

Approximation of the Degree-Constrained Minimum Spanning Hierarchies

Miklós Molnár, Sylvain Durand, Massinissa Merabet

► **To cite this version:**

Miklós Molnár, Sylvain Durand, Massinissa Merabet. Approximation of the Degree-Constrained Minimum Spanning Hierarchies. 21th International Colloquium on Structural Information and Communication Complexity (SIROCCO 2014), Jul 2014, Takayama, Japan. pp.96-107, 10.1007/978-3-319-09620-9_9. lirmm-01056259

HAL Id: lirmm-01056259

<https://hal-lirmm.ccsd.cnrs.fr/lirmm-01056259>

Submitted on 16 Apr 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Approximation of the Degree-Constrained Minimum Spanning Hierarchies

Miklós Molnár, Sylvain Durand, and Massinissa Merabet

University Montpellier 2, Laboratory LIRMM UMR 5506

CC477, 161 rue Ada

34095 Montpellier Cedex 5 - France

{miklos.molnar, sylvain.durand, massinissa.merabet@lirmm.fr

Abstract. Degree-constrained spanning problems are well known and are mainly used to solve capacity constrained routing problems. The degree-constrained spanning tree problems are NP-hard and computing the minimum cost spanning tree is not approximable. Often, applications (such as some degree-constrained 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 under constraints. This structure permits a new formulation of some degree-constrained spanning problems. In this paper we show that although the newly formulated problem is still NP-hard, it is approximable with a constant ratio. In the worst case, this ratio is bounded by $3/2$. We provide a simple heuristic and prove its approximation ratio is the best possible for any algorithm based on a minimum spanning tree.

Keywords: Graph theory, Networks, Degree-Constrained Spanning Problem, Spanning Hierarchy, Approximation

1 Introduction

Solving spanning problems in a cost efficient manner is important in several domains. For instance, implementing a minimum cost communication network or solving the routing in micro-circuits are classic examples for optimal spanning problems. Often in graphs, a given set of vertices should be spanned by the minimum cost structure. In the literature, solutions are mainly considered to be sub-graphs. For example, the structure which spans all the vertices in a graph with minimum cost is a minimum spanning tree (MST).

In some practical cases, different additional constraints are imposed. Various constrained spanning problems have been analyzed in graphs (cf. some examples in [1,2,3]). Here we are interested in the degree-constrained spanning problem. In this constrained spanning problem, a positive integer value $d(v)$ is assigned to each vertex $v \in V$ of an undirected graph $G = (V, E)$. This value represents the maximum degree of the vertex in the spanning structure (usually in a tree). 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 two different facts:

1. the vertex has a global "budget" to connect neighbor vertices (this budget approach can be found in [4])
2. because of its limited instantaneous "capacity", the vertex can perform a given action (a branching) for each of its visit only for a limited number of neighbor vertices.

The first case corresponds to the *degree-constrained spanning tree problem*. It has been formulated in [5] and has been extensively studied. For a long time, it is known that it is not always possible to span the vertices using trees with respect of the degree constraints. Moreover, negative results are also known on the approximability of the degree-constrained spanning tree problems [4].

In our paper we suppose that the degree bound expresses the limited capacity of the vertex for each visit (case 2). Moreover, we suppose that the limit is the same constant value, valid for all vertices in the graph.

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 [6]. This structure, called hierarchy, is obtained by a homomorphic mapping of vertices between a tree and an arbitrary graph (cf. Section 3). A new formulation of degree-constrained problems is possible and profitable for some applications if the constraints concern each visit of the vertices. To solve these problems, it was demonstrated that

- a) it is possible to span the vertices of the graph with respect of the degree bounds even if spanning trees satisfying the constraints do not exist
- b) in some cases, a spanning hierarchy with lower cost can be found even if spanning trees respecting the constraints exist [7].

