Hierarchies to Solve Constrained Connected Spanning Problems

Miklós Molnár

Hierarchies to Solve Constrained Connected Spanning Problems

Miklós Molnár

University Montpellier 2, IUT, Dep. of Comp. Sci.,
Laboratory LIRMM - UMR 5506- CC 477,
161, rue Ada
34095 Montpellier Cedex 5, France
molnar@lirmm.fr

Abstract. Given a graph and a set of vertices, searching for a connected and minimum cost structure which spans the vertices is a classic problem. When no constraints are applied, the minimum cost spanning structure is a sub-graph corresponding to a tree. If all the vertices in the graph should be spanned the problem is referred to as minimum spanning tree (MST) construction and polynomial time algorithms exist to find these trees. On the contrary, if only a subset of vertices is concerned then the problem becomes NP-hard. The computation of partial minimum spanning trees is known as the Steiner problem in graphs.
In some cases, constraints are present in the problem formulation: several constrained spanning and Steiner problems are known. Generally, only the tree-like spanning solutions were investigated. In this paper, we demonstrate that the cost optimal spanning structure in constrained situations is not necessarily a spanning tree. To find the optimal spanning, we propose the extension of the tree concept and we define the hierarchies. A hierarchy is obtained by a graph homomorphism from a tree to a given target graph which may refer vertices (and so edges) of the target graph several times. We prove that generally (including constrained spanning problems) the minimum cost connected structure spanning a set of vertices is a spanning hierarchy. To justify the introduction of spanning hierarchies, some spanning problems with various constraints are presented. The constrained spanning problems are frequently NP-hard and intensive research will be required to explore the properties of the optimal spanning hierarchies and to find exact and efficient heuristic algorithms to solve these newly formulated problems.

Keywords: Graphs, constrained spanning problems, spanning trees, spanning hierarchies, homomorphism, minimum spanning hierarchies, network

1 Introduction

One of the first studies on graphs was related to the computation of paths and walks. Many problems lead to a (shortest) path between a vertex pair in a given graph. The underlying graph can be weighted on the edges and/or the vertices, and different problems need paths computation (*e.g.,* computation of the shortest path or the path with maximum capacity, etc.). A path can be considered as a structure spanning its end points and the shortest path corresponds to the partial minimum cost spanning structure (sub-graph) of the end points. Walks corresponding to the solutions of some problems describe ordered visits of graph vertices and edges. Often, walks and spanning problems are related. Eulerian and Hamiltonian paths (if they exist) spanning the edges or the vertices of a given graph also correspond to special walks. Paths (or *elementary walks*) contain any graph element (vertex or edge) at most once. Several walk problems cannot be resolved using elementary walks. For example, Eulerian paths may contain repeated vertices. In *non-elementary walks* vertex and edge repetitions may occur.

When a set of vertices containing more than 2 vertices should be spanned and there is no further constraint, the minimum cost spanning sub-graph is a tree. (In our study, we suppose that the goal is to span a set of vertices by a connected structure.) If the spanning tree must cover a subset of vertices with minimum length, then the solution corresponds to a Partial Minimum Spanning Tree (PMST) or Steiner tree. The problem of finding a Steiner tree is NP-hard and several propositions for exact and approximated solutions exist (cf. a large state of the art for example in [12]). The Steiner problem is in APX and good approximated solutions have also been proposed (cf. [26] for the best solution currently available).

It is known that some minimum cost spanning problems under constraint cannot be solved using trees. Several cases have been analyzed in the literature both in constrained spanning and Steiner problems (cf. some examples in [10]). Generally, constrained spanning trees or sets of trees are supposed as minimum cost solutions. In some cases, the authors state that there is no tree-like solution for the given constrained spanning problem (cf. the analysis of the degree bounded spanning problem in [2]). Often, the constrained spanning problems are related to networks. For example, in all optical WDM networks switches which can have a degree greater than two in the spanning structure are not available anywhere. To perform multicast[1] routing with respect of the splitting constraints, the authors of [1] propose a special walk containing returns to some vertices. As also demonstrated for a different constrained multicast routing problem in [15], finding a spanning tree subject to multiple QoS requirements

[1] Multicast communications aim at point to multi-point and multi-point to multi-point communications

is not always possible. Nevertheless, the optimal (minimum cost) solution of these constrained partial spanning problems was not yet analyzed.

To find the minimum cost solution, we propose hierarchical structures related to graphs which can refer vertices and edges several times. *As our main result, we demonstrate that the optimal solution of the spanning problems always corresponds to a hierarchy and trivially spanning trees are special cases of spanning hierarchies.*

In the next of the paper, after the definition of the hierarchy concept in Section 2, Section 3 shows examples of the discussed constrained spanning problems. A new formulation and its hierarchy based solution is presented in Section 4. Section 5 describes briefly some known applications in which the hierarchy concept is applied successfully to find the minimum cost solution. The presentation is closed by some perspectives of this inchoate work.

2 Hierarchies in Graphs

To introduce the here proposed spanning structure, we begin with a trivial analogy. Let $G = (V, E)$ an undirected graph. An *elementary walk* is given by a sequence of vertices and edges forming a connected sub-graph beginning and ending with a vertex. It can be done by the sequence of the successively visited vertices, for instance:

$$P = (a, b, c, d) \tag{1}$$

In elementary walks vertices (and thus edges) are not repeated. The corresponding sub-graph is a path.

Non-elementary walks may contain vertices and/or edges several times and consequently several cycles. The following walk contains the vertices b and e twice (and so two cycles):

$$P_2 = (a, b, c, d, b, d, e, f, g, e, h) \tag{2}$$

Notice that the sub-graph generated by the walk (*i.e.* its *image*) is more simple as it is illustrated by Figure 1.

Homomorphism of graphs was proposed first in [27]. To handle similitudes between structures, the homomorphism has been studied in [17]. For graphs, the homomorphism is defined as follows. Let $H = (W, F)$ and $G = (V, E)$ two (undirected) graphs. An application $x : W \to V$ associating a vertex in V to each vertex in W is a homomorphism if the mapping preserves the adjacency: $(u, v) \in F$ implies $(x(u), x(v)) \in E$.

