Exact solution for bounded degree connected spanning problems

Massinissa Merabet, Sylvain Durand, Miklós Molnár

The exact solution of the degree bounded connected spanning problems

Massinissa Merabet[a], Sylvain Durand[b], Miklos Molnar[c]

[a]*LIRMM, University Montpellier 2, Montpellier, France (e-mail: massinissa.merabet@lirmm.fr).*
[b]*LIRMM, University Montpellier 3, Montpellier, France (e-mail: sylvain.durand@lirmm.fr)*
[c]*LIRMM, University Montpellier 2, Montpellier, France (e-mail: miklos.molnar@lirmm.fr)*

Abstract

The degree constrained minimum spanning tree problem (DCMST) is well known since all researches on degree constrained spanning structures are based on spanning trees, but the routing in networks do not explicitly impose a sub-graph as solution. A more flexible structure to solve the degree constrained spanning structure is proposed in [Mol08]. This structure is called hierarchy. In contrast with trees, this structure is not a sub-graph but a homomorphism of a tree in a graph. In this paper we investigate the problem of degree constrained minimum spanning hierarchy (DCMSH). Given a connected, edge-weighted graph G and a positive integer R, the problem DCMSH consists in finding a minimum spanning hierarchy of G such that the degree of each vertex in the hierarchy is less than or equal to R. As DCMST, the DCMSH problem is also NP-complete. In this paper, the first ILP formulation of this problem is given. Some theorems and propositions concerning this problem are proved, which allow to add valid inequalities to the ILP. To evaluate the difference of cost between optimal trees and optimal hierarchies, we solve both linear programmes (of DCMST and DCMSH) using GLPK applied to a weighted graph modelling the NSF network first and to random graphs generated by NetGen after. it appears from these experiments that the cost of the optimal degree constrained spanning hierarchy is always lower than the cost of the optimal degree constrained spanning tree.

Preprint submitted to Elsevier *September 11, 2012*

1. Introduction

Several problems in the design of optical networks can be modeled as finding a network obeying certain connectivity specifications. For instance, the network may be require to connect all the nodes (spanning tree problem), a specified subset of the nodes (Steiner tree problem) or only interconnect a set of set of nodes (a generalized Steiner forest problem).

Finding a (minimum) spanning tree of a graph is polynomial [Kru56]. contrariwise, finding a spanning tree of whose nodes do not exceed a given maximum degree with minimum total edge length is NP-compete [DH68] and known as the Degree Constrained Minimum Spanning Tree problem (DCMST). The DCMST problem has many practical applications, for example to support broadcast in optical network where nodes should be equipped with optical splitters which split the light signal into several copies. For this reason it is necessary to upper bound vertex degrees in the topology graph such that splitting capability is represented by a degree constraint. The DCMST problem also arises in many other areas such as the design of integrated circuits, energy networks, transportation, logistics, sewage networks and plumbing for maximum network reliability, and optimality such as rerouting of traffic in case of vertex failures, and improve the network performance by distributing the traffic across many vertices [KES01].

All researches on degree constrained spanning problems are based on spanning trees but the routing in networks do not explicitly impose a sub-graph as solution. A more flexible structure to solve the degree constrained spanning problem is proposed in [Mol08]. In contrast with trees, this structure (called hierarchy) is not a sub-graph but a homomorphism of a tree in a graph.

As we will show in the paper, the interest of the hierarchy concept to solve the constrained spanning problem is evident often the convergence of the graph is possible with the hierarchies, even if the constraints exclude the spanning trees. In some other cases, the cost of spanning hierarchy can be less than the cost of the optimal spanning tree.

In this paper we investigate the problem of degree constrained minimum spanning hierarchy (DCMSH). The problem DCMSH consists in finding a minimum spanning hierarchy in a graph such that the degree of each vertex in the hierarchy is less than or equal to a given integer R. As DCMST, the DCMSH problem is also NP-complete. In this paper, the first ILP formulation of the DCMSH problem is given. To improve the performance of the ILP Some theorems and propositions concerning this problem are proved, which allow to add valid inequalities to the ILP. To compare the cost of optimal trees and optimal hierarchies, the solution (of DCMST and DCMSH) are computed using GLPK. It appears from these experiments that the cost of the optimal degree constrained spanning hierarchy is often lower than the cost of the optimal degree constrained spanning tree.

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

2. Previous work

Degree Constrained Minimum Spanning Tree problem (DCMST) is firstly introduced and solved by linear programming in [DH68]. It is defined as follow:

Definition 2.1. *Let $G = (V_G, E_G)$ be an undirected connected, edge-weighted graph such that V_G is the set of vertices and E_G the set of edges. $W : E_G \rightarrow R_*^+$ be a weight function and R a positive integer. The problem DCMST consists in finding a minimum spanning tree of G such that the degree of each vertex in the tree is less than or equal to R.*

In [DH68] the parallel between the DCMST and the Hamiltonian problems was drawn. Evidently finding a minimum spanning tree with degree bound equal to two is equivalent to finding a Hamiltonian path, therefore DCMST is NP-complete [GJ79]. In fact, even approximating optimal DCMST solutions within a constant factor is NP-hard [RMR+93, BDK96, KC00].

In [BV95] it is shown that for some instances there is no spanning tree which meets the degree constraint. The instance G in Figure 1 can not be spanned by a tree with degree bound $R = 3$ because all spanning trees of G must have the degree of vertex b or e equal to 4.

Due to the hardness of the DCMST problem, a few heuristics had been introduced. The branch and bound algorithm of Narula and Ho [NH80] appears to be the very first heuristic approach proposed for the problem. The branch and bound algorithm of Savelsbergh and Volgenant [SV85] improved in [NH80] by using more efficient branching rules and by implementing an edge exchange scheme that allows variable fixation tests to be carried out. Gavish [Gav82] was the first to use Lagrangian relaxation for the DCMST problem. Other approaches are known : colony optimization [BHE05, BZ06], primal and dual approach [NH80], evolutionary algorithms [KES01, KC00], genetic algorithms [KES01, RJ00], parallel algorithms [MDL99], problem space search [KES01] and variable neighbourhood search [RS02].

The reality of routing problems in networks explicitly impose nor a tree neither a sub-graph as solution. As it is shown in [KM02], the optimal/feasible routes satisfying several QoS constraints are not always trees, because in some cases, cycles are presents in the routing structure. To solve the routing Maher and Deogun [AD00] propose in the case of optical networks with any splitter to find an trail that start from the source and visits all destination nodes. This trail is walk which visiting several time vertices. Molnar [Mol08] introduced a more flexible structure called hierarchy, which can correspond to the minimum cost connected spanning structure.

3. Problem formulation

A hierarchy is neither a tree nor a sub-graph. It is a graph related structure obtained by a homomorphism of a tree in a graph. In graphs, a homomorphism can be defined as follows [HZ94]: Let $Q = (W, F)$ and $G = (V, E)$ two (undirected) graphs. An application $h : W \to V$ associating a vertex in V to each vertex in W is a homomorphism if the mapping preserves the adjacency: $(u, v) \in F$ implies $(h(u), h(v)) \in E$. If Q is graph which has not any vertex with degree greater than two, (Q, h, G) defines a walk in G, if in addition to this the application h is injective, then the walk is an elementary walk in G. By cons if several vertices in W can correspond to a same vertex in V then (Q, h, G) gives a non-elementary walk in G [Mol11]. If Q is a connected graph without cycle (a tree) then the triple (Q, h, G) defines a hierarchy in G.