One possible application domain of the spanning hierarchies is the broadcast in all-optical WDM networks where the splitting capacity of the vertices is limited (for example in [8]). To solve the optical routing problem under the degree constraints, a set of light trees (abusively called light forest) is usually proposed. Let us notice that in the literature not only tree-based solutions can be found. In [9], a special walk (a light-trail) is computed to cover the vertices without branching. The spanning hierarchies give a good alternative to find efficient spanning structures generalizing walks when branching are allowed. In this paper we demonstrate that the optimum of the degree bounded spanning hierarchy problem can be approximated. We propose a simple and efficient algorithm providing a good approximation of the optimal value.

In Section 2, we propose a quick presentation of the well-known degree-constrained spanning tree problem. After the related definitions, the degree constrained minimum spanning hierarchy problem and its complexity are presented in Section 3. The algorithm proposed in Section 5 uses the result of Section 4 to span stars and computes polynomially a spanning hierarchy respecting a given degree bound. The proposed algorithm guarantees a constant approximation ratio. The presentation is closed by discussions on the performances of the algorithm and on some perspectives.

2 Related works

The Degree-Constrained Minimum Spanning Tree DCMST problem was firstly introduced and investigated in [5] (it is also briefly mentioned in [10]). Let us suppose that the maximal degree of any vertices in the spanning tree must be at most $B \geq 2$. The authors justified the fact that this problem is NP-hard by stating that solving the DCMST problem with the degree bound B equal to two is equivalent to solve the minimum Hamiltonian path problem. Otherwise, by reducing the DCMST problem to an equivalent symmetric traveling salesman problem (TSP), Garey and Johnson [11] 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 the optimal tree is NP-hard [12]. In unweighted graphs, Furer and Raghavachari [13] gave an elegant algorithm that returns a spanning tree in which the degree of each vertex is at most $B + 1$, or returns a witness certifying that the degree bounds are infeasible. Goemans proved in [14] that this result can be generalized to weighted graphs. In polynomial time, we can find a spanning tree of maximum degree at most $B + 1$ whose cost is no more than the cost of a minimum cost tree with maximum degree at most B . Note that these results are formulated for homogeneous degree bound. When the degree bounds depend on the vertices, Goemans proved that one can find in polynomial time a spanning tree of maximum degree at most $B + 2$ whose cost is no more than the cost of a minimum cost tree with maximum degree at most B . The best result was presented by Singh and Lau in [15]. Their algorithm computes a spanning tree of minimum cost which violates the degree upper-bounds by at most one. Since it is not possible to obtain any approximation algorithm for the original problem, insisting on the satisfaction of all the degree upper bounds, this result is the best possible.

To solve spanning problems with different constraints, the hierarchy concept was proposed in [6].

3 Problem definition

Our objective is to find a minimum cost spanning structure without the hypothesis that this structure must be a sub-graph. For instance, it may be an arbitrary route connecting vertices.

Let $G = (V, E)$ be an undirected connected graph with vertex set V and edge set E . The graph G is valued by a strictly positive cost $c(e)$ associated to every edge $e \in E$. We are searching for routes in this graph. We suppose that the logical scheme of a route (the adjacency relation of nodes, vertices in the route, the succession of operations, etc.) is given by a connected graph $F = (W, D)$. For instance, F can be a path (a sequence of adjacent vertices and edges), it can be a cycle (if the vertices and operations have to be repeated in a cyclical manner), or an other graph. The association between the logical route F and the physical topology G can be given by a homomorphic mapping h and this "structure" can then be given by a triplet (F, h, G) . Trivially, the resulting structure (route) is not

necessarily a sub-graph in G . For instance, a *walk* or a *traversal* are connected routes, which may contain vertices and edges in G several times.

