < |V_1''|$, there is no hierarchy spanning M that respects the constraints.*

Proof. Since there is no separator in V_1'' , the nodes in V_1'' are directly connected to v_b . We suppose that 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 contradiction is trivial.

The nodes in V_1'' must be leaves in any spanning hierarchy, and there is no return from leaves. Consequently, only one occurrence of v_b exists in any hierarchy. This occurrence of v_b can have at most $D(v_b)$ neighbors, but $|V_1''| > D(v_b)$. \square

Lemma 5. *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'' is a star with the central node v_b such that $D(v_b) \geq |V_1''|$. The nodes in V_1'' are connected to it (there is no separator in V_1''). Since the leaves are in M and $D(v_b) \geq |V_1''|$, the star covers M and satisfies the degree constraints. \square

The cases where $|V'' \setminus V_1''| > 1$ with an arbitrary number of nodes in V_1'' are covered by the following lemma. We prove that one node with a degree bound of at least two and another with a degree bound greater than two are sufficient to build degree-constrained hierarchies spanning the set M for an arbitrary number of nodes in V'' . As previously stated, we assume that there is no separator in V_1'' .

Lemma 6. *If $|V'' \setminus V_1''| \geq 2$ (there are at least two nodes with degree bound greater than one) and $|V'' \setminus \{V_2'' \cup V_1''\}| \geq 1$ (there is at least one node with degree bound greater than two), a Steiner hierarchy respecting the degree constraints exists.*

Proof. We let $b \in V'' \setminus V_1''$ be a node with $D(b) \geq 2$, and another $c \in V'' \setminus V_1''$ with $D(c) > 2$. A degree-constrained hierarchy can be constructed as follows. Because there is no separator in V_1'' separating the nodes of M , each node in $M \setminus (\{b\} \cup \{c\})$ can be connected to c with a path and each path respects the degree constraints. The set of paths can be partitioned into groups 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 of at least two) such that the degree of each central node occurrence is at most $D(c)$. The obtained structure is a degree-constrained hierarchy spanning the node-set M . \square

A simple example shows the chain of the spiders in Figure 2.

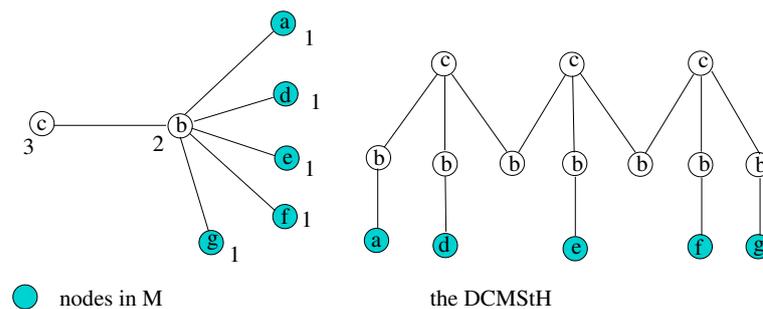


Figure 2. Illustration of Lemma 6.

The necessary and sufficient conditions for the existence of a degree-constrained minimum Steiner hierarchy covering a given node set M in G can be formulated as follows. Remember that the graph obtained after the suppression of nodes in $V_1 \setminus M$ is $G' = (V', E')$ and the connected component in G' that contains the node set M is $G'' = (V'', E'')$.

Theorem 1. *Let the set of conditions in G'' be as follows:*

A: G' contains a connected component with all nodes of M ;

B: There is no separator composed from nodes of V_1'' separating nodes in M into at least two sub-sets in G'' ;

C: $|V_1''| < 3$ (there are at most two nodes with degree bound 1 in G'');