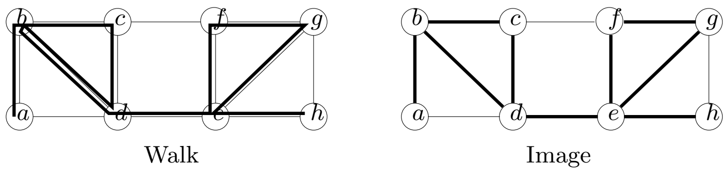

Fig. 1. A non-elementary walk and its image

The definition can also be given for directed graphs. Usually, in directed graphs, the homomorphism also preserves the direction of the arcs.

Graph homomorphism given by a triplet (H, x, G) can be applied to define spanning structures. The mapping from the base graph H to G determines a sub-graph in G which is called *image* of H in G. This image contains the vertices and edges in G which are mapped by the application x. Since several vertices in H can be associated with the same vertex of G (and so, several edges in H can be associated with the same edge of G), the image of H in G does not reflect the whole structure of H. The shortcoming of the classic image related to the homomorphism is that the generated sub-graph may correspond to different base graphs (this property is very useful for graph core determination but is useless disturbing for spanning problems). For spanning structures, we need a more detailed "image" in G preserving the basic properties of the graph H. Similarly to non elementary walks, where repetitions of vertices and edges can be clearly distinguished in the walk, the entire projection of H in G may be very useful. Moreover, significant values can be associated to the spanning structure: its length, its diameter and other derived values. In certain network applications, the topology graph is a valuated graph and the routes have they proper derived values (as the capacity, the end-to-end delay, etc.). The total knowledge of the spanning structure permits to define these derived values (cf. in [11] the definition of some lengths associated to walks).

Figure 2 illustrates a case resulted by a homomorphism from a non-connected graph H to a connected graph G such that the image obtained by the homomorphism is connected. Trivially, despite the fact that the image of H in G is a path, it does not correspond to a walk. Notice that connected images can be obtained only in connected components of the target graph G. In the following, we suppose that the target graph is always connected. To distinguish homomorphism corresponding connected walks, we propose the following definition.

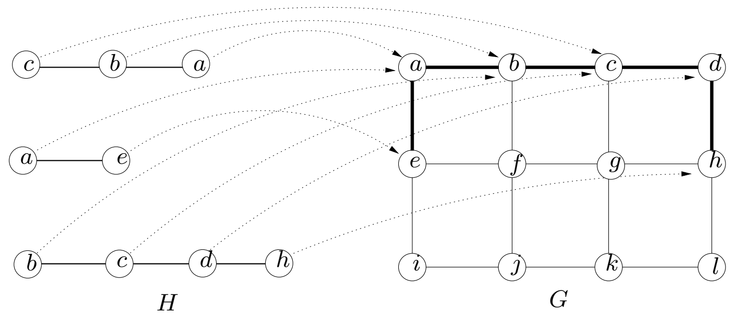

Fig. 2. Connected image of a non-connected graph

Definition 1 (Token connectivity). *A structure obtained by homomorphism from a graph H to a graph G is token connected iff H is connected.*

Since a connected image can correspond to a non-connected origin, the token connectivity is an important property of the spanning structure. In our study, we consider the token-connectivity of the spanning structures.

The goal of the spanning problems is the coverage of the vertices in a set $M \subseteq V$ of the graph $G = (V, E)$. In some cases the objective is the coverage by a sub-graph which is not necessary connected (*e.g.,* to create a matching, a Steiner forest) or in other cases by a connected one (*e.g.,* spanning tree, Steiner tree construction). Unfortunately, the sub-graphs are not sufficient to give all useful spanning structures. With the help of the following definition, we try to extend the possible spanning structures of a vertex set M applying a homomorphism.

Definition 2 (Token connected spanning of a vertex set). *A structure given by (H, x, G) spans the vertex set M, if the image of H in G covers all vertices in M and it is token-connected.*

To define both elementary and non-elementary finite walks in an undirected graph $G = (V, E)$, the following definition is used in [11].

Definition 3 (Walks in a graph). *Let $P = (W, F)$ be a connected graph which has not any vertex with degree greater than two (the graph P is a path). Let $x : W \to V$ be a homomorphism which associates a vertex $v \in V$ to each vertex $w \in W$. Trivially, (P, x, G) defines a walk in G.*

As x is a homomorphism, it preserves the adjacency of vertices: the edge $f = (w_1, w_2)$ belongs to F implies that the edge $e = (v_1, v_2)$ with

$v_1 = x(w_1)$ and $v_2 = x(w_2)$ is in E. If the application x is injective (that is a vertex $v \in V$ corresponds to only one vertex $w \in W$), then the walk defined in this way is an elementary walk in G (its image is also a path). If several vertices in W can correspond to a same vertex in V (and also several edges in F can be associated with a same edge in E), then (P, x, G) gives a non-elementary walk in G (the image of a non-elementary path can be an arbitrary connected sub-graph). Several homomorphisms from paths and cycles are discussed in [11].

The concept of the *tree* is used to describe connected sub-graphs without cycles. Trees are minimum connected graphs without path redundancies. Similarly to walks, tree-morph structures which we will call hierarchical walks or *hierarchies* can be defined with the help of homomorphisms. (Naturally, the same approach can be applied from different graphs, not only from trees.)