Definition 1 (Hierarchy) *If F is a tree, the connected structure defined by $H = (F, h, G)$ is called a hierarchy (cf. an example in Figure 1).*

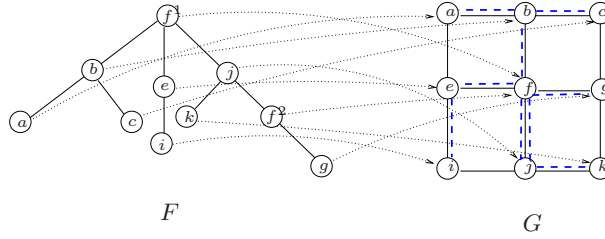


Fig. 1. Homomorphic mapping of vertices to define a hierarchy

To formulate the optimal spanning problem under capacity-like constraints, some simple definitions are needed. The *cost* of a structure $H = (F, h, G)$ is the sum of the costs of the edges used in H : $c(H) = \sum_{e' \in D} c(e)$, where $e \in E$ is the edge associated with $e' \in D$.

If an edge in G is used several times (it is associated to several edges in F), its cost is summarized several times. Since a hierarchy H in a graph G is given by a triplet (F, h, G) , and F is a tree, we talk about a leaf of the hierarchy when the concerned vertex is a leaf in F . Similarly, we talk about internal vertices concerning the non-leaf vertices in F . Several vertices of F may correspond to the same vertex v of G . These different occurrences will be labeled v^1, v^2, \dots if needed.

Our analysis deals with the minimum cost spanning problem of a graph G , where a positive integer B is given to bound the degree of vertices in the optimal route.¹ That is, the degree of each vertex in F (and not in G) is limited by B . Trivially, in interesting cases $2 \leq B < \max_{v \in V} d_G(v)$. The minimum cost, connected structure spanning the vertex set of G and respecting the degree constraints is always a hierarchy. With these considerations, we define our spanning problem as follows.

Definition 2 (Degree Constrained Minimum Spanning Hierarchy problem)

Given a connected graph $G = (V, E)$, a cost $c(e)$ for each $e \in E$ and an integer $B \geq 2$, the problem consists in finding a hierarchy $H = (F, h, G)$ where h is a homomorphism from a tree $F = (W, D)$ to $G = (V, E)$ such that:

¹ In DCMST problems, the degree bound expresses the overall capacity or budget of a vertex, but in our problem this bound corresponds to the maximal degree of each occurrence of the vertex in the spanning structure

- Each vertex $v \in V$ is associated with at least one vertex $v' \in W$.
- The degree constraints are respected in F : $d_F(v') \leq B, \forall v' \in W$.
- The cost $c(H)$ is minimal.

In the following, we will call the optimal solution "Degree Constrained Minimum Spanning Hierarchy" abbreviated by DCMSH.

Lemma 1. *For any degree bound $B \geq 2$, the DCMSH problem always has a solution.*

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

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

Lemma 2. *If among all the Minimum Spanning Trees (MST) of a graph G there exists one satisfying the degree constraint, it is an optimal solution for the DCMSH problem and all the optimal solutions are trees in G .*

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

Now suppose that an optimal hierarchy $H = (T, h, G)$ exists and it is not a tree in G . Because the MST T^* is an optimal solution of our problem, the cost $c(H)$ of the optimal hierarchy must be the same that the cost $c(T^*)$ of the MST solution. Trivially, the cost of a hierarchy is greater than or equal to the cost of its image in G : $c(I) \leq c(H)$, where I is the image (the 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 cost of the MST is therefore a lower bound for the DCMSH problem.

Remark 2: The result is not true if we only consider the *spanning trees* (and not the MSTs) satisfying the degree constraint.

Theorem 1. *The DCMSH problem 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 degree-constrained spanning hierarchy 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 degree-constrained spanning hierarchy $H = (T, h, G')$ 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 a DCMSH in G' because of Lemma 2. ■

Since the problem is NP-hard, guaranteed approximation algorithms are interesting to solve it in practical cases. To obtain an approximation of the DCMSH in an arbitrary connected graph, our approach is based on two elements:

- We consider the MST of the graph (which cost is a lower bound for every spanning hierarchy) as a start point.
- We decompose this tree into a set of connected stars. Each star is spanned by hierarchies with guarantee of cost and with respect to the degree constraint.

