Approximation of the Degree-Constrained Minimum Spanning Hierarchies

Miklós Molnár, Sylvain Durand, Massinissa Merabet

Approximation of the Degree-Constrained Minimum Spanning Hierarchies

Miklós Molnár, Sylvain Durand, and Massinissa Merabet

University Montpellier 2, Laboratoty LIRMM UMR 5506
CC477, 161 rue Ada
34095 Montpellier Cedex 5 - France
{miklos.molnar, sylvain.durand, massinissa.merabet@lirmm.fr

Abstract. Degree-bounded spanning problems are well known and are mainly used to solve capacity constrained communication (routing) problems. The degree-constrained spanning tree problems are NP-hard and the minimum cost spanning tree is not approximable nor in the case where the entire graph must be spanned neither in partial spanning problems. Most of applications (such as communications) do not need trees as solutions. Recently, a more flexible, connected, graph related structure called hierarchy was proposed to span a set of vertices in the case of constraints. The application of this structure permit the reformulation of the degree-constrained spanning problem. In this paper we present not only the advantages of the new structure but also that the new and optimal structure is approximable with the help of fast polynomial algorithm. The obtained approximation ratio is very advantageous regarding the negative approximability results with spanning trees.
Keywords: Graph theory, Degree-Constrained Spanning Problem, Spanning Hierarchy, Approximation

1 Introduction

To solve spanning problems in an efficient manner (generally minimizing the cost of the spanning structure) is important in several domains. For instance, connecting towns with a minimum cost wired structure, implementing a minimum cost telecommunication network or solving the routing in micro-circuits are classic examples for optimal spanning problems. Often, the spanning problem is given in a metrical space, and the optimal spanning structure minimize an arbitrary positive length function (a cost). Generally, the model corresponds to a valuated graph, where a given set of vertices should be spanned by the minimum cost structure. In the literature, sub-graphs are considered as solutions. The sub-graph which spans the set of vertices with minimum cost is a minimum spanning tree (MST), if there are no more constraint. Several polynomial time algorithms are known (cf. Prim's algorithm [1] or Kruskal's algorithm [2]) to solve the minimum spanning tree problem.

In some practical cases, different additional constraints are imposed. Various constrained spanning problems have been analyzed in graphs (cf. some examples in [3,4,5]). Here we are interested in the degree-bounded spanning problem. In this constrained spanning problem, each vertex $v \in V$ of an undirected graph $G = (V, E)$ is assigned a positive integer value $d(v)$ which represents the maximum degree of the vertex in any spanning structure (for example in the spanning trees). This degree is potentially different from the degree $d_G(v)$ of v in G. Note that only values $0 < d(v) \leq d_G(v)$ need to be considered for realistic cases. This degree bound can express the fact that a vertex can perform a given action (a branching) only for a limited member of neighbor vertices. In our paper we suppose that the degree bound is the same constant value, valid for all vertices in the graph.

The *degree-bounded spanning tree problem* has been formulated very early in [6] and has been extensively studied. For a long time, it is known that there is not always possible to span the vertices using trees with respect of the degree constraints (cf. [7]). Moreover, negative results are known on the approximability of the degree-bounded spanning tree problems [8].

For communications, the connectivity of the routes is inevitable but these routes can correspond to non-simple graph-related structures as walks, trails, etc. To span a set of vertices in a connected manner, a non-simple, tree-based structure has been proposed in [9]. This structure, called hierarchy, is obtained by a homomorphic mapping from a tree following the definition in [10]. It permit to respect the degree constraint for each vertex occurrence in the structure but a vertex (and an edge) can be concerned several times in the structure. That is: the structure is more flexible to respect the constraints than a spanning tree. In [11],it was demonstrated that

1. it is possible to span the vertices of a graph with respect of the degree bounds even if spanning trees satisfying the constraints do not exist
2. a spanning hierarchy with less cost can be found in some cases where spanning trees respecting the constraints exist.

In this paper we demonstrate that the optimal spanning hierarchy for the degree bounded spanning problem can be approximated. We propose a simple and efficient algorithm ensuring a good approximation of the optimum.

A possible application domain of the problem is the broadcast in all optical WDM networks where the splitting capacity of the vertices is limited (for example in[12]). To solve the optical routing problem, generally a set of trees (abusively called light forest) is proposed (for example in [13]) or a special walk (a light-trail) is computed as it is suggested in [14]. The spanning hierarchies give a good alternative to find efficient spanning structures (let us notice that a set of tree and a trail correspond also to hierarchies, cf. the definitions in [10]).

In the next of the paper, we propose a quick presentation of the well known and discussed degree-constrained spanning tree problem (cf. Section 2). After the related definitions in Section 3, Section 4 presents the algorithm which permits to compute a spanning hierarchy respecting a given degree bound with polynomial computation time and guaranteed approximation ratio. The presentation is closed by some perspectives ....

2 Related Works

The DCMST are firstly introduced and investigated in [6] (it is also briefly mentioned in [15]). They justified the fact that this problem is NP-hard by stating that solving the DCMST problem with B equal to two is amount to solve the Hamilonian path problem. Otherwise, By reducing the DCMST problem to an equivalent symmetric traveling salesman problem (TSP), Garey and Johnson [16] showed that this problem is NP-hard for any fixed constant $2 \leq B \leq |V - 1|$. Ravi showed that approximate the DCMST problem within a constant factor of the cost of an optimal tree is NP-Hard [17]. In unweighted graphs, Furer and Raghavachari [18] gave an elegant algorithm that returns a spanning tree in which the degree of each vertex is at most $B + 1$, or returns a witness certifying that the degree bounds are infeasible. Goemans proves in [19] that this result can be generalized to weighted graphs. In polynomial time, we can find a spanning tree of maximum degree at most $B + 1$ whose cost is no more than the cost of a minimum cost tree with maximum degree at most B. Note that this conjecture is formulated in the case where the degree bound is homogeneous. When there are different degree bounds on different vertices, Goemans proves that we can find, in polynomial time, a spanning tree of maximum degree at most $B+2$ whose cost is no more than the cost of a minimum cost tree with maximum degree at most B. The best result was presented by Singh and Lau in [20]; Their algorithm computes a spanning tree of minimum cost which violates degree upper-bounds by at most one. Since it is not possible to obtain any approximation algorithm, if we insist on satisfying all the degree upper bounds, this result is the best possible. The very first heuristic to solve the DCMST problem was proposed by Narula in [21]. The problem is formulated as a linear $0 - 1$ integer programming problem. A primal and a dual heuristic procedure and a branch-and-bound algorithm are proposed. The branch and bound algorithm of Savelsbergh and Volgenant [22] improved this solution by using more efficient branching rules and by implementing an edge exchange scheme. In [23], Gavish formulates the DCMST problem as a nonlinear combinatorial optimization problem, a Lagrangean relaxation of the problem is presented and effective solution procedures of the Lagrangean problem are developed. The Lagrangean solutions provide lower bounds on the optimal solutions to the DCMST problem. Other approaches are known: colony optimization [24] [25], evolutionary algorithms [26] [27], genetic algorithms [26], parallel algorithms [28], problem space search [26] and variable neighbourhood search [29].