Definition 4 (Hierarchies in a graph). *Let $T = (W, F)$ be a connected graph without cycle (a tree). Let $x : W \rightarrow V$ be a homomorphism which associates a vertex $v \in V$ to each vertex $w \in W$. The application (T, x, G) defines a hierarchy in G.*

The base graph $T = (W, F)$ is the *origin* of the hierarchy and the sub-graph generated in G by the applications x from T is the *image* of the hierarchy. The *hierarchy* itself in the graph G corresponds to a "hierarchical walk" in which some vertices are branching vertices (there are the vertex occurrences corresponding to the branching vertices of the tree T). When a hierarchical walk on the structure is performed, the incoming walking object or token is duplicated in these branching vertices to continue the walk. Figure 3 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. To simplify, we use different occurrences of the vertices in V to represent the vertices of W in the obtained hierarchy. To distinguish the occurrences related to a vertex v, we will use $v^1, v^2, \ldots v^k$ if needed. Figure 3 gives an example of a homomorphism from a tree resulting a hierarchy. In the figure, each vertex of the base graph T indicates the name of the corresponding vertex in the graph G. A vertex in G can correspond to a branching vertex in T and an other mapping to this vertex can have as origin a vertex in T having a degree two or less as it is the case of the vertex g in our example. The image of T in G is a sub-graph indicated by bold lines in Figure 4. The proposed hierarchy corresponds to the folded tree in the target graph (to facilitate the distinction of the different parts of this hierarchy, a part of the structure is plotted usingy dotted line).

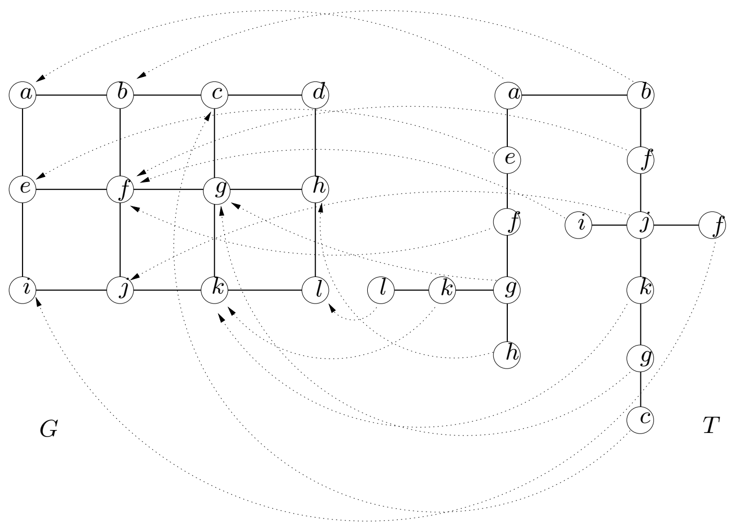

Fig. 3. Determination of a hierarchy using a homomorphism

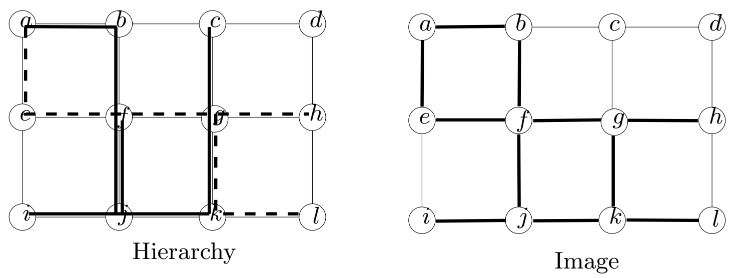

Fig. 4. The hierarchy and its image in the related graph

Trivially, if the application x is injective, then the hierarchy and also its image correspond to a tree in G. To simplify, we will say that the hierarchy is a tree. Using an analogy with elementary and non-elementary walks, a hierarchy can be seen as a structure regrouping elementary and "non-elementary" hierarchical objects. The gain of this definition is due to the possibility that hierarchical structures (which may contain several occurrences of the vertices and edges of G) can satisfy constraints when trees can not. Hierarchies offer more flexible structures to solve constrained spanning problems. We will see that the hierarchies permit the precise description of the cost optimal solution of some constrained spanning problems.

2.1 Particular Hierarchies

Hierarchies can be rooted or not. In some applications, rooted hierarchies should be applied (for example, there is a distinguished source node in a network corresponding to the root).

Definition 5 (Rooted hierarchy). *A hierarchy given by* (T, x, G) *is a rooted hierarchy if the base graph* $T = (W, F)$ *is a rooted tree.*

At each level of a non-empty rooted tree, there is a root vertex and a number of sub-trees. So, a rooted tree can be represented recursively:

$$T = (v(e_1, T_1, e_2, T_2, \ldots, e_k, T_k)) \tag{3}$$

where $T_1, \ldots, T_k$ are sub-trees linked to v by edges $e_1, \ldots, e_k$ respectively. According to the recursive decomposition of its origin $T = (W, F)$, a rooted hierarchy can also be given recursively:

$$H = (v(e_1, H_1, e_2, H_2, \ldots, e_k, H_k)) \tag{4}$$

Here, the sub-hierarchies are the hierarchies $(T_1, x_1, G), \ldots, (T_k, x_k, G)$ determined by the mappings $x_1, \ldots, x_k$ from the corresponding sub-trees $T_1, \ldots, T_k$. A rooted hierarchy can also be described using a hierarchically parenthesed enumeration of vertices (labeled by the vertices in the graph G). For instance, supposing that the root is the vertex a, the hierarchy in Figure 4 can be given by the following parenthesed enumeration of graph vertices:

$$H = (a(e(f(g(h, k(l)))), b(f(j(i, f, k(g(c)))))) \tag{5}$$

As it is shown, some vertices of the graph G are repeated several times and at different levels of this hierarchy.

Moreover, hierarchies can be directed or undirected according to the direction of the origin T. Basically, to define a directed hierarchy, the graphs T and G should be directed.

Definition 6 (Directed hierarchy related to a directed graph). *A directed hierarchy is given by* (T, x, G) *when the base graph* $T = (W, B)$ *and the target graph* $G = (V, A)$ *are directed having arc sets* B *and* A *respectively. The homomorphism preserves the direction of the arcs: if the arc* $b = (w_1, w_2)$ *belongs to* B *than the arc* $a = (v_1, v_2)$ *with* $v_1 = x(w_1)$ *and* $v_2 = x(w_2)$ *is in* A*.*

Since an undirected graph $G' = (V, E)$ can be replaced by a directed one $G = (V, A)$ defined on the same vertex set and containing a pair of opposite arcs in the place of each edge in E, the association of an undirected graph with a directed hierarchy is possible. As we will see later in this paper, directed hierarchies related to undirected graphs may have practical interest. For these cases, the homomorphic application is slightly extended in the following definition.