4 Degree constrained span of a star with hierarchies

Let S_k be a star with k edges, c its central vertex, and $c(S_k)$ the sum of its edges costs. Suppose that $B < k$. Then the minimum spanning hierarchy respecting the degree constraint contains several times the central vertex. Some leaves may also be duplicated. Since all edges of S_k must appear at least once in the hierarchy to ensure the spanning of all vertices, the computation of the DCMSH in a star is equivalent to the minimization of the length of the duplicated edges.

In the following, we propose a simple hierarchy computation to span stars with respect to the degree constraint B . The proposed algorithm does not guarantee the optimality of the hierarchy spanning the star, but it is enough to guarantee a good approximation ratio.

In our proposition, when edge duplications are needed, the less cost edges are used in an increasing order of edge costs. Moreover, these selected edges are duplicated at most once. Formally, let us make a partition of the edges of the star as follows. Let us create $\lfloor k/(B - 1) \rfloor + 1$ sets in the partition. Each set, except one (the last), contains $B - 1$ edges (if $k \bmod (B - 1) = 0$, the last edge set is empty). The $\lfloor k/(B - 1) \rfloor$ 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).

Each edge set corresponds to a "sub-star", which respects the degree constraint (with at most $B - 1$ edges). To obtain a connected hierarchy H_{S_k} spanning all the leaves, the sub-stars should be connected by the duplication of some edges. The less cost edge of each set is duplicated to make these connections. The central vertex is present in the final hierarchy as many times as there are sets in the partition. So, each central vertex occurrence respects the degree constraint B and the obtained structure is a hierarchy. Figure 2 illustrates the spanning hierarchy for $B = 4$ with $k \bmod (B - 1) = 0$ (the interest of the last occurrence of vertex c will be justified in the following).

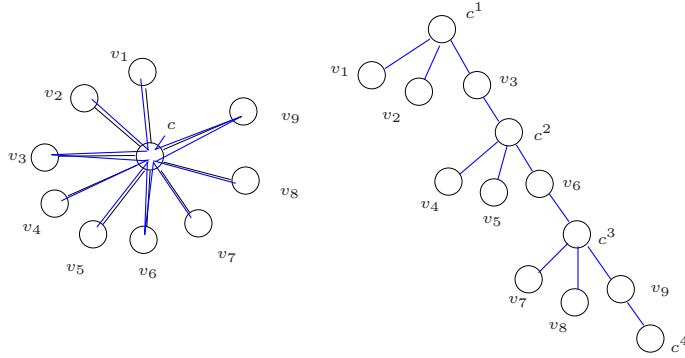


Fig. 2. Spanning hierarchy of a star computed by the proposed heuristic

Lemma 3. *The spanning hierarchy H_{S_k} computed by the proposed algorithm contains $N_c = \lfloor k/(B - 1) \rfloor + 1$ times the central vertex c s.t. each occurrence 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 $r = c(H_{S_k})/c(S)$ is bounded by $B/(B - 1)$.*

Proof. By construction, each occurrence of c in H_{S_k} have a degree at most B . In each sub-star of the partition, there is an occurrence of c and the number of exclusively spanned leaves is at most equal to $B - 1$ ². 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 $\lfloor k/(B - 1) \rfloor$ duplicated edges. Let D be the set of these duplicated edges. By choosing the less cost edges to duplicate, the cost of the duplicated part $c(D) = \sum_{e \in D} c(e)$ of the star is limited by

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

² a leave is spanned exclusively, if it belong to only one sub-star of the partition

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 $k < B - 1$), the central vertex has a degree strictly lower than $B - 1$.

The spanning hierarchy H_{S_k} corresponds to a caterpillar (tree in which all the vertices are within distance 1 of a central path), each vertex in this central path has a degree at most B . Moreover, it ensures that the central vertex occurrences in the first and in the last sub-stars have a degree less than B (if there is only one star, $\deg(c) < B - 1$, cf. Remark).