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

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

Proof. A and B are necessary conditions (cf. Lemmas 1 and 2). Moreover, one of the cases C, D , or E must be true. Following Lemmas 3, 5, and 6, each of these conditions is sufficient. Expression $(C \vee D \vee E)$ also provides a necessary condition; that is, one of them must be true for the existence. We prove that if none of them are true, there is no Steiner hierarchy respecting the constraints. Let us suppose that none of them are true: there are more than two nodes in V_1'' ($\neg C$), they are not neighbors of a node with sufficiently large degree bound ($\neg D$), and there is no node with degree bound greater than two or another with a degree bound of two ($\neg E$). We start our proof with $\neg E$. If there is a single node with a degree bound greater than two, it must be the neighbor node of the nodes in V_1'' , but following $\neg D$, its degree bound is not sufficiently large. If there is no node with a degree bound greater than two, 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) even if $A \wedge B$ is true. \square

5. Computation of an Exact Solution

The proposed procedure computes the image of the optimum using an ILP in a directed simplified graph. The graph is the directed version of G'' . To simplify, we also note that $G'' = (V'', A'')$. In the ILP, for each arc of G'' , an integer value is computed indicating how many times the arc is used in the directed Steiner hierarchy. These values allow the reconstruction of the directed degree-constrained Steiner hierarchy and finally its undirected version. The main steps of the procedure are similar to the computation of the minimum spanning hierarchy in [29]. With the necessary adaptations to the Steiner problem, the steps are the following.

5.1. Creation of a Digraph

Nodes in $V_1 \setminus M$ must be deleted from G . A connected component G'' containing M must be determined. If such a component does not exist, there is no solution. Each edge $\{a, b\} \in E''$ is replaced by two arcs (a, b) and (b, a) to create A'' .

5.2. Verification for the Existence

Further conditions are indicated by Theorem 1 and must be verified.

If a sub-set of nodes of V_1'' is a separator regarding the nodes in M , there is no solution.

If there is a single node $v \in V_m'', m > 1$ and $m < |V_1''|$, no solution exists.

If $|V_1''| > 2$ and the nodes in $V'' \setminus V_1''$ are in V_2'' , there is no solution.

5.3. Computation of the Image of the Solution

In the next step, the valued, directed image of the DCMStH is computed in G'' . The computation is performed by the ILP described in [29] and modified for the Steiner problem.

At first, a random node $s \in M$ is selected as the source of flows following the hierarchy. From s , a set of flows is computed, one flow to each destination node in $M \setminus \{s\}$. The integer values associated with the arcs indicate the number of transited flows. The image is composed of arcs with non-zero flows. Note that the number of flows is not equal to the number of occurrences of the arc in the DCMStH.

ILP variables:

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

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

Objective:

The cost of the weighted image (the cost of the corresponding hierarchy) must be minimized.

$$\text{minimize } \sum_{m \in V''} \sum_{n \in \Gamma_m^+} x(m, n) * c(m, n) \tag{1}$$

The values of the variable x indicate the number of incoming and outgoing arcs (the predecessors and the successors) of the nodes (all occurrences combined) in the DCMStH. In the optimum, each node has a minimal number of occurrences. The number of occurrences of a non-source node is equal to the number of its predecessors. Only the first node occurrence of the source has no predecessor. Moreover, the number of occurrences must be sufficient to connect the outgoing arcs. The degree constraints can be formulated as follows.

At the source,

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

At the non-source node m ,

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

The following lower bounds are also true.

At the source,

$$\sum_{n \in \Gamma_s^+} x(s, n) \geq D(s) + (D(s) - 1) * \left(\sum_{n \in \Gamma_s^-} x(n, s) - 1 \right) \tag{4}$$

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

$$\sum_{n \in \Gamma_m^+} x(m, n) \geq (D(m) - 1) * \left(\sum_{n \in \Gamma_m^-} x(n, m) - 1 \right) \quad \forall m \in V'' \setminus \{s\}, D(m) \neq 1 \tag{5}$$

A node with a degree bound of one cannot be the start point of outgoing arcs.

$$\sum_{n \in \Gamma_m^+} x(m, n) = 0 \quad \forall m \in V'' \setminus \{s\}, D(m) = 1 \tag{6}$$

Each node in M except the source has a predecessor in the hierarchy.

$$\sum_{n \in \Gamma_m^-} x(n, m) \geq 1 \quad \forall m \in M \setminus \{s\} \tag{7}$$

Flows can express the connectivity between the nodes in M .