Figure 1 shows an example of a hierarchy. Each vertex of the tree T is associated with a unique vertex of the graph G. In reverse direction, a vertex of G can be mapped (or not) to several vertices in T. A vertex in T can be identified by the vertex in G with which it is associated. To distinguish the occurrences related to a same vertex v in G, we will use the labels $v^1, v^2, ..., v^k$ if needed.
Regarding the mapping of vertices from G to the original tree T, a hierarchy can be given by two sets:

$$H = (U, D)$$

where U is the set of edges in H using the labels from G. U may contain a vertex label from G several times. the repetitions of the edges of G are also possible in D. For the instances, the two sets of the hierarchy in figure 1 are:

$U = \{a^1, b^1, b^2, c^1, d^1, e^1, f^1, g^1, h^1\}$
$D = \{(b^1, c^1), (b^1, a^1), (a^1, b^2), (a^1, b^2), (b^2, f^1), (b^2, d^1), (d^1, e^1), (e^1, h^1), (e^1, g^1)\}$

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

Before analysing proprieties and ILP formulation, some notations to define connected spanning structures for graphs supposing degree constraints are fixed :

We denote by $G = (V_G, E_G, W)$ a connected weighted graph G such that V_G is the set of vertices and E_G the set of edges and a weight function $W : E_G \rightarrow R_*^+$. We denote by $H = (T, h, G)$ a hierarchy on G. It is necessary to emphasis the different pre-images of a vertex $v \in V_G$ in a given hierarchical tree. For that, we label $v^1, v^2, ..., v^\alpha$ the α pre-images of v in the hierarchical tree (u is pre-image of v under h if $h(u) = v$). We denote by $OC_H(v)$ the number of pre-images of v in H and by $d_G(v)$ the degree of v in the graph G. We denote by $N_M(G)$ the number of vertices with degree equal to M in G.

The degree of a vertex occurrence in a hierarchy should also be defined. Let $H = (T, h, G)$ be a hierarchy, u is a vertex in T such that the mapping associates $v \in V_G$ ($h(u) = v$) to this vertex in the hierarchy.

Definition 3.1. *The degree $d_H(v^i)$ of the vertex occurrence v^i in H is : $d_H(v^i) = d_T(w)$ where $d_T(w)$ gives the degree of w in T.*

Since the spanning structure must be connected but not obligatory a subgraph, the spanning problem can be reformulated using hierarchies [Mol11]. It is defined as follows.

Definition 3.2. *The problem DCMSH consists in finding a minimum spanning hierarchy of G such that the degree of each vertex in the hierarchy is less than or equal to R.*

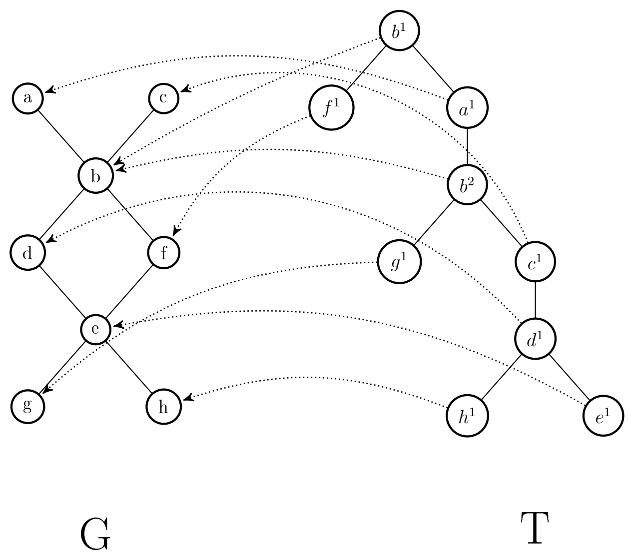


Figure 1: Mapping of vertices for a hierarchy

Figure 1 illustrate how a hierarchy can satisfy the degree constraints. In all connected spanning trees of G at least one of the vertices b or e have a degree equal to 4. However the maximum degree vertices in H is equal to 3 because of duplication of the vertex b. So there is no feasible spanning tree of G for the degree bounded spanning problem with $R = 3$ but there is at least one feasible hierarchy of G (the mapping T of hierarchy of G).

4. Properties of degree bounded hierarchies

In order to efficiently construct the optimal hierarchy, it is important to study its relevant properties. In this section we present an upper bound of the vertex occurrences in optimal hierarchy of the DCMSH problem (Theorem 4.1). This propriety is directly used in our ILP.

Theorem 4.1. *Let $H = (T, h, G)$ be an optimal hierarchy in G for DCMSH problem with degree constraint R. If $R = 2$, the number of occurrences of v is less or equal to $|V_G| - 1$. Else, the number of occurrences of vertex $v \in V$ is less or equal to $\frac{|V_G|-2}{R-2}$.*

Proof. $\underline{R = 2}$**:**

Trivially, the tree T is a path and the spanning hierarchy H is a walk. Let be $v \in V$ and $v^1, v^2, ..., v^\alpha$ the α occurrences if this vertex in the optimum enumerated from an arbitrary end point of the walk. That is, v^i is positioned before v^{i+1} in the walk. The cycle from v^i to v^{v+1} must contains at most a vertex occurrence $u \in V$ which is exclusively covered by the cycle (v^i, v^{i+1}). If such an exclusively covered vertex does not exist, than the cycle (v^i, v^{i+1}) can be deleted and the remained walk H' covers the same set of vertices. So, H can not be the cost minimum solution.
The two end points of the walk cannot correspond to the same vertex $v \in V$. If the two end points are labelled by the same vertex v, then one of them can be deleted without loss of the graph coverage. Obviously, there are at most as many occurrences of v in the optimum as exclusively covered vertices on the "cycle" of the walk. Trivially, there are at most $|V| - 1$ possibilities.