5 An approximation algorithm for the DCMSH problem

Since the cost of an MST gives a lower bound for the DCMSH problem, upper bounds for approximation algorithm can be computed regarding the MST instead of the optimal spanning hierarchy. In the following, we propose an approximation algorithm based on a decomposition of the MST in the graph.

5.1 A star decomposition of the MST

The MST, can be decomposed into a set of stars in the following way. Let $T = (V_T, E_T)$ be an MST with $|V_T| > 2$ and v_1 an arbitrary vertex in T . 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 while 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. Figure 3 illustrates the decomposition.

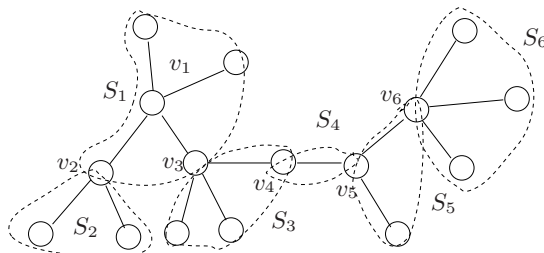


Fig. 3. A star decomposition of a tree

Since the stars are edge disjoint and cover all edges of T , trivially: $c(T) = \sum_{i=1}^k c(S_i)$, where $S_i, i = 1, \dots, k$ indicate the stars in the decomposition.

5.2 The proposed algorithm 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, \dots, S_k .
3. For each star S_i , compute a spanning hierarchy H_i as proposed in the previous section.
4. "Re-connect" the spanning sub-hierarchies H_i to form a connected spanning hierarchy H_A . 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 . 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 corresponds to the central vertex c_j of S_j associated to a vertex v_k in the original graph. Remember that c_j can be repeated in H_j but in this case its first occurrence has a degree $B-1$.
 - (a) If l_i has a degree 1 in H_i , it can be aggregated with the first occurrence of c_j in H_j and only one vertex can represent this vertex in the final hierarchy (this vertex in H_A corresponding to v_k respects the degree constraint B). It is the case of the vertex v_3 in our figure.
 - (b) If l_i has a degree 2 in H_i (it is not a leaf), the connection can be made as follows.
 - (b.a) 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).
 - (b.b) 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 without the violation of the degree constraint by the different vertices (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 hierarchy H_A can contain useless return edges (edges returning to a central vertex occurrence of a star s.t. the degree of this occurrence is equal to one). The useless edges can be deleted.

Theorem 2. *The previous algorithm 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 $S_i, i = 1, \dots, k$ in the decomposition. Trivially $c(T^*) = \sum_{i=1}^k c(S_i)$.

Using the result of Lemma 3, the obtained spanning hierarchy length is bounded by

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

The approximation ratio is immediately.

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

■

Remark 1: If $\deg(c) < B$ for all vertices $c \in V_{T^*}$, then the algorithm returns the 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.

Moreover, we propose to discuss the fact that our computation is not directly related to the optimal spanning hierarchy but to the MST of the graph.

5.3 Discussion about the heuristic

Since the proposed algorithm only uses the edges of an MST, the resulting hierarchy may be of poor quality for small values of B but the following theorem shows that its cost is the best which can be obtained when computing based on an MST.

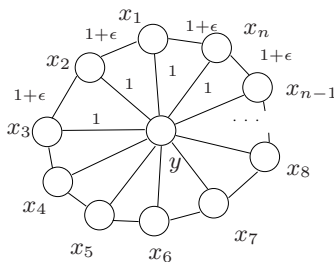


Fig. 4. A wheel graph used in Theorem 3

Theorem 3. *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 wheel graph with a central vertex y (see Figure 4). Suppose that $c(y, x_i) = 1$ for $i = 1, \dots, n$, $c(x_i, x_{i+1}) = 1 + \epsilon$ for $i = 1, \dots, n-1$, and $c(x_n, x_1) = 1 + \epsilon$.

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

The Minimum Spanning Tree of G is the star S of center y with n leaves. Let $H^* = (T^*, h^*, S)$ be an optimal hierarchy spanning the star S and respecting the degree constraint.