Definition 7 (Directed hierarchy related to an undirected graph). *A directed hierarchy in an undirected graph can be given by (T, x, G) where the base graph $T = (W, B)$ is a directed tree, the target graph $G = (V, E)$ is undirected and by using a direction-less homomorphism x. The direction-less homomorphism regards only the existence of an edge corresponding to the arcs of the base graph: if the arc $b = (w_1, w_2)$ belongs to B than the edge $e = (v_1, v_2)$ with $v_1 = x(w_1)$ and $v_2 = x(w_2)$ must be in E.*

2.2 Trivial Properties of Hierarchies

Some properties known for trees are also valid for hierarchies but not all. On the contrary, since trees in a graph can be obtained having particular (injective) homomorphism, the properties characterizing hierarchies are always true for trees. For instance, elements of Property 1 were formulated for trees but they may be formulated more generally for hierarchies.

Property 1. Due to its definition:

- a non-empty hierarchy in a graph is a token-connected structure: there is a walk between an arbitrary pair of vertex occurrences w_1,w_2 in the hierarchy
- the walk between w_1 and w_2 in a hierarchy is unique (the origin T of the hierarchy does not contain loops)
- the image of the walk between w_1 and w_2 in a hierarchy may be a sub-graph of G containing eventual cycles.

Property 2. In a hierarchy, a vertex $v \in V$ can be visited several times; and also an edge (arc) of the target graph G can be used by the hierarchy several times.

This property also follows from the definition.

Property 3. A hierarchy is not a sub-graph of G.

Property 4. Unlike trees, the image of a hierarchy in the graph G may contain cycles.

Property 5. Trees correspond to hierarchies obtained by injective homomorphic application x, and so without repetition of any vertex of G.

Property 6. Because the mapping function x associates one vertex of G to each vertex in H, the number of vertex occurrences in the hierarchy is equal to $|W|$.

Consequently, the number n_s of vertices spanned by a hierarchy obtained from a tree $T = (W, F)$ is limited by $|W|$: $n_s \leq |W|$.

For rooted hierarchies, we also mention the following important property.

Property 7. In a rooted hierarchy, each *vertex occurrence* has at most one parent.

2.3 Metrics Associated to Hierarchies

Length, cost and other significant values for applications can be associated to a hierarchy. These associations (the computation of the derived values) are natural for paths, cycles, trees, etc. Some extensions to walks as the net length of a walk (which is the difference between the number of forward arcs and the number of backward arcs in the walk performed in a digraph) have been formulated in [11]. Note that the distance between two vertex occurrences in a walk is not obligatory equal to the length of the path between the vertices in the image of the walk.

Different values can be associated to vertices and edges in a weighted graph. Since our paper deals with the introduction of the hierarchies, we note only some important aspects of the associated values.

Usually, one can distinguish three types of metrics associated to graph elements: additive, multiplicative and bottleneck type of metrics. This type indicates, how derived values for paths, tress, cycles, etc. can be computed. Typically, a cost is an additive metric: the cost of a path (a tree,...) is the sum of the cost of the edges (and/or vertices) composing the path (the tree). In the case of a multiplicative metric, the derived value corresponds to the product of the edge/vertex values composing the implied sub-graph. For the bottleneck type of metrics, the sub-graph (path, tree, ...) can be characterized by the minimum/maximum value of the elements composing it. (A good example for a bottleneck metric is the bandwidth in networks: the capacity of a route is equal to the minimal capacity of the links composing the route.)

There are several possibilities to associate values to graph related structures as walks and hierarchies. We propose to keep the interpretation known for trees to formulate the derived values associated to hierarchies. let us suppose that a hierarchy defined by (T, x, G) concerns a weighted graph G (otherwise the graph T can also be weighted, but we ignore here this case). Generally, a tree contains a graph element (vertex, edge) only once but graph elements may be repeated several times in hierarchies. To compute a derived value for a hierarchy, all components (occurrences) should be considered. In this way, an associated value based on additive metrics in G, can be computed as follows. Let us suppose that a cost $c(v)$ and $c(e)$ is associated to each vertex $v \in V$ and to each edge $e \in E$ respectively.

Definition 8. *The cost of the hierarchy H obtained is equal to:*

$$c(H) = \sum_{w \in W} c(x(w)) + \sum_{f \in F} c(x(f))$$

where $x(w), x(f)$ are the vertex / the edge corresponding to w and to f in G respectively using the mapping x.

The multiplicative and bottleneck values associated to a hierarchy can be formulated very similarly.

Given its importance in constrained spanning problems, the degree of a vertex occurrence in a hierarchy should also be defined. Let us suppose that H is a hierarchy obtained by (T, x, G), w is a vertex in T and the mapping associates $v \in V$ to this vertex in the hierarchy. With other world, a vertex occurrence v^i of v belong to H.

Definition 9. *The degree $d(v^i)$ of the vertex occurrence v^i in H is :*

$$d(v^i) = d(w)$$

where $d(w)$ gives the degree of w in T.

It is important to emphasize that v can also be mapped to an other vertex $z \in W$. The corresponding vertex occurrence v^j in H has a degree equal to the degree of z. So, different occurrences of a same vertex of G can have different degrees in a hierarchy. In directed cases, the out and in degrees can be defined in the same way.

Constraints can be defined on the different weights and associated values. For example, the QoS requirement of a network connection (usually path or tree) can be given as a set of constraints, which are classified as link constraints, path constraints, or tree constraints in [4]. The same logic can be followed for spanning hierarchies: constraint can consider vertex, edge, walk and hierarchy associated values. Our goal is the analysis of some constrained spanning problems. So, we give examples for the constrained problems in the following of the paper.