$\underline{R > 2}$**:**

Proposition 4.2. *For any tree $T(V_T, E_T)$ with $|V_T| \geq 2$: $N_M(T) \geq h \Rightarrow N_1(T) > h * (M - 2) + 2$.*

Proof. By induction :

1. The base case : Trivially true for $h = 0$ and $|V_T| \geq 2$ since any tree with at least 2 vertices has at least 2 leaves.
2. Suppose that the proposition true for any $h' < h$. Let $T = (V_T, E_T)$ be a tree with $N_M(T) = h$. Let $s \in V_T$ such that $d_T(v) = M$ and let $s_1, s_2, s_3, ..., s_M \in V_T$ neighbours of v . We denote by $F = (T_1, ..., T_M)$ the forest obtained by deleting s from T and adding to each connected component C_i a vertex s'_i and edge (s_i, s'_i).

$$
\begin{aligned}
N_1(T) &\geq \sum_i^M N_1(T_i) - M \\
&\geq \sum_i^M (h_i * (M-2) + 2) - M \\
&\geq \sum_i^M (h_i - 1) * (M-2) + M \\
&\geq (h_i - 1) * (M-2) + M \\
&\geq h * (M-2) + 2
\end{aligned}
$$

□

Proposition 4.3. *Any optimal hierarchy H for DCMSH with $OC_H(x)$ occurrences of vertex $x \in V_H$ has degree sum of its occurrences strictly grater then $R * (OC_H(x) - 1)$.*

Proof. Proceed by contradiction. Suppose that there exists an optimal hierarchy with $OC_H(x)$ occurrences of vertex $x \in V_H$ such that the degree sum of that occurrences is less than or equal to $OC_H(x) * R - R$. In this case the number of edges that can be added to the $OC_H(x)$ occurrences without exceeding the degree constraint R is greater than or equal to R.
Choose an arbitrary root different from occurrences of x. Direct H from the root to the leaves. We can remove one occurrence x^i of x and link at most the $R-1$ successors of x^i to $OC_H(x) - 1$ other occurrences of x without exceeding the degree constraint R. When we remove the orientation of arcs we obtain a feasible hierarchy H' with weight strictly lower than H (H' has at least one less edge than H). This is absurd because we assumed that H is optimal.

□

Proposition 4.4. *Given a connected graph $G = (V, E)$ feasible instance for DB-MSH problem with fixed degree constraint R. There exists a hierarchy H of G optimal for DCMSH such that for all $v \in V_G \mid d_G(v) > R$ and $OC_H(v) > 1$ there are at least $OC_H(v) - 1$ occurrences of v with degree equal to R in H.*

Proof. We prove that it is possible to transform any optimal hierarchy H in optimal hierarchy H' who respects conditions of the proposition. Let H a hierarchy of a graph G optimal for DCMSH problem with fixed R, we know that for all $v \in V_G \mid d_G(v) > R$ the degree sum of $OC_H(v)$ occurrences of v is strictly granter than $R * (OC_H(v) - 1)$ and equal or less than $R * OC_H(v)$ (proposition 2) .

If the degree sum of $OC_H(v)$ occurrences of v is equal to $R * OC_H(v)$ then each occurrence of v has degree equal to R in H(conditions of the proposition are respected).

If the degree sum of $OC_H(v)$ occurrences of v is strictly less than $R * OC_H(v)$: We select the vertex v^i occurrence of v in H. We know that the degree of v^i plus the degree sum of the $OC_H(v) - 1$ occurrences of v is strictly grater than $R * (OC_H(v) - 1)$ and in an other hand the degree sum of the $OC_H(v) - 1$ is strictly less than $R * (OC_H(v) - 1) + 1$. Therefore we can remove K edges linking v^i to its successors and connect the K successors to the $OC_H(v) - 1$ other occurrences of v such that the degree sum of $OC_H(v) - 1$ occurrences plus K equal to $R * (OC_H(v) - 1)$. The degree of v^i will be equal or grater than 2 because we have supposed that H is optimal.

□

Suppose that there exists an optimal hierarchy $H(V_H, E_H)$ of G for DCMSH problem with fixed R such as there exist a vertex $x \in V$ for which the number of occurrences in H is strictly grater than $\frac{V-2}{R-2}$. We suppose that this hierarchy respects conditions of the proposition 4.4.
The number of leaves $\#F_T$ in a tree T is grater or equal to $N_M(T)(M-2)+2$. arguing by contradiction, let $N_M(T) > \frac{V-2}{R-2}$ thus $N_M(T) - 1 \geq \frac{V-2}{R-2}$, and $M = R$ for $N_M(T) - 1$ occurrences.

$$\begin{aligned} \#F &\geq (\frac{V-2}{R-2}) * (R-2) + 2 \\ &\geq |V| \quad (Absurd) \end{aligned}$$

□

Proposition 4.5. *Any optimal hierarchy H for DCMSH has total number of vertices less than or equal to* $\frac{|V_G|-2}{R-2} + |V_G|$.

Proof. We know that $N_M(T) \geqslant h \Rightarrow N_1(T) \geqslant h*(M-2)+2$ and $\forall v \in G$: $\sum_{i=1}^{OC_H(v)} d_H(v_i) \geqslant R*(OC_H(v)-1)$. Thus :

$$\begin{aligned}
N_R(H) &\geqslant \sum_{v \in G}(OC_H(v)-1) \\
&\geqslant \sum_{v \in G} OC_H(v) - |V_G| \\
|V_G| \geqslant N_1(H) &\geqslant ((\sum_{v \in G} OC_H(v) - |V_G|)(R-2)+2) \\
\frac{|V_G|-2}{R-2} &\geqslant \sum_{v \in G} OC_H(v) - |V_G| \\
\frac{|V_G|-2}{R-2} + |V_G| &\geqslant \sum_{v \in G} OC_H(v)
\end{aligned}$$

□

Let $a, b \in V_G$. let $a^1, a^2, ..., a^k$ and $b^1, b^2, ..., b^j$ be occurrences of a and b respectively in the hierarchy H in G. The number of occurrences of an edge $(a,b) \in E_H$ is the total number of edges between all occurrences of a and all occurrences of b in H.

Proposition 4.6. *In any optimal hierarchy H optimal for DCMSH with $R = 2$, the number of occurrences of any edge is limited by 2.*

Proof. Let us suppose that an edge (a,b) has the smallest cost and used several times in the optimal walk W^* as it is illustrated by Figure 2.

$$x_1 \overset{W_1}{- - - -} a_1 \text{ ——— } b_1 \overset{W_2}{- - - -} a_2 \text{ ——— } b_2 \overset{W_4}{- - - -} b_3 \text{ ——— } a_3 \overset{W_4}{- - - -} x_2$$

Figure 2: Walk W^*

The graph generated by W^* si a sub-graph $G^* = (V, E^*) \subseteq G$. In the graph illustrated in the Figure 3, there is always a walk W^{**} using (a,b) at most twice.

Classification of the sub-walks $W_1, W_2, ..., W_k$:

1. C_1 = non-elementary cycles closed in a

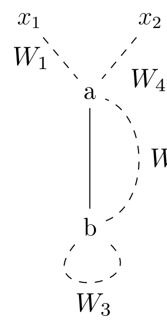


Figure 3: Graph G

2. C_2 = non-elementary cycles closed in b
3. non-elementary walks between a and b
4. non-elementary sub-walks relating an extremity x_i to a or b.

The sub-walks in C_1 (and in C_2 respectively) can be covered without using (a, b). In Figure 4, $W_1\#W_2\#...\#W_k$ is a non-elementary walk.