In T^* , there can be only one occurrence of every vertex corresponding to a leaf of S . If a leaf x_i of S has at least two occurrences in T^*

- If one of them is a leaf of T^* , it can be removed from T^* leading to a hierarchy spanning the same set of vertices with a smaller cost.
- Else, all occurrences are internal vertices of T^* . Let x_i^1 and x_i^2 be two occurrences. Let x_j be a leaf of T^* and $T^{*'}$ be the tree constructed from T^* by deleting the leaf x_j and replacing the label x_i^1 by x_j . Since all the neighbors of x_i^1 in T^* are occurrences of y , there still exists a homomorphism $h^{*'}$ between $T^{*'}$ and S leading to the same contradiction.

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 vertices corresponding to the leaves of S . Since T^* is a tree, its number of edges is equal to its number of vertices minus 1. Consequently, $c(H^*) = n + n_y - 1$. Any occurrence of y has at most B neighbors in T^* . So the number of edges of H^* is at most $n_y * B$ and we have $n + n_y - 1 \leq n_y * B$ which implies $\frac{n-1}{B-1} \leq n_y$. The cost of H^* is then at least $c(H^*) \geq n + \frac{n-1}{B-1} - 1 = \frac{B(n-1)}{B-1}$.

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

When the computation of the spanning hierarchy is not based on the MST, more interesting results can be obtained. For example, let the minimum Hamiltonian walk problem (case of) rapidly be reviewed. When $B = 2$, our approximation ratio is equal to 2, which is the worst case. Nevertheless, in this case, the problem is equivalent to find a minimum hamiltonian path in the metrical closure of G . It can thus be approximated with a ratio of 3/2 using for example the remarks of [16].

6 Conclusion

In this paper, we consider the problem of finding a minimum cost spanning structure when the degree of the vertices is bounded by an integer B . When this bound is due to a limited capacity each time the vertex is visited, the optimal structure is a hierarchy. We show that the problem is still NP-hard, but we provide an approximation algorithm to compute a degree constrained minimum spanning hierarchy with a ratio $B/(B - 1)$. Since the problem is equivalent to find a minimum hamiltonian path when $B = 2$, a ratio of 3/2 can always be assured. We also proved that the proposed approximation is the best possible with a heuristic based only on a minimum spanning tree. 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. 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
2. Cieslik, D.: The vertex degrees of minimum spanning trees. *European Journal of Operational Research* **125** (2000) 278–282
3. 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
4. Ravi, R., Marathe, M.V., Ravi, S.S., Rosenkrantz, D.J., Iii, H.B.H.: Approximation algorithms for degree-constrained minimum-cost network-design problems. *Algorithmica* **31** (2001) 58–78
5. Deo, N., Hakimi, S.: The shortest generalized Hamiltonian tree. In: *Sixth Annual Allerton Conference*. (1968) 879–888
6. Molnár, M.: Hierarchies to Solve Constrained Connected Spanning Problems. Technical Report 11029, LIRMM (2011)
7. Merabet, M., Durand, S., Molnar, M.: Exact solution for bounded degree connected spanning problems. Technical Report 12027, Laboratoire d’Informatique de Robotique et de Microélectronique de Montpellier - LIRMM (2012)
8. Zhou, Y., Poo, G.S.: Optical multicast over wavelength-routed wdm networks: A survey. *Optical Switching and Networking* **2** (2005) 176–197
9. 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
10. Obruca, A.K.: Spanning tree manipulation and the travelling salesman problem. *The Computer Journal* **10** (1968) 374–377
11. 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)
12. 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
13. 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
14. Goemans, M.: Minimum bounded degree spanning trees. In: *Foundations of Computer Science, 2006*. FOCS ’06. 47th Annual IEEE Symposium on. (2006) 273–282
15. 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
16. Hoogeveen, J.A.: Analysis of Christofides’ heuristic: Some paths are more difficult than cycles. *Oper. Res. Lett.* **10** (1991) 291–295