If the origin of a hierarchy is an infinite tree, the number of vertices in the obtained hierarchy is also infinite. For practical purposes, we focus only on finite hierarchies. A first network related presentation of hierarchies, some illustrations of optimal and quasi-optimal spanning hierarchies and related algorithms can be found in [19]. Although to discover further important properties, hierarchies should be thoroughly analyzed in future work.

After the definitions, we try to illustrate the usefulness of the hierarchies in spanning problems. It is known (cf.[12]) that minimum-cost spanning sub-graphs are minimum cost spanning trees. These structures respond to spanning problems when there are no constraints in the formulation. We are interested by token connected solutions of the spanning problems even if there are constraints to respect. We will demonstrate that *the minimum cost spanning structure is always a hierarchy* (cf. Theorem 1). In the following section, we mention some constraints and some constrained spanning problems analyzed in the literature.

3 Constrained Spanning Problems

A multitude of spanning problems in graphs with different constraints have been formulated. These problems aim at finding a spanning structure (generally a sub-graph) which covers a given subset of vertices in a target graph with respect of a set of constraints. In some cases the spanning structure can be unconnected (cf. for instance the Steiner forest problem in [8]), but often the connectivity of the spanning structure is required. The target graph can be weighted (usually by non-negative values associated to the edges and/or to the vertices) and an objective function can also be given. For instance, the cost or the diameter of the solution must be minimal. Most of the problems are mono-objective problems but some multi-objective formulations exist. In our study we are interested by mono-objective spanning problems when token connected solutions are wanted. Constraints can change the solution. For instance, if the degree of vertices in the spanning structure is limited by two, then the solution corresponds to a path (to the Hamiltonian path, if it exists) instead of a spanning tree. The feasibility and the complexity change from a problem to another by changing the constraints. Often, the presence of constraints makes the problem NP-hard, none the less that some unconstrained spanning problems are solvable in polynomial time [23] [5].

Some examples of constrained problems can be found in [10]. In this section, we enumerate a few of them. For the indicated problems, the introduction of the hierarchies improves the possibilities to cover the desired vertex set. Our enumeration is not exhaustive: hierarchies can be useful to solve other constrained spanning problems.

Problem 1 (Degree constrained and degree bounded spanning and Steiner problems). In these problems, a non-negative integer value is assigned to the vertices of the graph. When applying the constraint, the degree of a vertex occurrence in the spanning structure cannot exceed the given degree bound.

In the degree constrained minimum spanning tree problem the objective corresponds to finding the minimum cost tree among the spanning trees of the graph such that the tree meets the degree bounds on the vertices. For example, this problem corresponds to the well known minimum cost traveling salesman problem between two vertices when these vertices have degree bounds of one and the other vertces have degree bounds of two. The degree bound can be homogeneous or heterogeneous in the vertex set. The case of a homogeneous integer bound was introduced in [7] and intensively was analyzed (cf. some examples in [2] [24] [3] [5] [23] [25] [13]). The case of heterogeneous bounds is described for example in [9]. As it is mentioned in [2], unfortunately the tree-like solution does not always exist for this problem. We will show that the solution can be different from a tree and can be found in some cases where the spanning tree does not exist.

If only a subset $M \subset V$ of the vertices should be covered with respect of the degree bounds, then the problem is a degree constrained Steiner problem. In the literature, important analysis, exact and heuristic solutions can be found for example in [28] [2] [25] [16].

Problem 2 (Minimum cost spanning with size limit of the spanning trees). Let k be a positive integer value, $s \in V$ a source and $D \subset V$ a set of destinations in a weighted graph $G = (V, E)$. Let be the number of leaves and/or Steiner vertices in any spanning tree limited by k. The goal is to connect the destinations to the source by respecting the size constraint k. Trivially in some cases, for example when $|D| > k$, several trees rooted at the source are needed to connect all of the destinations to the source [20]. The solution is a set of trees rooted at the same vertex.

The combination of multiple constraints may be present in the spanning problems. Spanning tree and Steiner tree problem with revenues, budget and hop constraints have been formulated and analyzed to find a spanning tree with maximal revenue and with respect to constraints on the total budget and on the hop distances from a given source vertex in [6][14].

Problem 3 (Multiple constrained spanning problem with end-to-end constraint). Several multimedia applications in networks are based on multicasting and need to satisfy several criteria. They require multi-constrained multicast routes to do so. To model these problems, an m-dimensional vector $\overrightarrow{w}(e) = [w(e)_1, ..., w(e)_m]^T$ of additive-type metrics is associated with each edge $e \in E$. The end-to-end constraints (constraints from the source to the destinations) are also given by an m-dimensional vector $\overrightarrow{L} = [L_1, ..., L_m]^T$. Multi-constrained QoS path computation between nodes s and d consists in finding a feasible path satisfying:

$$w_i(p(s,d)) = \sum_{e \in p(s,d)} w_i(e) \leq L_i, \text{ for } i = 1, \ldots, m \tag{6}$$

This path computation is known to be NP-hard [18]. For multi-constrained multicast routing a feasible path should be exist for each destination and three formulations with different objectives have been proposed in [15]. In the Multiple Constrained Minimum Weight Multicast (MCMWM) problem the goal is to find a sub-graph G_M of G in which there exists a directed feasible path from a source s to each destination $d_j \in D$ and the length of the sub-graph is minimal. So, this routing problem corresponds to a particular spanning problem where end-to-end constraints limit the spanning structure.

4 A New Formulation of Constrained Spanning Problems

Hierarchies can be proposed to solve constrained spanning problems if the related problem does not obligatory needs a spanning sub-graph. Often, the solution of the constrained problems should not obligatory correspond to a connected sub-graph but can be a token-connected spanning structure. For example, the solution will be used to configure the control plane of a network. The spanning structure (the route, the scheduling plan, etc.) may cover the entire set or a subset of the vertices in a graph respecting constraints. Without loss of generality, we suppose that the objective of the spanning problem is the minimization of a cost function.