3 Problem definition

In this section, the new problem formulation is given and its complexity is analyzed. Our objective is to find a cost minimum spanning structure without the hypothesis that this structure must be a sub-graph. Several applications require the configuration and the usage of a route, of a sequence of operations that can be followed by tokens, product and messages but the route may be different from a sub-graph. Let $G = (V, E)$ be an undirected graph with vertex set V, edge set E. We suppose that the logical scheme of the route, the succession of operations, etc. is given by a graph $F = (W, D)$. For instance, F can be a path if the route corresponds to a path or to a walk. The association between the logical route F and the physical topology G is given by a homomorphic mapping h.

Definition 1 (Token connectivity) *A token connected structure related to a graph G contains an (eventually non-simple) path between the vertices of any vertex pairs in G (cf. [10]).*

For instance, a *w*alk or a *t*raversal are token connected routes which may contain vertices and edges several times. So they are not always sub-graphs.

In our problem, the graph G is valuated and a strictly positive cost $c(e)$ is associated to every edge $e \in E$. Moreover, a positive integer B is given to bound the degree of vertices in the spanning structure. Trivially, in interesting cases $2 \leq B < \max_{v \in V} d(v)$.

Definition 2 (Degree constrained minimum spanning problem) *The problem consists in finding a* t*oken connected structure s.t. all vertices in V are covered. Supposing that the structure is given by the triplet $S = (F, h, G)$ from a graph $F = (W, D)$ applying a homomorphism h, the following conditions should be met:*

- *Each vertex $v \in V$ must be associated with at least one vertex $v' \in W$.*
- *The degree constraints must be respected in F: $d_F(v') \leq B, \forall v' \in W$.*
- *The cost $c(S)$ must be minimized.*

The token connectivity is important to span the graph G. The image of a not connected graph F in G can be connected but a route, a walk defined by the homomorphic association must be suitable for a token. So, the graph F must also be connected.

Definition 3 (Hierarchy) *If F is a connected graph without redundancy (i.e. a tee), $H = (F, h, G)$ is called a hierarchy.*

Figure 1 illustrates a hierarchy. Notice that the cost of a hierarchy is the some of the costs of passings by the edges: if an edge is used several times in G, its cost is summarized several times.

$$c(H) = \sum_{e' \in D} c(e)$$

where $e \in E$ is the edge associated with $e' \in D$.

The token connected, minimum cost structure is always a hierarchy even if there are constraints as it is proved in [10]. This statement is also true for our problem.

Lemma 1 *The minimum cost, token connected structure spanning the vertex set of G and respecting the degree constraints is a hierarchy.*

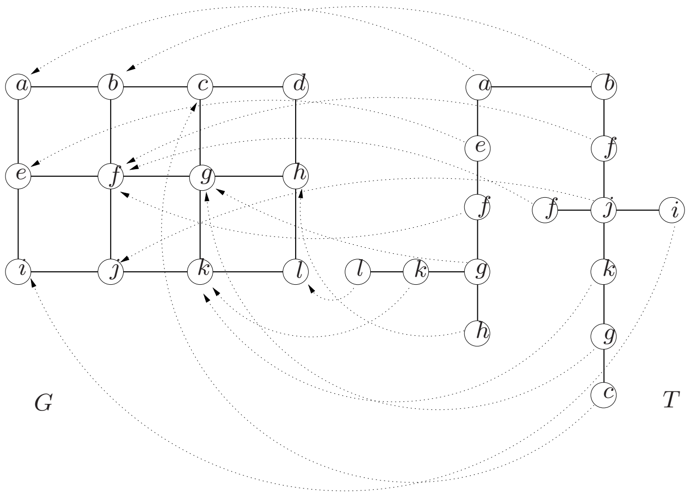

Fig. 1. Homomorphic mapping of vertices to define a hierarchy

Proof. The spanning structure given by the homomorphism (F, h, G) should be token-connected with respect to the degree constraint. That is, F can not contain any vertex with degree greater than B. Moreover it should be with minimum cost. It is sufficient to prove that the origin $F = (W, D)$ cannot contain any cycles: it is a tree.

Let us suppose that F contains a cycle. At most one edge of F can be omitted without less of the token-connectivity. Let d be an edge which can be deleted from D. The obtained graph $F' = (W, D \setminus \{d\})$ is connected and covers V (the set of vertices associated to W does not change). The structure (H', x, G) is a token-connected spanning solution, it respects the degree constraint and its cost is less then the cost of F (the deleted edge have a positive cost). So, F (which contains a cycle) cannot be of minimum cost. $\square$

Let us suppose that the optimal solution of the degree constrained minimum cost spanning problem is obtained by the triplet (T, h, G) and is a Degree Constrained Minimum Spanning Hierarchy, abbreviated by DCMSH in the following. This hierarchy always exists in connected graphs applying a homogeneous degree bound $2 \leq B < \max_{v \in V} d(v)$. The problem of the minimum spanning hierarchy is NP-hard as it is demonstrated in the following.