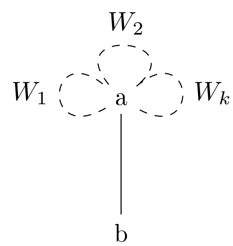


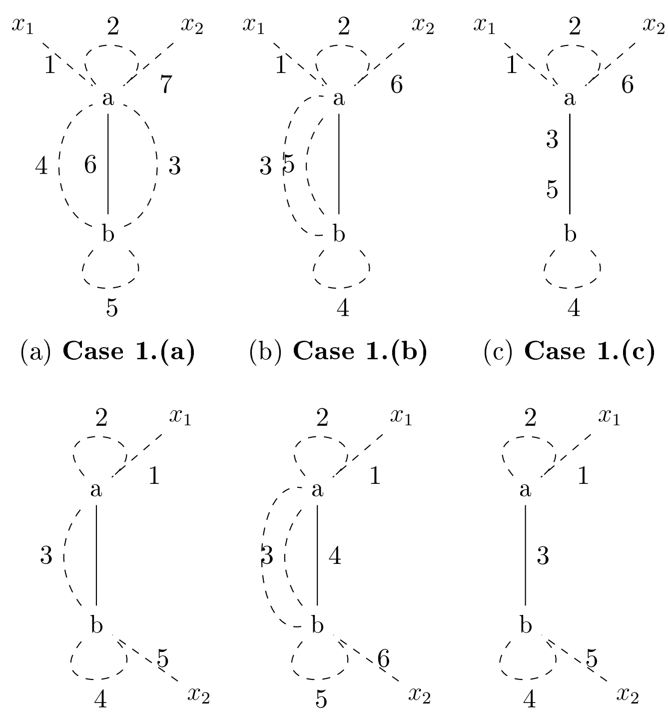
Figure 4: Graph containing C_1

<u>**Cases :**</u>

1. X_1 and X_2 are connected to the same extremity of (a, b) (to a or to b).
   (a) $|C_3|$ = odd. The proposed cover (Figure 5a). (a, b) is used ***only once***.
   (b) $|C_3|$ = even > 0. The proposed cover (Figure 5b). (a, b) is ***not used***.
   (c) $|C_3| = 0$. The proposed cover (Figure 5c). (a, b) is ***used twice***.
2. X_1 and X_2 are connected to two different extremities.
   (a) $|C_3|$ =odd. The proposed cover (Figure 5d). (a, b) is ***not used***.

(b) $|C_3|$ =even > 0. The proposed cover (Figure 5e). (a, b) is ***used once***.

(c) $|C_3| = 0$. The proposed cover (Figure 5f). (a, b) is ***used once***.



(a) **Case 1.(a)** (b) **Case 1.(b)** (c) **Case 1.(c)**

(d) **Case 2.(a)** (e) **Case 2.(b)** (f) **Case 2.(c)**

Figure 5: Illustration of the different cases

□

5. ILP Formulation of the DCMSH problem

As proved in [DH68], it is NP-hard to find a degree constrained spanning tree. Thus, a many integer linear programs (ILP) are proposed using different models [CGI09, DH68]. In [ZMC10] the ILP method was successfully applied to search a light-hierarchy structure with an optimal cost for all optical multicast routing problems.

In this section, the integer linear programming is applied to search the degree constrained optimal hierarchy. In our linear programme the connectivity is preserved by flow formulation. The direction of the flow transiting in an edge must be specified, this is why each vertex of the graph modelling a network is represented by two parameters corresponding respectively to the set of predecessors and the set of successors. Based on the analogy of hierarchies with elementary and non-elementary walks specified in section 2, a hierarchy can be considered as an "non-elementary tree". A flow can transits more than one time in each direction of an edge and each transit must be distinguish from the previous one. This is why each edge is duplicated. The number of duplications is bounded by the theorem 2. Each transit of flow between two vertices must be done on a distinguish duplication of the initial edge. If flow transit on n duplications of an edge then this edge must appear n times in the "non-elementary tree"

(a) Graph G

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

Figure 6: Our transformation of graph to feasible input of the PL

5.1. ILP Formulation

In the following the linear programme is presented :

Network parameters:

$In(m)$: The set of vertices which has an outgoing link leading to node m.

$Out(m)$: The set of vertices which can be reached from m.

$C_{m,n}$: The cost of the link from node m to node n. All duplications of this link have this same cost.

β : Global upper bound on the number of duplications of each vertex (theorem 4.1). In our configuration it correspond to the number duplications of each arc.

ILP variables:

$L_i(m,n)$: Binary variable. Equal to 1 if the occurrence i of the arc (m,n) is in the output graph, equals to 0 otherwise.

$F_i(m,n)$: Commodity flow variable. Denotes the quantity of flow transiting on the occurrence i of the arc (m,n).

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

$$Minimize: \sum_{m \in V} \sum_{n \in Out(m)} \sum_{i=1}^{\beta} L_i(m,n) * C(m,n) \tag{1}$$

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

Degree constraints:

$$\sum_{n \in out(s)} \sum_{i=1}^{\beta} L_i(s,n) \leq R + (R-1) * \sum_{n \in In(s)} \sum_{i=1}^{\beta} L_i(n,s) \tag{2}$$

$$\sum_{n \in out(m)} \sum_{i=1}^{\beta} L_i(m,n) \leq (R-1) * \sum_{n \in In(m)} \sum_{i=1}^{\beta} L_i(n,m) \qquad \forall m,n \in V \setminus \{s\} \tag{3}$$

Constraints (2) and (3) ensure that for each vertex except the source, the number of authorized successors is at most equal to the number of predecessors multiplied by the degree constraint minus 1. Concerning the arbitrary source, the number of authorized successors is equal to the number of predecessors multiplied by the degree constraint minus 1 plus R because the first occurrence of source can has at most R successors despite the fact that it has not predecessors.

$$(R-1) * \sum_{n \in In(m)} \sum_{i=1}^{\beta} L_i(n,m) - \sum_{n \in Out(m)} \sum_{i=1}^{\beta} L_i(m,n) \leq R-1 \qquad \forall m,n \in V \tag{4}$$

Constraint (4) ensures that each vertex can have k predecessors if and only if this vertex has at least $R*(k-1)$ successors. It allows to construct an optimal hierarchy such that each vertex v has at least $OC_H(v)-1$ occurrences with degree equal to R (Proposition 4.4). This constraint is not indispensable but accelerate the search of a specific optimal hierarchy by reducing the search space of the resolution.

$$\sum_{n \in In(m)} L_1(n,m) \geq 1 \qquad \forall m \in V \setminus \{s\} \tag{5}$$

Constraint (5) guarantees that each vertex except the source (source has certainly successors) has at least one predecessors. It ensures that there are no isolate vertex on the output graph.