In some cases, this objective can be different from the cost minimization, but our results can be easily extended to other objectives (for example: to minimize the diameter of the spanning structure, etc.).

Constraints can be arbitrary except for the following particular constraint. Sometimes, a set of vertices should be connected by a k-connected graph with $k > 1$ (for instance to satisfy fault tolerance criterion). In our case, the spanning structure should only be 1-connected. *Moreover, we have not constraint on the spanning structure itself: it can be different from a sub-graph.* The only constraint on this structure is that it must be *token-connected.* So, our problems can be formulated as follows.

Problem 4 (Token-connected constrained spanning problems). Let a set $M \subseteq V$ be given in a graph $G = (V, E)$ assigned (positive) values on the edges and/or on the vertices. Let K be a (possibly empty) set of constraints excluding the k-connectivity with $k > 1$. Let G' be a token connected structure defined by the homomorphism (H, x, G) (with mapping x). Let V' and E' be the set of vertex occurrences and edge occurrences of this spanning structure (V' contains zero or several occurrences of a vertex $v \in V$ and E' contains zero or several occurrences of the edges $e \in E$). The token-connected constrained spanning problem is solved by the structure defined by (H, x, G) if the image of H in G covers the vertices in M and vertice occurrences in V' and edge occrrences in E' satisfy the constraints in K.

Let $c(e)$ be the cost associated to the edge $e \in E$. We consider the cost of the structure following Definition 8. We are looking at the token-connected minimum cost spanning structure covering M.

Theorem 1. *The minimum cost token-connected structure spanning M and respecting the constraints K is always a hierarchy.*

Proof. The spanning structure given by the homomorphism (H, x, G) should be token-connected with respect to K. Moreover it is with minimum cost. It is sufficient to prove that the origin $H = (W, F)$ of the minimum cost spanning structure cannot contain any cycles: it is a tree.

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

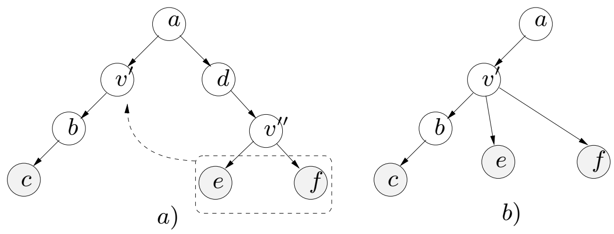

Fig. 5. A vertex $v \in V$ is present twice in the solution of an unconstrained problem

If the entire vertex set V of G must be covered, then the term *minimum spanning hierarchy* will be used. If a subset $M \subset V$ of vertices must be covered and the best (*i*.e. the minimum cost) spanning structure respecting the constraints is required we will talk about *partial minimum spanning hierarchies.*

The following, trivial lemma expresses that without constraints the solution is always a spanning tree.

Lemma 1. *The (partial) minimum spanning hierarchy of a given vertex set $M \subseteq V$ with no constraints in the graph $G = (V, E)$ corresponds to the (partial) minimum spanning tree of M.*

Proof. The minimum cost solution is always a hierarchy given by (H, x, G). To prove the current lemma, it is sufficient to prove that the homomorphism is injective: the (partial) minimum spanning hierarchy does not refer any vertex $v \in V$ twice when there are no constraints. Let us suppose that the optimal hierarchy contains a vertex $v \in V$ twice (v is associated to two different vertices of H). Let v' and v'' be these two occurrences of v in the hierarchy. The sub-hierarchies of v'' can be connected arbitrarily to v' or to v''. Let these sub-hierarchies be connected to v'. So v'' becomes a leaf and is superfluous when spanning M because the vertex v is already covered (cf. Figure 5). In this case, v'' can be deleted and a new shortest hierarchy can be obtained. This is in contradiction with the fact that the hierarchy from (H, x, G) has minimal length. □

4.1 The Interest of Spanning Hierarchies

Real-life spanning problems (routing problems in networks, constrained scheduling problems for example) do not explicitly impose a sub-graph as solution. From the technical point of view, the solution should be connected but may correspond to an arbitrary token-connected structure.

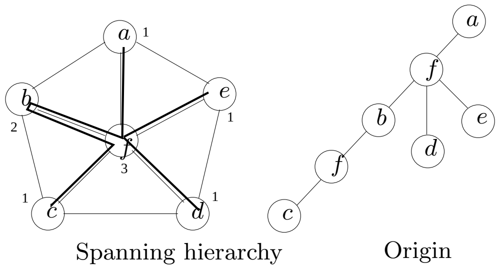

Fig. 6. A minimum cost spanning hierarchy

The cost minimization objective leads to the spanning hierarchies. Often, the spanning hierarchies are spanning trees, but in several cases, the hierarchies provide unquestionable advantages.

A. In some constrained problems a spanning hierarchy satisfying the constraints may exist even if spanning trees do not exist.

Figure 6 shows a graph in which constraints on the vertex degrees are imposed (these upper bounds are indicated in the figure). Due to technical limits (for example on the capacity of the elements modeled by the vertices), no vertex occurrence in the spanning structure may have a degree greater than the corresponding upper bound. Trivially in this graph, spanning trees cannot meet the degree constraints but spanning hierarchies can. To simplify, the cost of all edges is equal to one. The minimum cost structure is a spanning hierarchy drawn in bold. This hierarchy uses the central vertex f twice, but each occurrence of this vertex respects the degree constraint.

B. In some other constrained cases, a spanning hierarchy may be more favorable (cheaper) than the existing spanning trees satisfying the constraints.

To illustrate such a case, the degree bounded spanning problem can also applied. Even when spanning trees exist (and so there is at least a minimum cost spanning tree) satisfying the degree constraints, the minimum cost spanning tree is not necessarily the minimum cost spanning hierarchy. That is, a more favorable (lower cost) spanning hierarchy may exist. This phenomenon is illustrated in Figure 7, where the edges have different costs and the degree bounds are equal to two. The doted lines represent a minimum spanning tree (with length 5). The minimum spanning hierarchy also depicted with bold lines has a more favorable length (it is equal to 4).