3.1 NP-Hardness of DCMSH

Lemma 1. *If among all the Minimum Spanning Trees (MST) of a graph G, there exists one satisfying the degree constraint, it is an optimal solution for the DCMSH problem and all the optimal solutions are trees in G.*

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

Moreover, we prove that, in this case, all the optimal solutions are trees. Suppose that an optimal hierarchy $H = (T, h, G)$ exists and it is not a tree in G. Because the MST T^* is an optimal solution of our problem, the cost $c(H)$ of the optimal hierarchy must be the same that the cost $c(T^*)$ of the MST solution. Trivially, the cost of a hierarchy is lower bounded by the cost of its image in G: $c(I) \leq c(H)$, where I is the image (the sub-graph generated by H in G). Then, it contains at least a cycle in G (a duplicated edge is considered as a cycle). I covers the vertex set V. Two possibilities can arise.

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

■

Remark 1: The result is not true if we consider the $s$panning trees satisfying the degree constraint. In the graph of Figure 2), for $B = 2$, the shortest Hamiltonian path the DCMST with $B = 2$ have a length of 12 while the shortest walk (the shortest hierarchy) for the same constraint, indicated by dotted line have a length of 4 (b, a, d, a, c).

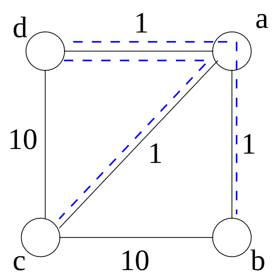

Fig. 2. The DCMSH is shorter than the minimum Hamiltonian path

Remark 2: The cost of the minimum spanning tree is therefore a lower bound for the DCMSH problem.

Lemma 2. *The problem of DCMSH is NP-Hard for all $B \geq 2$.*

Proof. Let $G = (V, E)$ be a graph with $c(e) = 1, \forall e \in E$. Let $G' = (V', E')$ be the graph obtained by adding $B - 2$ leaves connected by edges of cost 1 to each vertex of V. In G', $|V'| = |V| + |V|(B - 2) = (B - 1)|V|$. Any spanning tree of G' has a cost equal to $(B-1)|V|-1$. There is a DCMSH of cost $(B-1)|V|-1$ in G' if and only if there is a Hamiltonian path in G. (Remember, that the Hamiltonian path contains $|V| - 1$ edges.)

Suppose that there is a DCMSH $H = (T, h, G')$ of cost $(B-1)|V|-1$ in G'. Regarding its cost, H is a tree of G'. If we remove all the $(B-2)|V|$ vertices of $V' \setminus V$ from H, we obtain a connected subgraph in which all vertices have a degree lower or equal to two, which is a Hamiltonian path of G.

Reciprocally, adding $B - 2$ leaves to each vertices of a Hamiltonian path of G gives a tree satisfying the degree constraint which is an optimal DCMSH in G' because of the previous lemma. ∎

Since the problem is NP-hard, guaranteed approximation algorithms are interesting to solve it in practical cases.

4 Degree Constrained Span of Graph

At first, we propose the analysis of the intensively studied minimum Hamiltonian path and walk problems known also as the Traveler Salesman Path (TSP) and Traveler Salesman Walk (TSW) problems [30] [31].

4.1 Minimum Hamiltonian walk: case of $B = 2$

When $B = 2$, branching vertices are not authorized in the connected spanning structure. If cycles are also excluded, the minimum spanning structure, *i.e.* the minimum spanning tree respecting the degree constraint is the minimum Hamiltonian path (if it exists). It is known that in some graphs, there is no Hamiltonian path and that it is NP-Complete to decide. When cycles are authorized, but the spanning hierarchy can not have branching vertex occurrences, the solution is a walk. Following the term in [31], we talk about a "Hamiltonian walk", when the walk covers at least once all vertices in the graph. In connected graphs, a Hamiltonian walk and consequently a minimum Hamiltonian walk always exists. Nevertheless, we will show that the minimum Hamiltonian walk problem is NP-hard and it is in APX.

An equivalent problem: the minimum Hamiltonian path in the metrical closure

Lemma 3. *A connected graph $G = (V, E)$ always admits a minimum Hamiltonian walk.*

Proof. Any traversal of the graph is a Hamiltonian walk. Since the number of traversal is finite, there exists one with minimum length. ■

To find the minimum Hamiltonian walk of the graph, first we propose a decomposition of this walk.

Lemma 4. *A minimum Hamiltonian walk HW can be decomposed into a set of successive, connected shortest paths.*

Proof. Since the Hamiltonian walk HW covers the vertex set V, there is at least one occurrence of each vertex $v_i \in V$ in HW. Let v_i^1 be the first occurrence of v_i in a traversal HW. Let v_j^1 be the following first occurrence of a vertex ($v_i \neq v_j$). We show that the walk $W = (v_i^1, ..., v_j^1)$ of HW corresponds to a shortest path between v_i and v_j in G.

Suppose that a walk $W = (v_i^1, ..., v_j^1)$ does not correspond to a shortest path. Then there exist a shorter path $W' = (v_i^1, v_j)$ between these extremities. Replacing $W = (v_i^1, ..., v_j^1)$ by $W' = (v_i^1, v_j)$ in HW, we obtain a walk HW' covering the vertex set with a lower cost. Consequently, HW can not be a minimum Hamiltonian walk.

The last vertex of HW is a first (and so unique) occurrence of the vertex (otherwise, HW is not minimum). This complete the proof. ■

Theorem 1. *A minimum Hamiltonian walk HW of a graph G corresponds to a minimum Hamiltonian path HW' in the metrical closure of G.*

Proof. According to Lemma 4, a minimum Hamiltonian walk HW can be decomposed into a sequence $S = (P_1, P_2, P_{n-1})$ of shortest paths in G. The last vertex of P_i is the first of P_{i+1} for all $i < n-1$. The n extremities of these paths are the first occurrences of the n vertices of G in HW. Each shortest path P_i corresponds to an edge e_i in G'. Because of the previous properties, the set of edges corresponding to these paths is a Hamiltonian path HP in G'. The length of HP is equal to the length of HW. Suppose that HP is not a minimum Hamiltonian path in G'. Let HP' be a minimum Hamiltonian path in G'. This path corresponds to a walk HW' in G (each edge of HP' corresponds to a shortest path in G). Moreover, the length of HW' is equal to the length of HP', which contradicts the optimality of HW.