The arbitrary source s is the start point of one flow by destination in $M \setminus \{s\}$.

$$\sum_{n \in \Gamma_s^+} F(s, n) = |M| - 1 \tag{8}$$

Each node in $M \setminus \{s\}$ is the destination of one and only one flow.

$$\sum_{n \in \Gamma_m^+} F(m, n) = \sum_{n \in \Gamma_m^-} F(n, m) - 1 \quad \forall m \in M \setminus \{s\} \tag{9}$$

The other nodes conserve the flows.

$$\sum_{n \in \Gamma_m^+} F(m, n) = \sum_{n \in \Gamma_m^-} F(n, m) \quad \forall m \in V'' \setminus M \tag{10}$$

If an arc does not belong to the solution, it does not carry any flow. Conversely, each arc belonging to the solution carries at least one flow, and the number of flows carried cannot be less than the number of occurrences of the arc.

$$F(m, n) = 0 \quad \forall (m, n) \in A'' \text{ if } x(m, n) = 0 \tag{11}$$

$$F(m, n) \geq x(m, n) \quad \forall (m, n) \in A'' \text{ if } x(m, n) > 0 \tag{12}$$

The number of occurrences of an arc is limited by the number of the “destinations”.

$$x(n, m) < |M| \quad \forall (m, n) \in A'' \tag{13}$$

This linear program computes the directed images of the solution in G'' . The solution (the DCMStH) must be determined from its image.

5.4. Reconstruction of an Optimal Hierarchy

The variable $x(n, m)$ indicates how many times the arc $(n, m) \in A''$ is used in the directed solution. At first, a minimal Steiner arborescence T'' rooted at s of the solution $H = (T'', h, G'')$ can be built.

If the number of occurrences of some arcs is different from one, several arborescences may correspond to the image. We let $i(n) = \sum_{m \in \Gamma^-(n)} x(n, m)$ and $o(n) = \sum_{m \in \Gamma^+(n)} x(n, m)$ be the number of incoming and outgoing arcs of the node n , respectively. In an arborescence, each node occurrence has one and only one predecessor (except the source). The node n must be duplicated $i(n)$ times in the arborescence (the source must be duplicated $i(s) + 1$ times). With each occurrence of n , one incoming arc can be associated. The successors can be distributed arbitrarily among the occurrences of n while respecting the degree constraints: each occurrence of n can have at most $D(n) - 1$ successors. For instance, all occurrences of n except one can have the maximal degree $D(n)$.

The outline of the construction is as follows.

1. For each node n , a set of successors is created from values $x(n, m)$. (For each node $m \in \Gamma^+(n)$, $x(n, m)$ occurrences of m should be added to the set of successors of n). Each node occurrence is labeled by its identifier in G .
2. The construction of the labeled tree T'' starts at the source node s .
3. The first occurrence of the source has no predecessor. If the number of successors is greater than $D(s)$, the first occurrence of the source has $D(s)$ successors. The remaining successors are associated with the other source occurrences.
4. The nodes are processed recursively. We let n be the current node.
 - (a) The successors (using the corresponding outgoing arcs) are distributed among the occurrences of n . All occurrences except one have the maximal degree $D(n)$. (the successors can be distributed between the occurrences of n such that $D(n) - 1$ successors are connected to the first $|\Gamma^+(n)| - 1$ occurrences of n). The remaining successors are associated with the last occurrence.
 - (b) For each successor node, a sub-tree is built recursively.
 - (c) If the current node n is a leaf, there is nothing to do.