Connectivity constraints:

In order to guarantee the connectivity of the output graph, we have introduced in our LP some flow constraints modified to better take into account specificities of degree constrained hierarchies.

$$\sum_{i=1}^{\beta} \sum_{n \in Out(s)} F_{(i)}s,n = \sum_{i=1}^{\beta} \sum_{n \in In(s)} F_{(i)}n,s + |V| - 1 \tag{6}$$

The source, like the other vertices of the input graph, can be duplicated in the optimal hierarchy, only the first duplication is really source because of emitting commodity flow. The other duplications are only relays. For this reason, we permit the source to has predecessors. Naturally, the sum of commodity flow outgoing from the source minus the flow incoming to the source is equal to $|V|-1$ which corresponds to the flow originally emitted from the source.

$$\sum_{i=1}^{\beta} \sum_{n \in In(m)} F_{(i)} n, m = \sum_{i=1}^{\beta} \sum_{n \in Out(m)} F_{(i)} m, n + 1 \qquad \forall m \in V - \{s\} \quad (7)$$

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

$$F_{(i)} n, m \geq L_{(i)} m, n \qquad \forall m, n \in V \quad (8)$$

$$F_{(i)} m, n \leq (|V| - 1) * L_{(i)} m, n \qquad \forall m, n \in V \quad (9)$$

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

Valid inequalities:

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

$$L_{i-1}(m, n) \geq L_i(m, n) \qquad \forall m \in V, \ \ \forall n \in Out(m), \ \ 2 \leq i \leq \beta \quad (10)$$

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

$$\sum_{m \in V} \sum_{n \in In(m)} L_1(n, m) \geqslant |V - 1| \qquad \forall m \in V \quad (11)$$

This constraint assure that the total number of arc of occurrence 1 must be imperatively greater than or equal to $|V|$ minus 1.

$$\sum_{m \in V} \sum_{n \in Out(m)} \sum_{i=1}^{\beta} L_i(m, n) \leqslant \frac{|V_G| - 2}{R - 2} + |V_G| \quad (12)$$

According to the proposition 4.5, this constraint assure that the total number of arcs must be less than or equal to $\frac{|V_G|-2}{R-2} + |V_G|$.

5.2. Construction of an optimal hierarchy

The output of our ILP is the values of the variables $L_i(n,m) \setminus \{n,m\} \in E_G$. In this subsection we show that only know this parameter permit to construct the hierarchical tree T corresponding to the mapping of the optimal hierarchy in the graph. Two mainly informations can be extracted from this parameter:

1. Each vertex in T must has at most 1 predecessor. If for a vertex n we have

$$\sum_{i=1}^{\beta} \sum_{m \in In(n)} L_{(i)} m, n = k$$

   then n must be duplicated k times on the optimal hierarchy ($k+1$ times if n is considered as source on the ILP).

2. If for a vertex n we have

$$\sum_{i=1}^{\beta} L_{(i)} m, n = p$$

   then put p arcs between occurrences of m and p occurrences of n such that the p occurrences of n has any predecessors and the occurrences of m do not exceed the degree constraint.

More formally, the algorithm of construction of an optimal hierarchy is shown on the algorithm (1) :

Algorithm 1: Construction of tree corresponding to the mapping of the optimal Hierarchy

input : Parameters $L_i(n, m) \ \backslash \ \{n, m\} \in E_G$
output: Mapping of optimal hierarchy $T(V_T, E_T)$

Integer p, i, j; Table of integer k; $V_H \leftarrow \emptyset$; $E_H \leftarrow \emptyset$; $i \leftarrow 1$; $j \leftarrow 1$;

foreach *vertex* $n \in V_G$ **do**

- $k[i] \leftarrow \sum_{i=1}^{\beta} \sum_{m \in In(n)} L_{(i)} m, n;$
- **if** $n = source$ **then**
  - $V_H \leftarrow V_H \cup \{n^1, n^2, ..., n^{k[i]}, n^{k[i]+1}\};$
- **else**
  - $V_H \leftarrow V_H \cup \{n^1, n^2, ..., n^k[i]\};$
- $i \leftarrow i + 1;$

foreach *vertex* $n \in V_G$ **do**

- $i \leftarrow 1; \quad j \leftarrow 1;$
- $p \leftarrow \sum_{i=1}^{\beta} L_{(i)} m, n;$
- **while** $(i \leq k[i]$ *and* $p > 0)$ **do**
  - **while** $(j \leq k[i]$ *and* $p > 0)$ **do**
    - **if** $(d_T(n^i) < R$ *and* $d_T(m^j) < R$ *and* $|In(n^i)| = 0)$ **then**
      - $E_T \leftarrow E_T \cup (m^j, n^i);$
      - $p \leftarrow p - 1;$
    - $j \leftarrow j + 1;$
  - $i \leftarrow i + 1;$

Delete the orientation

We must prove that the output of the algorithm 1 is an optimal hierarchy $H = (T, h, G)$ with total edges cost equal to c.

Proof. By contradiction: Suppose that there exists a hierarchy $H' = (T', h, G)$ with total edge cost c' strictly lower then c.
To find the values of the ILP variables it suffices to take the opposite way of the algorithm 1: Choose an arbitrary source s and orient H' from this source. If there are P arcs between P occurrences of a vertex n and occurrences of a vertex m then

$$\sum_{i=1}^{P} \sum_{m \in In(n)} L_{(i)}m, n = P \text{ and } \sum_{i=P+1}^{\beta} \sum_{m \in In(n)} L_{(i)}m, n = 0$$

Note that P is less than of equal to β because a vertex can not be duplicated more than β times and a occurrence of a vertex n can only have one predecessor in the hierarchy.
Once the values founded, we must prove that the corresponding solution is feasible for the ILP. For that it is sufficient to prove that the degree constraints are respected:

Constraint 2 and 3: Sum of successors of all duplications of a vertex n in the oriented hierarchy is equal to the sum of successors of n in the transformed graph. Let $E(n^i, m^j)$ equal to 1 if there are an arc between the occurrence i of the vertex n and the occurrence j of the vertex m in the oriented hierarchy.

$$\sum_{m \in Out(n)} \sum_{i=1}^{\beta} \sum_{j=1}^{\beta} E(n^i, m^j) = \sum_{i=1}^{\beta} \sum_{m \in Out(n)} L_{(i)}n, m$$

Sum of predecessors of all duplications of a vertex n in the oriented hierarchy is equal to the sum of predecessors of n in the transformed graph:

$$\sum_{m \in In(n)} \sum_{i=1}^{\beta} \sum_{j=1}^{\beta} E(m^i, n^j) = \sum_{i=1}^{\beta} \sum_{m \in In(n)} L_{(i)}m, n$$

Each vertex except the source has only 1 predecessor and at most $(R-1)$

successors in the oriented hierarchy. Therefore:

$$\sum_{m \in In(n)} \sum_{i=1}^{\beta} \sum_{i=1}^{\beta} E(m^i, n^j) \leq (R-1) * \sum_{m \in Out(n)} \sum_{i=1}^{\beta} \sum_{j=1}^{\beta} E(n^i, m^j)$$

only one occurrences of the arbitrary vertex is considered as source. This occurrence can have successors despite it has not predecessors. Therefore:

$$\sum_{m \in In(s)} \sum_{i=1}^{\beta} \sum_{i=1}^{\beta} E(m^i, s^j) \leq R + (R-1) * \sum_{m \in Out(s)} \sum_{i=1}^{\beta} \sum_{j=1}^{\beta} E(s^i, m^j)$$

This implies that:

$$\sum_{n \in out(m)} \sum_{i=1}^{\beta} L_i(m,n) \leq (R-1) * \sum_{n \in In(m)} \sum_{i=1}^{\beta} L_i(n,m) \qquad \forall m, n \in V \setminus \{s\}$$

$$\sum_{n \in out(s)} \sum_{i=1}^{\beta} L_i(s,n) \leq R + (R-1) * \sum_{n \in In(s)} \sum_{i=1}^{\beta} L_i(n,s)$$