The reciprocity is obvious using the same kind of arguments.

■

The theorem gives an interesting equivalence between the minimum Hamiltonian walk problem and the minimum Hamiltonian path problem in the metrical closure of the original graph. The image of a minimum Hamiltonian walk is always a minimum Hamiltonian path in the metrical closure and vice versa.

Remember, that the minimum (shortest) Hamiltonian path problem is NP-hard in general [32]. This problem can easily be transformed into a TSP [33].

The minimum Hamiltonian path through all vertices and independently from the endpoints can be found in a slightly modified graph: by inserting a dummy vertex which is connected with zero cost edges to all other vertices. The solution of the TSP (the minimum cost tour) in this graph gives the minimum Hamiltonian path in the original graph after deletion of the dummy vertex.

The minimum Hamiltonian path problem can always be solved in the metrical closure (the closure is a complete graph and several Hamiltonian paths exist in it). Moreover, the minimum Hamiltonian walk problem can be solved using the approximation algorithms known for the minimum Hamiltonian path in metrical graphs. Thus, the problem is APX-Complete and can be approximated [34]. Following the results of Hoogeveen, a 3/2-approximation algorithm can be applied if the end-points of the requested path are not specified [35].

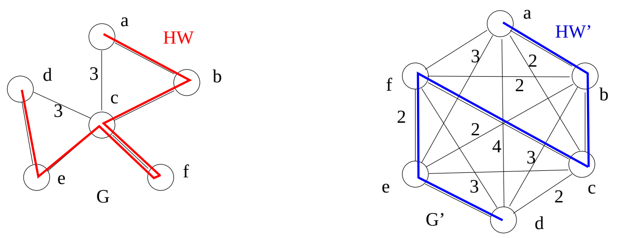

Fig. 3. A minimum Hamiltonian walk corresponds to a minimum Hamiltonian path in the metrical closure

We are interested by the solution of the DCMSH problem with approximation guarantee for arbitrarily value of B. At first, we analyze the case of particular star graphs.

4.2 Span a star with hierarchies

A star $S = (V, E)$ can easily be covered by spanning hierarchies with respect of a degree constraint B. In a star, the only one vertex with degree greater than two is the central vertex c which is an articulation vertex. The minimum spanning hierarchy respecting the degree constraint $B \geq deg(c)$ is a hierarchy which may contain several times the central vertex s.t. each occurrence of c respects the degree constraint. Some leaves may also be duplicated with respect of the same degree constraint. Intuitively, in an optimal solution, the less cost edges of the star are duplicated to return to c. The computation of the DCMSH in the star is equivalent to the minimization of the length of the duplicated edges.

Minimum Hamiltonian Walk At first, let us rapidly summarize our result on the minimum Hamiltonian walk computation in a star (case of $B = 2$).

Lemma 5. *The minimum Hamiltonian walk in a star graph can be computed in $O(|E| \cdot \log |E|)$ time.*

Proof. To span a star supposing $2 < deg(c)$, the Hamiltonian walk must return to c several times and some edges are duplicated in the walk. The number of edges which may be not duplicated is two (the edges leaving to the extremities of the walk). Trivially, the minimum cost walk can be obtained if the two, not duplicated edges are the cheapest ones in the star. A simple sort of edges is sufficient to select the edges duplicated in the Hamiltonian walk. The best cost of the sort is $O(|E| \cdot \log |E|)$ and the selection of the duplicated edges is in $O(|E|)$ from where the complexity. ■

In the case of $B > 2$, the coverage of a star with respect of the degree constraint is more complicated. The analyze of the problem and its exact solutions are important challenges. Without this analysis and to design an efficient approximation algorithm to span arbitrary connected graphs, we propose a simple (but not optimal) spanning hierarchy computation to span stars. The following heuristic guarantees an interesting approximation ratio of the cost.

Heuristic to span a star Recall, each vertex occurrence in the hierarchy H_S spanning a star must respect the degree constraint B. Let us suppose that the number of leaves is sufficiently high and returns to the central vertex are needed to cover all the vertices. To return to the central vertex, the less cost edges are used in an increasing order of edge costs. In this heuristic, selected edges are duplicated at most once. Formally, let us make a partition of the edges of the star as follows. There are $\lfloor deg(c)/(B-1) \rfloor + 1$ sets in the partition. Each set, except the last, contains $B-1$ edges. (If $deg(c) \bmod (B-1) = 0$, the last edge set is empty.) The $\lfloor deg(c)/(B-1) \rfloor$ less cost edges are distributed in the partition: each of them is in a separated set (if the last edge set is empty, there is no less cost edge in this set, trivially). In the spanning hierarchy, the central vertex is duplicated several times. Each edge set is connected to a different occurrence of the central vertex forming a star with at most $B-1$ edges (leaves). The less cost edge of each set is duplicated to return to the next central vertex occurrence. So, each central vertex occurrence respects the degree constraint B and the obtained graph is connected. Figure 4 illustrates two hierarchies for $B = 4$: one with $deg(c)/(B-1) \not\equiv 0$ and one with $deg(c)/(B-1) \equiv 0$.

The algorithm is formally described by Algorithm 6 in Annex.