Lemma 7. *The result of the given construction is a directed labeled tree $T'' = (V_T, A_T)$.*

Proof. For each node occurrence, a set of successors is connected using an arc from the node to each successor. Each successor is a distinct node occurrence. There, in the no cycle, a node occurrence and its successors form a directed sub-tree. In the tree, the nodes are labeled. Recursively, a successor is the root of a subsequent sub-tree. Trivially, T'' is a labeled directed tree. \square

Theorem 2. *Omitting the direction of the arcs of $T'' = (V_T, A_T)$, an undirected, labeled tree $T = (V_T, E_T)$ is obtained that corresponds to a DCMStH.*

Proof. The computed image is a connected sub-graph in G and covers the node set M . The tree construction processes each node of the image, and each node in M is associated with at least one node of T'' ($M \subseteq V_T$). The node set of V_T is the same in T'' and T with the same labels. The labels of nodes in T correspond to a node in G and the association can be given by h . The hierarchy $H = (T, h, G)$ spans the node set M .

By construction, the nodes in T'' and T 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 M .

It is with minimum cost. Its cost is the sum of the cost of the edge occurrences belonging to it. An edge $\{m, n\}$ is present $x(m, n) + x(n, m)$ times in H (in tree T).

$$c(H) = \sum_{\{m,n\} \in E_T} c(m, n) * (x(m, n) + x(n, m))$$

Because $x(m, n)$ is equal to zero for edges not belonging to the image,

$$c(H) = \sum_{\{m,n\} \in E} c(m, n) * (x(m, n) + x(n, m))$$

The ILP minimizes this sum, H is a DCMStH. \square

6. Approximations

Remember that the problem is NP-hard. Some initial results on its approximation are presented in this section. If there are a few numbers of nodes in V_1 , guaranteed ratios can be found. Namely, in the case of an empty node set V_1 and in the cases where V_1 contains at most two nodes, the computation of approximations is easy but differs from case to case. The existence of approximations in further graph classes is an open question.

6.1. Case of $V_1 = \emptyset$

Theorem 3. *Let us suppose that there is no node with degree bound 1 ($V_1 = \emptyset$). The cost of a DCMStH can be approximated within a ratio $\rho \frac{D}{D-1}$, where ρ is the ratio of the approximation of the Steiner tree and $D = \min_{v \in V} D(v)$.*

Proof. The cost of the minimum Steiner tree T_S without the degree constraints is a lower bound for all connected spanning structures of the group M . The Steiner problem is NP-hard. We let T_A be an approximation of the minimum Steiner tree with ratio ρ . The tree T_A can be decomposed into a set $S = \{S_i, i = 1, \dots, k\}$ of edge-disjoint stars. In each star S_i , a spanning hierarchy H_{S_i} respecting the degree constraint of the (eventually multiplied) central node c_i can be built. Since $V_1 = \emptyset$, possible duplicates concern the best cost edges in the construction of H_{S_i} . 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_{S_i} 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. [28]). The hierarchies $H_{S_i}, i = 1, \dots, k$ spanning the k different stars can be connected, and a hierarchy H spanning the node set V and respecting the degree constraints is obtained. With $D = \min_{i=1, \dots, k} D(c_i)$, since $\frac{D(c_i)}{D(c_i)-1} \leq \frac{D}{D-1}, i = 1, \dots, k$, the cost of the spanning hierarchy is bounded:

$$\begin{aligned}
 c(H) &= \sum_{i=1}^k c(H_{S_i}) \leq \sum_{i=1}^k \frac{D(c_i)}{D(c_i)-1} c(S_i) \leq \\
 &\leq \frac{D}{D-1} c(T_A) \leq \rho \frac{D}{D-1} c(T_S) \leq \rho \frac{D}{D-1} c(DCMStH) \quad (14)
 \end{aligned}$$

□

The decomposition of the Steiner tree into a set of connected stars is illustrated in Figure 3. In the worst case, the degree bound of all nodes is two, D is equal to 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.