Constraint 4: In the optimal oriented hierarchy we have :

$$\sum_{m \in In(n)} \sum_{i=1}^{\beta} \sum_{i=1}^{\beta} E(m^i, n^j) \leq (R-1) * \sum_{m \in Out(n)} \sum_{i=1}^{\beta} \sum_{j=1}^{\beta} (n^i, m^j)$$

and

$$(R-1) * \sum_{m \in In(n)} \sum_{i=1}^{\beta} \sum_{i=1}^{\beta} E(m^i, n^j) \geqslant (R-1) * \sum_{m \in Out(n)} \sum_{i=1}^{\beta} \sum_{j=1}^{\beta} (n^i, m^j)$$

so

$$(R-1) * \sum_{m \in Out(n)} \sum_{i=1}^{\beta} \sum_{j=1}^{\beta} E(n^i, m^j) - \sum_{m \in In(n)} \sum_{i=1}^{\beta} \sum_{i=1}^{\beta} E(m^i, n^j) \leq (R-1)$$

This implies that:

$$(R-1) * \sum_{n \in In(m)} \sum_{i=1}^{\beta} L_i(n,m) - \sum_{n \in Out(m)} \sum_{i=1}^{\beta} L_i(m,n) \leq R-1 \qquad \forall m, n \in V$$

Flow constraints : The only purpose of the flow constraints is to assure the connectivity of the structure. Hierarchy H' is connected and oriented from the source, so $|V| - 1$ unites of flow can be emitted from the source to the $|V| - 1$ other vertices of occurrence 1. Therefore, the constraint 6 is respected. In the oriented hierarchy each vertex is reachable from the source, so the constraint 7 is also respected.

The Hierarchy $H' = (T', h, G)$ respect all the degree and connectivity constraints. That is absurd because we assumed that H is optimal. □

6. Simulation and performance evaluation

In this section, we present the simulation results to compare the performance of the DCMST problem and the DCMSH problem. the only metric considering in this simulation is the total cost of the spanning structures.

6.1. Simulation setup

In order to demonstrate the advantage of the proposed hierarchy structure, simulation is conduced to compare it with the spanning tree structure. ILP formulations are implemented in C with GLPK package [Mak09] by using the 14−nodes NSF network which is considered as a concrete case and random graphs generated with NetGen [KNS74] to take into account the general case. Figure 10a illustrate the modelling of the NSF Network by a weighted graph. The optimal hierarchy in Figure 10c achieve lower total edges cost than the optimal spanning tree (Figure 10c).

We consider five different values for number of vertices of random graph: $|V| \in \{15, 20, 25, 30, 35\}$. For each value of $|V|$, we consider a single density value (ration between the number of edges and the number of vertices) $d = 2$. Graphs with this density are called sparse graphs. It is natural that the advantage of hierarchies is more evident in sparse graphs. A random graph associated with a fixed number of vertices is called scenario. In order to have a set of significant tests, one hundred feasible instances for the DCMST are generated for each scenario. Then ILP formulations are executed to search the optimal degree constrained spanning tree and the optimal degree constrained spanning hierarchy with $R = 2$ and $R = 3$ for each instances of each scenario.

6.2. Simulation results

To analyse results in meaningful way, it is imperative tu consider the percentage of infeasible instances for a given scenario with a specified bound on the degree R. Note that any instance is feasible for the DCMSH problem whatever R. The proportions of infeasible instances for DCMST problem with degree bound $R = 2$ and $R = 3$ are presented in Table 1.

Spanning trees				
$\|V\|$	15	20	25	30
$R = 2$	45%	52%	69%	79%
$R = 3$	4%	9%	18%	33%

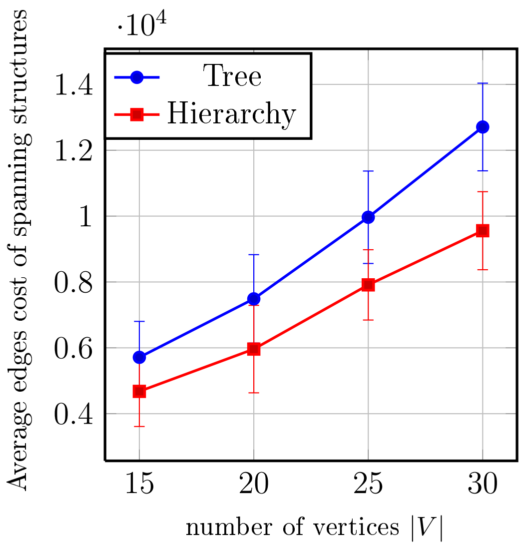
Table 1: Proportion of infeasible instances for DCMST

A feasible graph for DCMST with $R = 2$ implies that this graph is Hamiltonian. However, our random graphs are spares, this is why the probability to generate a Hamiltonian graph is low. So the proportions of infeasible instances for DCMST problem are important and increase with graph size increasing. All the generates random graphs are feasible for the DCMSH. Thus, a first asset of hierarchies beside trees arose : contrary to the DCMST, whatever the topology of a connected graph G this graph is necessary feasible for the DCMSH.

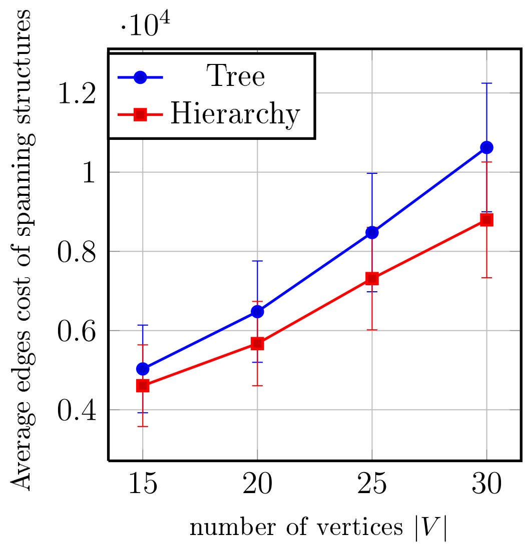
$R = 2$				
$\|V\|$	15	20	25	30
Trees average cost	5711.31	7487.38	9965.15	12706.48
Hierarchies average cost	4677.63	5961.74	7911.76	9556.71
Proportion of amelioration	18.10%	20.37%	20.60%	24.79%
$R = 3$				
$\|V\|$	15	20	25	30
Trees average cost	5031.34	6477.8	8475.19	10623.27
Hierarchies average cost	4608.15	5671.22	7312.24	8795.19
Proportion of amelioration	8.20%	11.80%	13.15%	17.20%

Table 2: Hierarchies average cost versus trees average cost

The numerical results are presented in Table 2. Both for $R = 2$ and $R = 3$, the hierarchy average cost is always lower than the tree average cost. This because in the worst case, the optimal hierarchy for the DCMSH correspond to the optimal tree for the DCMST. The average percentage of improvement of the cost varied between 18% and 25% when $R = 2$ and between 8% and 18% when $R = 3$. The improvement increase with graph size increasing. This because when the graph size is hight, the possibilities of duplications of vertices in hierarchies are important, which arises the probability of improvement of the cost. But the improvement decrease with R increasing. this because when R increase, the probability that the minimum cost spanning tree be optimal for the DCMST increase, mainly in sparse graphs. Knowing that in this case the the minimum cost spanning tree is also optimal for the DCMSH, it is normal that the average improvement decrease.