Lemma 6. *The spanning hierarchy H_S computed by Algorithm 6 contains $N_c = \lfloor deg(c)/(B-1) \rfloor + 1$ times the central vertex. Each occurrence of c respects the degree constraint. If $N_c \geq 2$, the first and the last occurrences have a degree strictly lower than B. The cost ratio $r = c(H_S)/c(S)$ is bounded by $B/(B-1)$.*

Proof. By construction, each occurrence of c in H_S have a degree at most B. In each set of the partition, the number of exclusively spanned leaves is at most

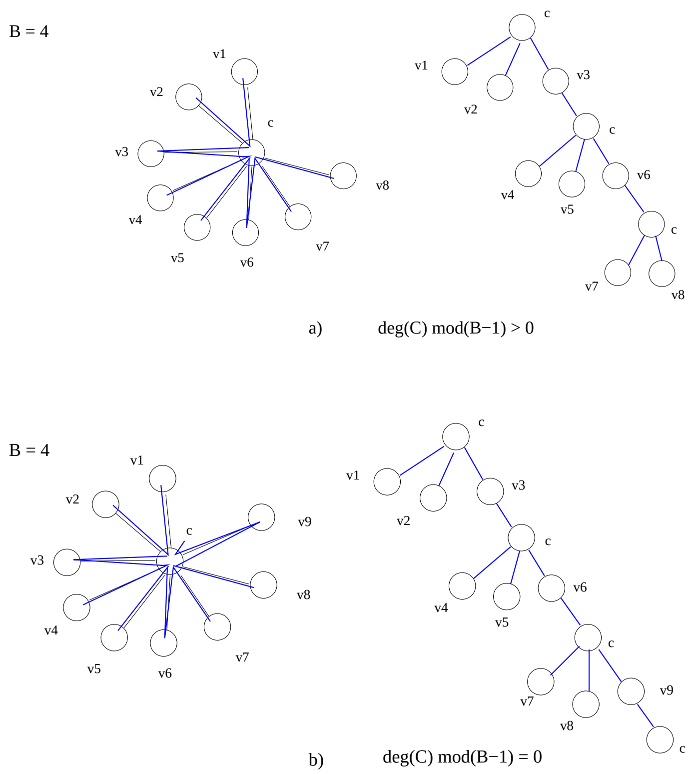

Fig. 4. Spanning hierarchies of stars computed by the proposed heuristic

equal to $B-1$. It is $B-1$ for all occurrences of c except eventually one (the last occurrence has not obligatory $(B-1)$ adjacent vertices). There are at most $\lfloor deg(c)/(B-1) \rfloor$ duplicated (return) edges. By choosing the less cost edges to duplicate, the cost of the duplicated part $c(D)$ of the star is limited by

$$c(D) \leq \frac{\lfloor deg(c)/(B-1) \rfloor}{deg(c)} c(S) \leq \frac{deg(c)/(B-1)}{deg(c)} c(S) = \frac{1}{B-1} c(S)$$

An upper bound of the cost ratio is given by:

$$r = \frac{c(H_S)}{c(S)} = \frac{c(S)+c(D)}{c(S)} \leq \frac{B}{B-1}$$

■

Remark: If $N_c = 1$ (case of $deg(c) < B-1$), the central vertex has a degree strictly lower than $B-1$.

This spanning hierarchy corresponds to a "chain" of stars, each star having a central node with degree at most B. Moreover, it ensure that the central vertex occurrences in the stars at the "extremities" of the chain have a degree less than B (if there is only one star, $deg(c) < B-1$, cf. Remark).

The following section presents how to compute an approximated solution of the DCMSH problem in an arbitrary, connected graph.

5 An approximation algorithm of the DCMSH

The first idea is that upper bounds for approximation algorithm can be computed regarding the MST instead of the optimal (often not known) spanning hierarchy. In the following, we propose an approximation algorithm and we give an upper bound of the approximation ratio. The algorithm is based on a decomposition of the MST in the graph.

5.1 A star decomposition of the MST

An arbitrary tree, and in this manner the MST, can be decomposed into a set of stars in the following way. Let $T = (V_T, E_T)$ be a tree with $|V_T| > 2$ and v_1 an arbitrary vertex in T (the case where $|V_T| \leq 2$ is trivial for spanning). Then v_1 can be considered as the central vertex of a star S_1. Some neighbor vertices of v_1 in S_1 are leaves in T some others may be branching vertices. The branching vertices can be considered as central vertices of following stars. Recursively, the entire tree can be covered by stars which are edge disjoint. The decomposition is illustrated by Figure 5.

Since the stars are edge disjoint and cover all edges of T, trivially:

$$c(T) = \sum_{i=1}^{k} c(S_i)$$

where $S_i, i = 1, \ldots, k$ indicate the stars in the decomposition.

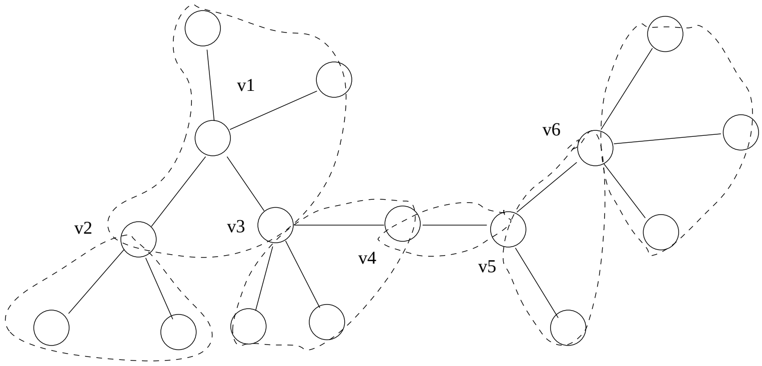

Fig. 5. A star decomposition of a tree

5.2 The proposed heuristic to approximate the DCMSH

To compute an approximation of the DCMSH in a given graph, we propose the following algorithm.