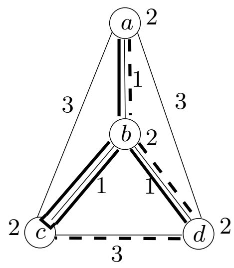

Fig. 7. The spanning hierarchy has lower cost than the minimum cost spanning tree respecting the constraints

As demonstrated in the previous section, spanning hierarchies can contain graph elements several times. These spanning structures are obtained by a homomorphism from a tree T to a target graph G. Real spanning problems can impose additional constraints on the applied mapping function x.

4.2 Constraints on the Spanning Hierarchies

The repetition of graph elements (the number of times an element is used) in the spanning structure itself may also be constrained.

Constraint 41 (Unique usage of the edges). *This additional constraint on the spanning hierarchy expresses that an edge of the target graph may belong to the spanning hierarchy at most once.*

Notice that the extremities of a unique edge can belong to the hierarchy several times. The constraint only prohibits the multiple usage of the target graph edges in the spanning structure. For instance, this constraint is imposed, if a label (address, wavelength, etc.) can be used at most once in a communication link of a network.

In some real cases of directed communications, the same label (*e.g.,* a wavelength) can be used in both direction of the communication links. Let us suppose that a directed spanning structure is required for these communications in an undirected graph corresponding to the network topology having bidirectional links.

Constraint 42 (Unique directed usage of the edges). *Applying this constraint, any edge of the graph may be used in a given direction at most once.*

In some cases, the origin and also the target graph are directed. Beyond the preservation of the direction of the arcs in the homomorphism applying between these directed graphs, an additional constraint on the number of occurrences of an target graph arc in the spanning hierarchy can be given as follows.

Constraint 43 (Unique usage of the arcs). *in a directed target graph this constraint implies that an arc can belong to the spanning structure at most once.*

Constraint 44 (Unique visite of the vertices). *This constraint implies that any vertex of the related graph can be used in the solution only once.*

Constraint 45 (k-constraints). *The previously mentioned constraints can be generalized to express that each vertex and/or edge can be used in the spanning hierarchy at most k times.*

The k-constraints may be useful to describe the spanning possibilities in k-graphs (for example, the optical routing in k-fiber optical networks).

Property 8. Trivially, when there are no vertices that can be present twice in the hierarchy then no edge/arc can be present twice. So Constraint 44 on the vertices is more strict than Constraints 41, 42 and 43.

The following example (cf. Figure 8) illustrates the impact of the mentioned constraints on the optimal hierarchy. In this example a directed spanning hierarchy with minimum cost is supposed from a source (vertex a) to a set of destinations (vertices d and e). The degree of vertices is limited to two in the spanning structure. In the first case (cf. Figure 8/a), when Constraint 44 is applied neither of the vertices can participate twice in the spanning structure: here a directed Steiner tree is the solution of the problem. This solution (whose length is equal to 6) is indicated by arrows. If the tree construction constraint is relaxed and a vertex can associated several times to the spanning hierarchy, but an edge of the graph cannot be used twice corresponding to Constraint 41, then a minimum partial spanning hierarchy with length 5 is the optimum as indicated on Figure 8/b. In the third case, Constraint 41 is replaced by Constraint 42, so an edge can be used twice: once in each direction. The optimum is another hierarchy with a length of 4 (cf. Figure 8/c).

Generally, minimum cost spanning problems have been formulated with the assumption that the solution is a spanning tree even if constraints are imposed in the problem formulation. The use of spanning

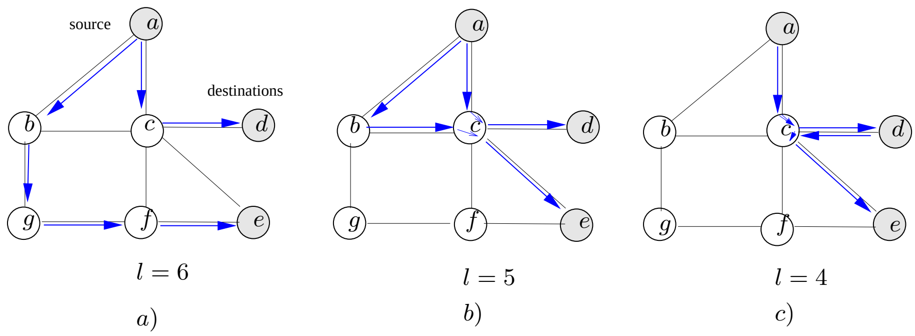

Fig. 8. Partial minimum cost directed hierarchies with different constraints

hierarchies in stead of trees can be profitable and permit a pertinent reformulation of these problems. In the following, we present three concrete applications corresponding to constrained optimization problems where spanning hierarchies are profitable.

5 Initial Results Related to Spanning Hierarchies

The first case presented here after corresponds to a special application of the degree constrained partial spanning problem, in the second the size of the spanning structures is limited and in the third there are several additive end-to-end constraints to respect from a source vertex to a set of destinations. For these constrained spanning problems, the usefulness of the spanning hierarchies is proved by several studies.

5.1 All Optical Multicast Routing

This problem is done in all optical WDM networks where "light-trees"[2] have been proposed for multicast communication from a source to multiple destinations. In this kind of networks, the light splitting capacity of optical switches is limited (and there are splitting incapable vertices) [22]. Multicast incapable nodes correspond to vertices having a degree bound two in the undirected graph modeling the network topology, whereas the degree of the optical splitters is not limited. Generally, optical switches can be traversed several times with the same wavelength if the incoming and outgoing links of the different crossing are different (nevertheless

[2] A light-tree is a directed partial spanning tree which can be realized using a single wavelength in the optical network

the wavelength should be unique in the fibers). In other worlds, there is no constraint on the number of occurrences of a vertex in the spanning structure but some of the constraints 41, 42, 43 must be satisfied. We saw that the optimum does not always correspond to a tree but to a hierarchy. To solve the multicast routing problem in all optical networks, "light-hierarchies"[3] have been proposed [30] [29]. The examples in Figure 9 illustrate a minimum cost light-hierarchy in a small optical network (circles and squares indicate multicast capable and multicast incapable nodes respectively).