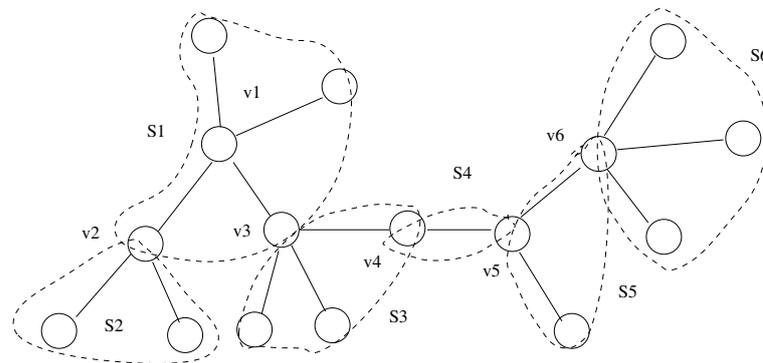


Figure 3. The decomposition of the Steiner tree into a set of stars.

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 4. *In the case of at most two nodes in V_1 , the cost of a DCMStH can be approximated within a factor 2ρ , where ρ is the ratio of the approximation of the minimum Steiner tree.*

Proof. We let a and b be the two nodes in V_1 and let T_A be an approximated tree of the Steiner tree. Following T_A , a walk visiting the nodes of the tree can be computed starting from a and ending at b . This walk W contains the edges of T_A at most twice and corresponds to a spanning hierarchy satisfying the degree constraints (the internal nodes in W have only a degree of two).

$$c(W) \leq 2 * c(T_A) \leq 2 * \rho * c(T_S) \leq 2 * \rho * c(DCMStH).$$

□

Note that if there are nodes with a degree bound greater than two, a hierarchy with lower cost can be computed.

7. Evaluation of the Solutions

We compared three solutions: the minimum Steiner trees, the DCMStT, and the DCMStH if they exist. Random graphs based on the method of Albert–Barabási [30] parameterized by (m_0, m_f, m) were generated in the following two steps:

- (1) creation of a connected small graph of m_0 nodes (a tree for example),
- (2) addition of m_f new nodes such that each of them is randomly connected to at most m existing nodes.

For the algorithms, we used C++ and the Leda library [31]. The ILPs of the three exact solutions were implemented in Gurobi [32]. Fortunately, in the tested random graphs, the computation of the exact solutions was feasible. It took from a few seconds to a few hours.

7.1. Parameter Settings

Some parameters of the topology were variable: the number of nodes in the random graphs, the number of nodes to cover, the degree bounds of the nodes, and the costs of the edges. Medium-size topologies were generated containing 70–170 nodes. The number of edges in the graphs was not deterministic, since the edges were generated randomly. In each experiment presented hereafter, we indicated the average number of edges. The degree bounds were set randomly from an interval of $[D_{min}, D_{max}]$ using a uniform distribution. The edge costs were also uniformly generated from an interval of $[1, C_{max}]$.

7.2. Runs and Results

7.2.1. Effect of the Degree Bounds

In the first case, random graphs were generated with 70 nodes and 30 randomly selected nodes for the group to cover. The following parameters were set: $C_{max} = 5$, $D_{min} = 2$, and D_{max} varied from 3 to 12. Note that with these values, the node set V_1 was always empty. For each value of D_{max} , 10 random graphs were generated, and the values in Table 2 are the average values of 10 runs.

Table 2. Results by varying D_{max} .

D_{max}	c(StT)	arc(StT)	c(DCMStT)	arc(DCMStT)	c(DCMStH)	arc(DCMStH)
3	75.56	37.04	92.898	38.9388	84.28	42.32
4	73.1	35.78	83.9	37.42	79.2	39.28
5	72.82	36.1	82.58	37.72	77.4	39.38
6	74.2	36.12	80.68	37	77.72	38.6
7	74.62	36.3	80.42	37.2	78.08	38.76
8	71.94	36.08	77.32	36.82	74.86	38
9	73.8	35.72	77.68	36.2	76.12	37.22
10	71.46	35.78	74.96	36.36	73.34	37
11	73.18	36.28	77.14	36.78	75.62	37.82
12	70.58	35.88	73.54	36.04	72.42	37.1

In the table, c(St), c(DCMStT), and c(DCMStH) are the average costs of the minimum Steiner trees, the degree-constrained Steiner trees, and the hierarchies, respectively; arc(St), arc(DCMStT), and arc(DCMStH) are the number of arcs in the three solutions.