1. Compute an MST of the graph.
2. Decompose this MST using stars S_1, S_2, ..., S_k (as proposed here before in the paper)
3. For each star S_i, compute a spanning hierarchy H_i as proposed in the previous sub-section.
4. "Re-connect" the spanning sub-hierarchies to form a connected spanning hierarchy. The step is illustrated by Figure 6. A connection is needed, if a leaf in a star coincides with the central vertex of another one. For example, between two neighbor sub-hierarchies spanning stars S_i and S_j, a leaf of S_i corresponds to the central vertex in S_j (case of vertices v_2 and v_3 in the figure). In H_i, the leaves of S_i are not duplicated and have a degree 1 or 2. Let us indicate by l_i a leaf in S_i which correspond to the central vertex c_j of S_j corresponding to a vertex v_k in the original graph. c_j can be repeated in H_j but its first occurrence has a degree $B-1$.
   (a) If l_i has a degree 1 in H_i, it can aggregated with the first occurrence of c_j in H_j and only one vertex can represent this vertex in the final hierarchy (the union of these occurrences of the same vertex v_k respects the degree constraint B). This is the case of vertex v_3 in our figure.
   (b) If l_i has a degree 2 in H_i, the connection can be made as follows. If the corresponding central vertex c_j has only one occurrence in H_j, than this occurrence is of degree strictly less than $B-1$. Consequently, l_i and $c-j$ can be aggregated in the final hierarchy and the aggregated vertex respects the degree constraint (cf. vertex v_2 in the figure). If there are several occurrences of c_j in H_j, the first and the last occurrences have a degree at most B_1 and the two adjacent edges of l_i can be attached to these two occurrences (with other world: l_i can be duplicated and each

occurrence of l_i can be aggregated by one occurrence of c_j with degree less than B in H_j).

5. The connected hierarchy can contain useless return edges. An edge returning to a central vertex occurrence of a star s.t. the degree of this occurrence is equal to one is useless (this occurrence is a leaf). The useless edges must be deleted.

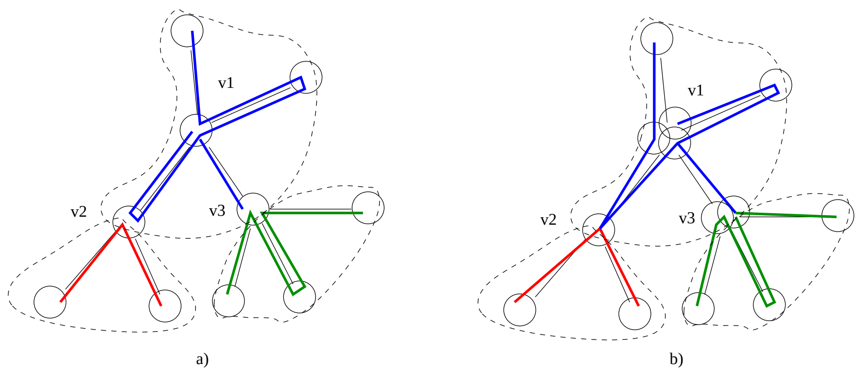

Fig. 6. Connection of sub-hierarchies spanning the stars

A more formal description of the algorithm is given by Algorithm 6 in Annex.

Lemma 7. *Algorithm 6 offers an $R \leq \frac{B}{B-1}$ approximation of the optimal solution.*

Proof. The algorithm is based on a decomposition of the MST T^* into a set of edge disjoint stars. Let $c(S_i)$ be the cost of the star $S_i, i = 1, \ldots, k$ in the decomposition.

$$c(T^*) = \sum_{i=1}^{k} c(S_i)$$

The obtained spanning hierarchy length is equal to

$$c(H) = \sum_{i=1}^{k} c(H_{S i})$$

Using the result of Lemma 6:

$$c(H) \leq \sum_{i=1}^{k} \frac{B}{B-1} c(S_i)$$

$$l(H) \leq \frac{B}{B-1} c(T^*)$$

The approximation ratio is immediately

$$R = \frac{l(H)}{l(H^*)} \leq \frac{l(H)}{l(T^*)} \leq \frac{B}{B-1}$$

■

Remark 1: If $deg(c) < B$ for all vertices $c \in V_{T^*}$, then the solution corresponds to an MST, which is the optimum in this case.

Remark 2: If $B = 2$, the algorithm performs a deep-first search type traversal in the MST.

Finally, the reader can find here an important remark discussing the fact, that the heuristic is not directly related to the optimal spanning hierarchy but to the MST of the graph.

Discussion about the heuristic Since the proposed heuristic only uses the edges of an MST, the resulting hierarchy may be of poor quality. In our discussion, we use the following terms. Since a hierarchy H in a graph G is given by a triplet (T, h, G), and T is a tree, we talk about a leaf of the hierarchy when the concerned vertex is a leaf in T. Similarly, we talk about internal vertices concerning the non leaf vertices in T.

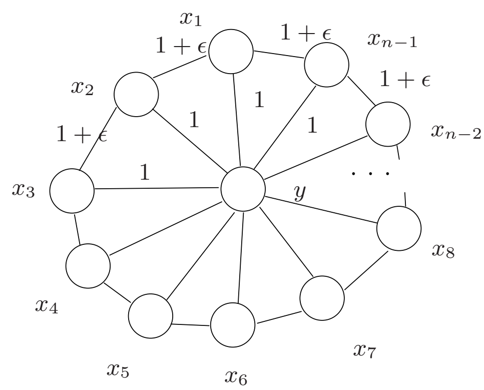

Fig. 7. A wheel graph used in Theorem 2

Theorem 2. *No constant approximation ratio lower than $B/(B-1)$ can be achieved for any heuristic only based on an MST.*

Proof. Let $G = (V, E)$ be a graph with $V = (y, x_1, x_2, ..., x_{n-1})$ and $E = \{(y, x_1), (y, x_2), ..., (y, x_{n-1}), (x_1, x_2), (x_2, x_3), ..., (x_{n-1}, x_1)\}$ (see Figure 7). Suppose that $c(y, x_i) = 1 \quad i = 1, ..., n-1$ and $c(x_i, x_{i+1}) = 1 + \epsilon \quad i = 1, ..., n-2$ and also $c(x_{n-1}, x_1) = 1 + \epsilon$.

Trivially, the path $P = (y, x_1, x_2, x_3, ..., x_{n-2}, x_{n-1})$ is a spanning hierarchy of G, which respects the degree constraint for any $B > 1$ and with a cost $c(P) = 1 + (n-2)(1+\epsilon)$.