(a) $R = 2$



(b) $R = 3$

Figure 7: Optimal spanning tree versus optimal spanning hierarchy

The cost improvement in hierarchies is more clear in Figures 7a and 7b which represent the average costs of trees and hierarchies for $R = 2$ and $R = 3$ respectively. As plot in these Figures, the average hierarchies cost is lower than trees in any situation (the curve of the average hierarchies cost is always under the curve of the the average trees cost). In both figures, the standard deviation is not very meaningful. This is why Figures **??** and **??** are added to allow more precision to our results. These figures shows, for $R = 2$

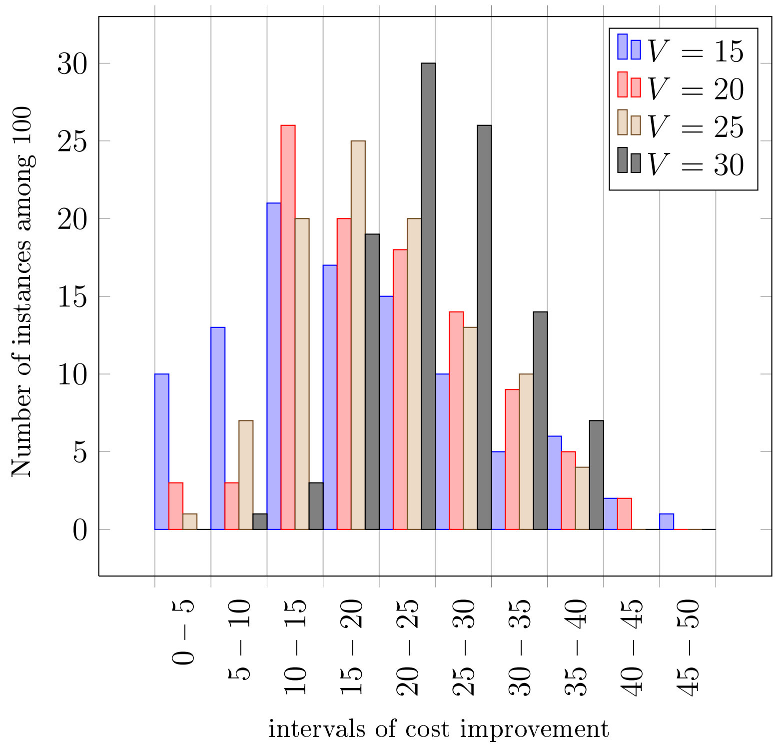


Figure 8: Number of instances regarding intervals of improvement of the cost by hierarchies when $R = 2$

and $R = 3$, the number of instances for which there are an improvement included in a specific interval concerning each scenario. When $R = 2$, it can be observed that for $|V|$ between 20 and 30, hierarchies improve the cost from more than 10 percent of more than 92 instances among 100. It can be also observed that hierarchies improve the cost from more than 20 percent of more than 50 instances among 100 whatever $|V|$. The improvement increase with graph size increasing. Indeed, for $|V| = 30$ the hierarchies improve the cost from more than 20 percent of 96 instances among 100. when $R = 3$, the improvement is less important. But we can regardless observe in Figure **??**, that for $|V|$ between 20 and 30, hierarchies improve the cost from more than 10 percent of more than 50 instances among 100.

7. Conclusion and future works

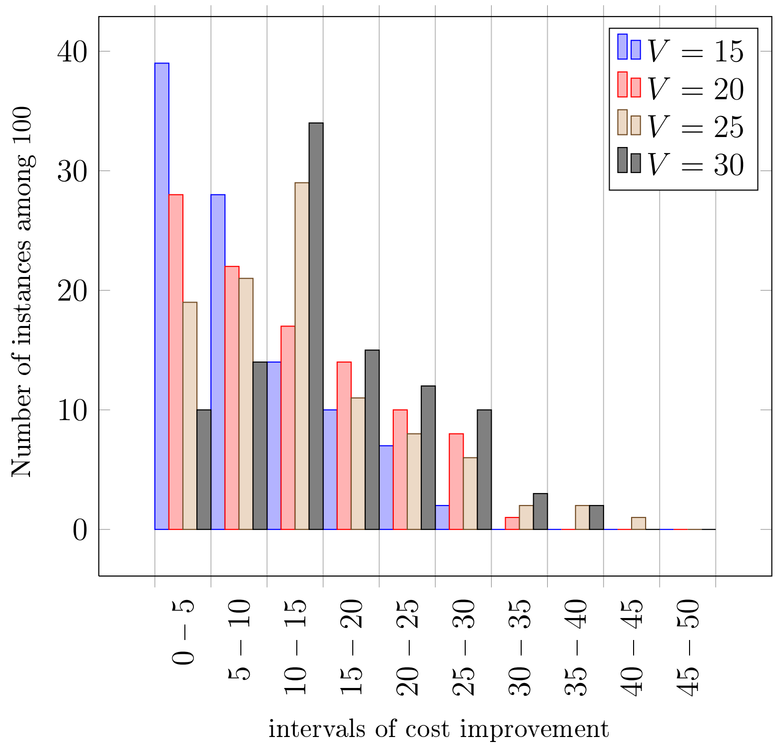
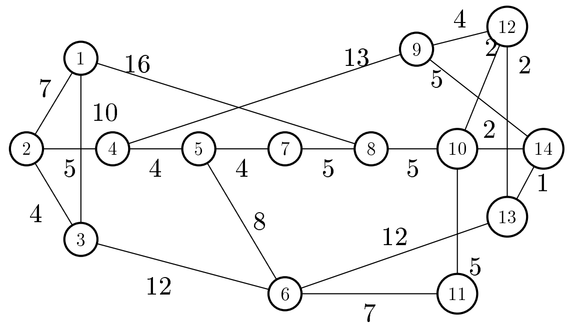


Figure 9: Number of instances regarding intervals of improvement of the cost by hierarchies $R = 3$

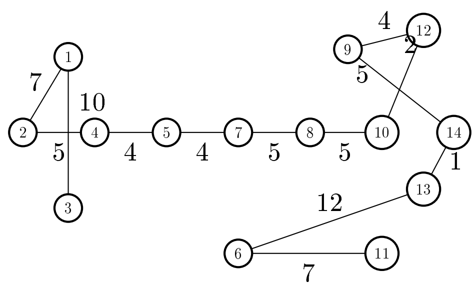
[AD00] Maher Ali and Jitender S. Deogun. Power-Efficient Design of Multicast Wavelength-Routed Networks. In *IEEE Journal of Selected areas in communication*, pages 1852–1862, 2000.

[BDK96] Bruce Boldon, Narsingh Deo, and Nishit Kumar. Minimum-weight degree-constrained spanning tree problem: Heuristics and implementation on an simd parallel machine. *Parallel Computing*, 22(3):369 – 382, 1996.