7.2.2. Effect of the Leaves in V_1

In the second case, we examined the effect of the nodes with a degree bound of one. For this, $D_{min} = 1$ was set. In the generated graphs, the number of nodes and edges were 70 and 136, respectively. The average number of arcs in the directed graphs after the reduction is indicated in Table 3.

Table 3. Effect of nodes in V_1 by varying D_{max} .

D_{max}	$ V_1 $	$ A $	c(StT)	arc(StT)	c(DCMStT)	arc(DCMStT)	c(DCMStH)	arc(DCMStH)
3	22.5	184	74	37	89	39.2	81.6	41.2
4	17.3	198.4	71	35.8	77.6	36.2	75.6	37.8
5	13.1	214.4	69.4	35.2	75.2	36.2	73	37.2
6	11.4	227.9	69.6	36.2	78.2	37.6	74.2	39.6
7	10.3	232.9	73.4	37.2	78.6	38	76.4	39.2
8	7.2	245.2	77	35.8	80.8	36.6	79.6	37.2
9	9.2	233.4	71.6	34.8	74.4	35.8	74	37.2
10	7.5	242.3	78	36.8	84	37	82	38
11	6.7	246.9	74.4	36.6	81	38.6	78.6	40.4
12	5.1	251.5	76.6	35.8	79.4	36.6	78.6	36.6

The average number of nodes in V_1 and the number of cases without solution are also indicated. When D_{max} is increased, the proportion of nodes in V_1 decreases (the degree bounds are uniformly distributed in $[D_{min}, D_{max}]$). The number of arcs $|A|$ after the reduction increases. Note that the number of failed computations can be high when the number of nodes in V_1 is high due to the separators. In some cases, the degree-constrained Steiner tree cannot be computed even though the hierarchy exists. The values correspond to the average values of the cases with successful computation.

7.2.3. Effect of the Group Size

In random graphs with 70 nodes and using the following parameters, the effect of the group size on the optimal spanning structures was analyzed. The results are in Table 4. Using $D_{min} = 1$ and $D_{max} = 10$, 10% of the nodes were nodes with a degree bound of one. The edge costs was selected from the interval of $[1, 4]$. The group size was varied from 2 to 47. For each configuration, 50 runs were executed.

Table 4. Results by varying $|M|$.

$ M $	c(StT)	arc(StT)	c(DCMStT)	arc(DCMStT)	c(DCMStH)	arc(DCMStH)
2	7.14286	3.42857	7.14286	3.4	7.14286	3.51429
7	21	11.4324	21.9189	11.7297	21.4324	11.6486
12	30.303	17.1515	31.5455	17.5152	31.0303	17.697
17	42.0741	23.4074	44.2692	23.6538	43.8889	24.1852
22	53.2222	29	56.4722	29.6667	54.8889	29.8333
27	60.7353	33.0882	63.7941	33.6176	62.5294	34.3824
32	69.0345	37.6552	71.963	38.1852	71.4286	39.3929
37	76.2222	42.3611	81.9143	43.2571	79.7778	44.8333
42	87.6364	46.6667	93.1935	47.6452	91.2727	49.1212
47	92.2	50.9429	99.3824	51.5882	96.3714	53.5714

Note that the first line corresponds to the computation of the shortest path. By increasing the size of the group M , the difference between the costs of the three spanning structures increases.

An interesting result can be observed from the three series of calculations. Even though there are more arcs in the DCMStH than in the DCMStT, the cost of the DCMStH is more favorable. This is the consequence of the fact that DCMStHs, to circumvent the constraints, can return several times to some nodes and use favorable edges.