The Minimum Spanning Tree of G is the star S of center y with $n-1$ leaves. Let $H^* = (T^*, h^*, S)$ be an optimal hierarchy spanning the star S and respecting the degree constraint. In H^*, every vertex corresponding to a leaf of S is present only once. This is true for the leaves of H^*, which can be removed if they are spanned somewhere else in H^*. This is also true for internal nodes of H^*: if an internal node is duplicated, its label can be replaced by the label of a leaf of H^*, and this leaf can then be removed. So T^* is a particular bipartite graph where the partition of the vertices can be made as follows: one vertex set with the n_y occurrences of y and the other with the $n-1$ vertices corresponding to the leaves of S. The number of edges in H^* is equal to the number of edges in T^*. Since T^* is a tree, its number of edges is equal to its number of vertices minus 1. Consequently, $c(H^*) = (n-1) + n_y$. Any occurrence of y have at most B neighbors in T^*. So the number of edges of H^* is at most $n_y * B$ and we have $(n-1) + n_y \le n_y * B$ which implies $\frac{n-1}{B-1} \le n_y$. The cost of H^* is then at least $c(H^*) \ge (n-1) + \frac{n-1}{B-1} = \frac{B(n-1)}{B-1}$.

Hence, the approximation ratio of any heuristic only based on an MST is greater or equal to $\frac{c(H^*)}{c(P)} = \frac{\frac{B(n-1)}{B-1}}{1+(n-2)(1+\epsilon)}$ and $\frac{(n-1)}{1+(n-2)(1+\epsilon)}$ can be as close to 1 as wanted for n large enough and ϵ small enough. ∎

6 Conclusion

In this paper, we consider the problem of finding a minimum cost degree constrained spanning structure. It has already been proved that the optimal structure is a hierarchy. In the special case where the degree is bounded by 2, the problem is equivalent to find a minimum Hamiltonian path in the metric closure. In this case, the known approximation algorithms to solve the minimum Hamiltonian path problem in metric graphs can be applied. In the general case, we provide the first approximation algorithm to compute the degree constrained minimum spanning hierarchy. The ratio obtained shows that this problem is in APX. More the degree bound is high, more interesting is the ratio guaranteed by the algorithm. Future work will consist in an improvement of the ratio and showing that the problem is APX-Complete (or to find a PTAS).

References

1. Prim, R.C.: Shortest connection networks and some generalizations. Bell System Technology Journal **36** (1957) 1389–1401

2. Kruskal, J.B.: On the Shortest Spanning Subtree of a Graph and the Traveling Salesman Problem. Proceedings of the American Mathematical Society **7** (1956) 48–50
3. Papadimitriou, C.H., Yannakakis, M.: The Complexity of Restricted Minimum Spanning Tree Problems (Extended Abstract). In Maurer, H.A., ed.: ICALP. Volume 71 of Lecture Notes in Computer Science., Springer (1979) 460–470
4. Cieslik, D.: The vertex degrees of minimum spanning trees. European Journal of Operational Research **125** (2000) 278–282
5. Ruzika, S., Hamacher, H.W.: A Survey on Multiple Objective Minimum Spanning Tree Problems. In Lerner, J., Wagner, D., Zweig, K., eds.: Algorithmics of Large and Complex Networks: Design, Analysis, and Simulation, LNCS 5515. Springer-Verlag, Berlin, Heidelberg (2009) 104–116
6. Deo, N., Hakimi, S.: The shortest generalized Hamiltonian tree. In: Sixth Annual Allerton Conference. (1968) 879–888
7. Bauer, F., Varma, A.: Degree-constrained multicasting in point-to-point networks. In: INFOCOM '95: Proceedings of the Fourteenth Annual Joint Conference of the IEEE Computer and Communication Societies (Vol. 1)-Volume, Washington, DC, USA, IEEE Computer Society (1995) 369
8. 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 **31** (2001) 58–78
9. Molnár, M.: Optimisation des communications multicast sous contraintes. Mémoire of habilitation (in French), University Rennes 1 (2008)
10. Molnár, M.: Hierarchies to Solve Constrained Connected Spanning Problems. Technical Report 11029, LIRMM (2011)
11. Merabet, M., Durand, S., Molnar, M.: Exact solution for connected spanning problems with degree constraint. **submitted** (2013)
12. Zhou, Y., Poo, G.S.: Optical multicast over wavelength-routed wdm networks: A survey. Optical Switching and Networking **2** (2005) 176–197
13. Zhang, X., Wei, J., Qiao, C.: Constrained multicast routing in WDM networks with sparse light splitting. In: IEEE INFOCOM. Volume 3. (2000) 1781–1790
14. Ali, M., Deogun, J.: Cost-effective implementation of multicasting in wavelength-routed networks. IEEE J. Lightwave Technol., Special Issue on Optical Networks **18** (2000) 1628–1638
15. Obruca, A.K.: Spanning tree manipulation and the travelling salesman problem. The Computer Journal **10** (1968) 374–377
16. 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)
17. Ravi, R., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Hunt, III, H.B.: Many birds with one stone: multi-objective approximation algorithms. In: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing. STOC '93, New York, NY, USA, ACM (1993) 438–447
18. Fürer, M., Raghavachari, B.: Approximating the minimum degree spanning tree to within one from the optimal degree. In: Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms. SODA '92, Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (1992) 317–324
19. Goemans, M.: Minimum bounded degree spanning trees. In: Foundations of Computer Science, 2006. FOCS '06. 47th Annual IEEE Symposium on. (2006) 273–282
20. Singh, M., Lau, L.C.: Approximating minimum bounded degree spanning trees to within one of optimal. In: STOC '07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing, New York, NY, USA, ACM (2007) 661–670