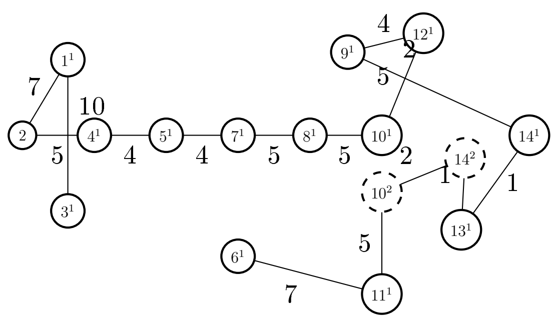
[BHE05] Y. Bau, C. Ho, and H. Ewe. An ant colony optimization approach to the degree-constrained minimum spanning tree problem. In Yue Hao, Jiming Liu, Yuping Wang, Yiu-ming Cheung, Hujun Yin, Licheng Jiao, Jianfeng Ma, and Yong-Chang Jiao, editors, *Computational Intelligence and Security*, volume 3801 of *Lecture*



(a) NSF Network



(b) Spanning tree T optimal solution of the DCMST ($R = 2$) with total edges cost equal to 68.



(c) Hierarchy $H = (T, h, G)$ optimal solution of the DCMSH ($R = 2$) with total edges cost equal to 64.

Figure 10: Optimal tree versus optimal hierarchy of the NSF network

Notes in Computer Science, pages 657–662. Springer Berlin / Heidelberg, 2005.

[BV95] Fred Bauer and Anujan Varma. Degree-constrained multicasting in point-to-point networks. In *in Proc. IEEE INFOCOM*, pages 369–376, 1995.

[BZ06] Thang N. Bui and Catherine M. Zrncic. An ant-based algorithm for finding degree-constrained minimum spanning tree. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, GECCO '06, pages 11–18, New York, NY, USA, 2006. ACM.

[CGI09] R. Cerulli, M. Gentili, and A. Iossa. Bounded-degree spanning tree problems: models and new algorithms. *Comput. Optim. Appl.*, 42(3):353–370, April 2009.

[DBL10] *Proceedings of the 15th IEEE Symposium on Computers and Communications, ISCC 2010, Riccione, Italy, June 22-25, 2010.* IEEE, 2010.

[DH68] N. Deo and L. Hakimi. THE SHORTEST GENERALIZED HAMILTONIAN TREE. In *6^{th} Annual Allerton Conference*, pages 879–888, Illinois, USA, 1968.

[Gav82] Bezalel Gavish. Topological design of centralized computer networks—formulations and algorithms. *Networks*, 12(4):355–377, 1982.

[GHSV02] Luisa Gargano, Pavol Hell, Ladislav Stacho, and Ugo Vaccaro. Spanning trees with bounded number of branch vertices. In *Proceedings of the 29th International Colloquium on Automata, Languages and Programming*, ICALP '02, pages 355–365, London, UK, UK, 2002. Springer-Verlag.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co, New York, NY, USA, 1979.

[HCT02] Jingyi He, S.-H. Gary Chan, and Danny H. K. Tsang. Multicasting in WDM Networks. *IEEE Communications Surveys and Tutorials*, pages 2–20, 2002.

[HZ94] Pavol Hell and Xuding Zhu. Homomorphisms to oriented paths. *Discrete Mathematics*, 132(1–3):107 – 114, 1994.

[KC00] J. Knowles and D. Corne. A new evolutionary approach to the degree-constrained minimum spanning tree problem. *Evolutionary Computation, IEEE Transactions on*, 4(2):125 –134, jul 2000.

[KES01] Mohan Krishnamoorthy, Andreas T. Ernst, and Yazid M. Sharaiha. Comparison of algorithms for the degree constrained minimum spanning tree. *Journal of Heuristics*, 7(6):587–611, November 2001.

[KM02] Fernando Kuipers and Piet Van Mieghem. Mamcra: a constrained-based multicast routing algorithm. *Computer Communications*, 25(8):802 – 811, 2002.

[KNS74] D. Klingman, A. Napier, and J. Stutz. A program for generating large scale capacitated assignment, transportation, and minimum cost flow network problems. *Management Science*, 20(5):814–821, 1974.

[Kru56] Jr. Kruskal, Joseph B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, 7(1):pp. 48–50, 1956.

[Mak09] Andrew O. Makhorin. *GNU Linear Programming Kit (GLPK) v 4.38*, gnu project edition, May 2009.

[MDL99] Li-Jen Mao, Narsingh Deo, and Sheau-Dong Lang. A parallel algorithm for the degree-constrained minimum spanning tree problem using nearest-neighbor chains. In *Parallel Architectures, Algorithms, and Networks, 1999. (I-SPAN '99) Proceedings. Fourth InternationalSymposium on*, pages 184 –189, 1999.

[Mol08] Miklós Molnár. Hierarchies for Constrained Partial Spanning Problems in Graphs. Technical report PI-1900, 2008.

[Mol11] Miklos Molnar. hierarchies to solve constrained spanning problem. *Private communication*, 2011.

[MZQ98] R. Malli, X. Zhang, and C. Qiao. Benefits of multicasting in all-optical networks. In J. M. Senior & C. Qiao, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3531 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 209–220, October 1998.

[NH80] S C Narula and C A Ho. Degree-constrained minimum spanning tree. *Computers and Operations Research*, 7(4):239–249, 1980.

[RJ00] Günther R. Raidl and Bryant A. Julstrom. A weighted coding in a genetic algorithm for the degree-constrained minimum spanning tree problem. In *Proceedings of the 2000 ACM symposium on Applied computing - Volume 1*, SAC '00, pages 440–445, New York, NY, USA, 2000. ACM.

[RMR+93] R. Ravi, M. V. Marathe, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, III. Many birds with one stone: multi-objective approximation algorithms. In *Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, STOC '93, pages 438–447, New York, NY, USA, 1993. ACM.

[RS02] Celso C. Ribeiro and Maurício C. Souza. Variable neighborhood search for the degree-constrained minimum spanning tree problem. *Discrete Appl. Math.*, 118(1-2):43–54, April 2002.

[Sab61] Gert Sabidussi. Graph derivatives. *Mathematische Zeitschrift*, 76:385–401, 1961.

[SV85] Martin Savelsbergh and Ton Volgenant. Edge exchanges in the degree-constrained minimum spanning tree problem. *Computers amp; Operations Research*, 12(4):341 – 348, 1985.

[Vol89] A. Volgenant. A lagrangean approach to the degree-constrained minimum spanning tree problem. *European Journal of Operational Research*, 39(3):325 – 331, 1989.

[ZMC09] Fen Zhou, M. Molnar, and B. Cousin. Is light-tree structure optimal for multicast routing in sparse light splitting wdm networks? In *Computer Communications and Networks, 2009.*

ICCCN 2009. Proceedings of 18th Internatonal Conference on, pages 1 –7, aug. 2009.

[ZMC10] Fen Zhou, Miklós Molnár, and Bernard Cousin. Light-hierarchy: the optimal structure for multicast routing in wdm mesh networks. In *ISCC* [DBL10], pages 611–616.