8. Conclusions and Perspectives

A special version of the degree-constrained Steiner problem is initiated in our paper. The degree bounds are the maximal capacities of the nodes and the nodes can be used several times in the spanning structure. The degree constraints must be satisfied for each occurrence of the nodes. This model can be applied to several scheduling and routing problems. It is demonstrated that the optimum of the minimum cost capacity degree-constrained partial spanning problem is a hierarchy. The problem is NP-hard. As an important result, necessary and sufficient conditions for the existence of the solution are formulated.

An ILP-based computation of the degree-constrained minimum-cost Steiner hierarchies is also presented.

For the cases where the number of nodes with a degree bound of one is less than three, we present an approximation of the analyzed NP-hard problem.

The degree-constrained Steiner hierarchies are compared to the degree-constrained Steiner trees and also to the Steiner trees without constraint. The experiments also prove that, in some cases, degree-constrained Steiner hierarchies may exist where Steiner trees respecting the constraints do not exist. Moreover, even if trees that respect the constraints exist, it is possible that an optimal hierarchy with lower cost can be found. Future investigations are needed for the development of efficient heuristics and improved approximation ratios. Artificial Intelligence is a good candidate to create efficient solutions.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Papadimitriou, C.H.; Yannakakis, M. The complexity of restricted minimum spanning tree problems. In Proceedings of the Automata, Languages and Programming, Graz, Austria, 16–20 July 1979; Maurer, H.A., Ed.; Springer: Berlin/Heidelberg, Germany, 1979; pp. 460–470.
2. Cieslik, D. The vertex degrees of minimum spanning trees. *Eur. J. Oper. Res.* **2000**, *125*, 278–282. [[CrossRef](#)]
3. Ruzika, S.; Hamacher, H.W. A Survey on Multiple Objective Minimum Spanning Tree Problems. In *Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation*; Lerner, J., Wagner, D., Zweig, K.A., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 104–116. [[CrossRef](#)]
4. 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](#)]
5. Zhou, F.; Molnar, M.; Cousin, B. Is Light-Tree Structure Optimal for Multicast Routing in Sparse Light Splitting WDM Networks? In Proceedings of 18th International Conference on Computer Communications and Networks, San Francisco, CA, USA, 3–6 August 2009. [[CrossRef](#)]
6. Mukherjee, B. *Optical WDM Networks (Optical Networks)*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 26–27.
7. Ljubić, I. Solving Steiner trees: Recent advances, challenges, and perspectives. *Networks* **2021**, *77*, 177–204. [[CrossRef](#)]
8. Rehfeldt, D.; Franz, H.; Koch, T. Optimal connected subgraphs: Integer programming formulations and polyhedra. *Networks* **2022**, *80*, 314–332. [[CrossRef](#)]
9. Yang, B.; Zheng, W. A Powerful Local Search Method for Minimum Steiner Tree Problem. In Proceedings of the Web and Big Data, Jinhua, China, 30 August–1 September 2024; Zhang, W., Tung, A., Zheng, Z., Yang, Z., Wang, X., Guo, H., Eds.; Springer: Singapore, 2024; pp. 50–66.
10. Yang, B.; Zheng, W. Near-optimal Steiner tree computation powered by node embeddings. *Knowl. Inf. Syst.* **2023**, *65*, 1–21. [[CrossRef](#)]
11. Long, N.B.; Anh, D.T.; Ban, H.B.; Binh, H.T.T. An online transfer learning based multifactorial evolutionary algorithm for solving the clustered Steiner tree problem. *Knowl.-Based Syst.* **2024**, *296*, 111870. [[CrossRef](#)]
12. 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.
13. Voß, S. Problems with Generalized Steiner Problems. *Algorithmica* **1992**, *7*, 333–335. [[CrossRef](#)]
14. Bauer, F.; Varma, A. Degree-constrained multicasting in point-to-point networks. In Proceedings of the INFOCOM '95: The Fourteenth Annual Joint Conference of the IEEE Computer and Communication Societies, Boston, MA, USA, 2–6 April 1995; Volume 1, p. 369.
15. Agrawal, A.; Klein, P.; Ravi, R. *How Tough Is the Minimum-Degree Steiner Tree?: A New Approximate Min-Max Equality*; Technical Report TR CS-91-49; Brown University: Providence, RI, USA, 1991.
16. Könemann, J.; Ravi, R. Quasi-polynomial Time Approximation Algorithm for Low-Degree Minimum-Cost Steiner Trees. In Proceedings of the FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science, Mumbai, India, 15–17 December 2003; Volume 2914, pp. 289–301. [[CrossRef](#)]
17. Lau, L.C.; Singh, M. Additive approximation for bounded degree survivable network design. In Proceedings of the STOC '08: 40th Annual ACM Symposium on Theory of Computing, Victoria, BC, Canada, 17–20 May 2008; pp. 759–768. [[CrossRef](#)]
18. Lau, L.C.; Naor, J.S.; Salavatipour, M.R.; Singh, M. Survivable Network Design with Degree or Order Constraints. *SIAM J. Comput.* **2009**, *39*, 1062–1087. [[CrossRef](#)]
19. Louis, A.; Vishnoi, N.K. Improved Algorithm for Degree Bounded Survivable Network Design Problem. In Proceedings of the Algorithm Theory—SWAT 2010, Bergen, Norway, 21–23 June 2010; Kaplan, H., Ed.; Springer: Berlin/Heidelberg, Germany, 2010; pp. 408–419.
20. Bansal, N.; Khandekar, R.; Nagarajan, V. Additive guarantees for degree bounded directed network design. In Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing, Victoria, BC, Canada, 17–20 May 2008; pp. 769–778.
21. Amini, O.; Peleg, D.; Pérennes, S.; Sau, I.; Saurabh, S. Degree-Constrained Subgraph Problems: Hardness and Approximation Results. In Proceedings of the Approximation and Online Algorithms, Karlsruhe, Germany, 18–19 September 2008; Bampis, E., Skutella, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2009; pp. 29–42.
22. Khandekar, R.; Kortsarz, G.; Nutov, Z. On some network design problems with degree constraints. *J. Comput. Syst. Sci.* **2013**, *79*, 725–736. [[CrossRef](#)]
23. Lin, G.; Xue, G. On the terminal Steiner tree problem. *Inf. Process. Lett.* **2002**, *84*, 103–107. [[CrossRef](#)]
24. Fuchs, B. A note on the terminal Steiner tree problem. *Inf. Process. Lett.* **2003**, *87*, 219–220. [[CrossRef](#)]
25. Liers, F.; Martin, A.; Pape, S. Binary Steiner trees: Structural results and an exact solution approach. *Discret. Optim.* **2016**, *21*, 85–117. [[CrossRef](#)]