21. Narula, S.C., Ho, C.A.: Degree-constrained minimum spanning tree. Computers and Operations Research **7** (1980) 239 – 249
22. Savelsbergh, M.W.P., Volgenant, T.: Edge exchanges in the degree-constrained minimum spanning tree problem. Computers and OR **12** (1985) 341–348
23. Gavish, B.: Topological design of centralized computer networksformulations and algorithms. Networks **12** (1982) 355–377
24. Bau, Y., Ho, C., Ewe, H.: An ant colony optimization approach to the degree-constrained minimum spanning tree problem. In: Computational Intelligence and Security. Volume 3801 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg (2005) 657–662
25. Bui, T.N., Zrncic, C.M.: 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, New York, NY, USA, ACM (2006) 11–18
26. Krishnamoorthy, M., Ernst, A.T., Sharaiha, Y.M.: Comparison of algorithms for the degree constrained minimum spanning tree. Journal of Heuristics **7** (2001) 587–611
27. Knowles, J., Corne, D.: A new evolutionary approach to the degree-constrained minimum spanning tree problem. Evolutionary Computation, IEEE Transactions on **4** (2000) 125 –134
28. Mao, L.J., Deo, N., Lang, S.D.: 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. (1999) 184 –189
29. Ribeiro, C.C., Souza, M.C.: Variable neighborhood search for the degree-constrained minimum spanning tree problem. Discrete Appl. Math. **118** (2002) 43–54
30. Gutin, G., Punnen, A.: The Traveling Salesman Problem and Its Variations. Combinatorial Optimization. Springer (2002)
31. Lam, F., Newman, A.: Traveling salesman path problems. Math. Program. **113** (2008) 39–59
32. Papadimitriou, C.H.: Computational complexity. Addison-Wesley (1994)
33. Garfinkel, R.: Motivation and Modeling. In Lawler, E.L., Lenstra, J.K., Kan, A.H.G.R., Shmoys, D.B., eds.: The Traveling Salesman Problem. John Wiley and Sons (1985) 17–36
34. Christofides, N.: Worst-case analysis of a new heuristic for the travelling salesman problem. Technical Report 388, Graduate School of Industrial Administration, Carnegie Mellon University (1976)
35. Hoogeveen, J.A.: Analysis of Christofides' heuristic: Some paths are more difficult than cycles. Oper. Res. Lett. **10** (1991) 291–295

Annex

Heuristic to compute a degree constrained hierarchy spanning a star

The hierarchy $H_S = (T_S, h, S)$ spanning S is represented as a labeled tree $T_S = (V_{TS}, E_{TS})$ where the labels of vertices $v \in V_{TS}$ are vertices of S corresponding to the mapping h. In the description, the central vertex is c, the function opposite(e, v) returns the extremity of edge e opposite to the extremity v.

Algorithm 1 Heuristic to span a star with respect of a degree bound

Require: a weighted star graph $S = (V_s, E_s)$, a cost $c(e)$ for each edge $e \in E_s$ and a degree constraint B
Ensure: T_S a labeled tree representing the hierarchy spanning S with respect of the degree constraint B

Sort E_s in increasing order of costs
$I \leftarrow \lfloor deg(c)/(B-1) \rfloor + 1$ { the number of degree limited stars}
$M \leftarrow deg(c) \bmod (B-1)$
{initialize degree constrained stars}
$i \leftarrow 1$
while $i < I$ **do**
  $V_{ST[i]} \leftarrow \{c^i\}$ { one occurrence of c}
  $E_{ST[i]} \leftarrow \{E_s[i]\}$ { one of the less cost edges}
  $V_{ST[i]} \leftarrow V_{ST[i]} \bigcup \{\text{opposite}(E_s[i], c)\}$ { its extremity}
  $i \leftarrow i + 1$
end while
{ initialize the edge set of the last star}
if $M \neq 0$ **then**
  $E_{ST[i]} \leftarrow \{E_s[i]\}$
else
  $E_{ST[i]} \leftarrow \emptyset$
end if
{dispatch remained edges into constrained stars}
$k \leftarrow 1$
$m \leftarrow 1$
$i \leftarrow I + 1$
while $i < I$ **do**
  $E_{ST[k]} \leftarrow E_{ST[k]} \bigcup \{E_s[i]\}$
  $V_{ST[k]} \leftarrow V_{ST[k]} \bigcup \{\text{opposite}(E_s[i], c)\}$
  $m \leftarrow m + 1$
  if $m = B - 2$ **then**
    $m \leftarrow 1$
    $k \leftarrow k + 1$
  end if
  $i \leftarrow i + 1$
end while

{the result is the union of stars connected by redundant (less cost) edges}

$V_{TS} \leftarrow \bigcup_{i=1}^{I} V_{ST[i]}$

$E_{TS} \leftarrow \bigcup_{i=1}^{I} E_{ST[i]}$

$i \leftarrow 1$

while $i < I$ **do**

  $E_{TS} \leftarrow \{E_s[i]\}$ { duplicated, less cost edges}

  $i \leftarrow i + 1$

end while

return T_S

The proposed approximation algorithm

Algorithm 2 The proposed approximation algorithm of the DCMSH

Require: a weighted graph $G = (V, E)$, with costs $c(e)$ for each edge $e \in E_s$ and a degree constraint B
Ensure: T_G a labeled tree representing the hierarchy spanning G with respect of the degree constraint B

```
Compute T∗ = (V_T∗, E_T∗) the MST of G        {Prim's algorithm}
      {decomposition into stars}
i ← 1
Select an arbitrary vertex v_1 ∈ V
C ← {v_1}
while C ≠ ∅ do
  Select an arbitrary vertex c ∈ C
        {construct a star _i = (VS_i, ES_i)}
  VS_i ← {c}
  ES_i ← ∅
  for all e ∈ adjacent_edge (c, T∗) do
    ES_i ← ES_i ⋃{e}
    v ← opposite(e, c)
    VS_i ← VS_i ⋃{v}
    VS_i ← {c}
    if not_leaf(v, T∗) then
      C ← C ⋃{v}
    end if
  end for
  TS_i ← Algorithm1(S_i, B)
  i ← i + 1
end while
      { Connect the spanning hierarchies }
TO WRITE...
for all star S_j, j ≠ 1 do
  select the central vertex c_j
  l_i ← same_label(c_j)        { leaf with the same label in H_i }
  if degree(l_i, H_i) = 2 and number_of_occurrences(c_j, H_j) > 1 then
    create two occurrences l_i^1 and l_i^2 of l_i with one adjacent edge for each
    aggregate l_i^1 and c_j^1
    aggregate l_i^2 and c_j^last
  else
    aggregate l_i and c_j^1
  end if
end for
delete useless return edges
return T_S
```