26. Dinitz, M.; Kortsarz, G.; Li, S. Degrees and Network Design: New Problems and Approximations. In Proceedings of the Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2024), London, UK, 28–30 August 2024; Kumar, A., Ron-Zewi, N., Eds.; Leibniz International Proceedings in Informatics (LIPIcs); Dagstuhl: Saarland, Germany, 2024; Volume 317, pp. 3:1–3:17. [[CrossRef](#)]
27. Song, J.; Lu, Q.; Tian, B.; Zhang, J.; Luo, J.; Wang, Z. Prove Symbolic Regression is NP-hard by Symbol Graph. *arXiv* **2024**, arXiv:2404.13820.
28. 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.
29. Molnar, M. Degree-Constrained Minimum Spanning Hierarchies in Graphs. *Algorithms* **2024**, *17*, 467. [[CrossRef](#)]
30. Albert, R.; Barabási, A.L. Statistical mechanics of complex networks. *Rev. Mod. Phys.* **2002**, *74*, 47–97. [[CrossRef](#)]
31. Mehlhorn, K.; Näher, S. *LEDA: A Platform for Combinatorial and Geometric Computing*; Cambridge University Press: Cambridge, UK, 1999.
32. Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual. 2024. Available online: <https://www.gurobi.com> (accessed on 1 November 2024).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.