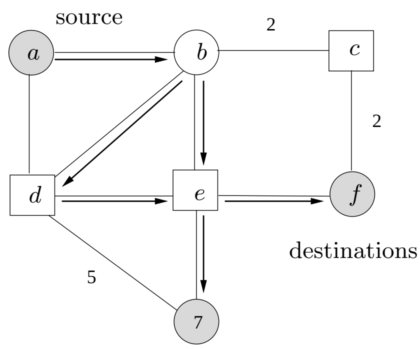

Fig. 9. A minimum cost light hierarchy

5.2 Size Constrained Spanning Problems

In the Steiner forest problem [8], a set of connection request is given by a set of vertex pairs $R = \{(s_1, t_1), (s_2, t_2), ...\}$ in an undirected graph $G = (V, E)$ and the goal is to find a minimum cost structure offering a path between the vertices s_i and t_i for all vertex pairs. Here we suppose a particular case of the Steiner forest problem, where a central vertex s belongs to each element of R. That is, this central vertex (a server or a source) must be connected with the other vertices (clients or destinations). In simple cases, the minimum cost spanning structure is a unique Steiner tree spanning the source and all destinations. Let us suppose that the spanning capacity of any connected spanning structure is limited. For

[3] A light-hierarchy is a directed partial spanning hierarchy realized by using a single wavelength respecting the optical constraints as the un by a homomorhismiqueness of the used wavelength in a fiber

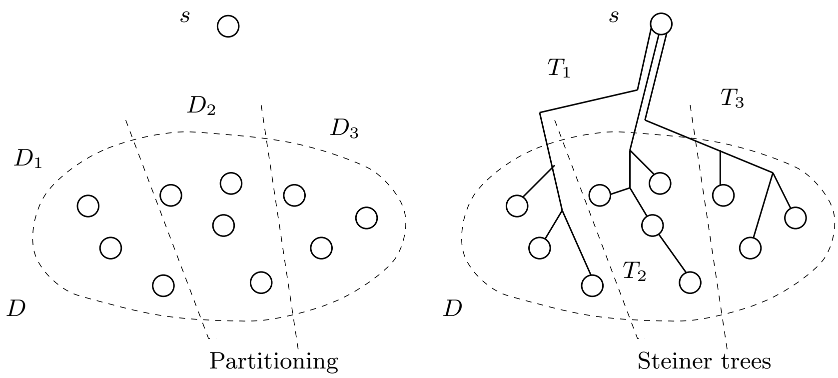

Fig. 10. The optimization is a superposition of a partitioning and several Steiner problems

example the number of destination vertices in the same connected spanning structure is limited and the objective is to find a spanning structure with minimum cost. It is easy to see that

1. the limitation may necessitate a partitioning of the destinations: several independent spanning trees are needed to cover the different (generally disjoint) destination subsets
2. the spanning tree of each subset in the partition is a Steiner tree rooted at the common source vertex s.

As Figure 10 illustrates, in this particular problem, the partitioning and the Steiner tree computation are inseparably related, and the common optimization is NP-hard (cf. in [20]). If the upper bound of the tree size is small, then several trees are needed to cover the set of destinations.

The optimal solution is a set $F^* = \{T_i, i = 1, ..., k(F^*)\}$ of trees T_i such that the tree T_i covers s and a subset D_i of destinations and $\bigcup_{i=1}^{k(F^*)} D_i = D$. Moreover the total cost is minimal:

$$F^* : \arg\min_{P \in \mathcal{P}} \sum_{i=1}^{k(P)} c(T_i) \tag{7}$$

where $\mathcal{P}$ denotes the set of possible partitionings of the destination set D and each partitioning P is composed of $k(P)$ non-empty subsets. The set of Steiner trees in the optimal solution forms a hierarchy as it is demonstrated by the following lemma.

Lemma 2. *A set of trees rooted at the same vertex (or having a shared common vertex) always corresponds to a hierarchy.*

Proof. Let $F = \{T_i, i = 1, ..., k\}$ a set of trees in $G = (V, E)$ having a common vertex s. This set can be obtained from a tree H using a homomorphic mapping x as follows. An origin corresponding to a tree H can be created as follows. H is initialized by s and each tree $T_i, i = 1, ..., k$ is added as an independent sub-tree to s. Each vertex of the obtained tree H is associated with one vertex of G and the mapping preserves the adjacency of vertices since the different T_i are trees in G. □

So, this optimization problem corresponds to a particular minimum spanning hierarchy problem. Notice that the sets of trees rooted in the same vertex are often called forests, but the hierarchy denomination following our definition is more precise.

5.3 Multiple Constrained Multicast Routing

The last example to illustrate the benefice of the hierarchies corresponds to the multi-constrained minimum cost multicast routing briefly presented in Section 3 as Problem 3.

It is stated in [15] that the solution of this routing problem is not always a tree. By comparing the original problem formulation (where a sub graph containing feasible paths is required) to the here proposed structure, the limitation of the MCMWM problem is due to the fact that the original formulation is based on the image G_M of the spanning structure and not on the structure itself. In a recent work, we proved that the cost/length optimal solution is a directed hierarchy [21]. Trivially, in the optimal spanning solution a vertex can be traversed and an edge can be used several times in both directions. A simple example of such a hierarchy is shown in Figure 11. In this example, the source is the vertex a and the destinations are the vertices f, g and h. A two-dimensional weight vector is assigned to each edge as depicted in the figure. Let 13 be the tolerated upper bound of each metric on the required paths. With the given values there is only one feasible directed path from the source to each destination. These feasible paths are also indicated in the figure. The unique feasible hierarchy and thus the minimum directed spanning hierarchy uses the edge (d, e) twice and traverses it in both directions.

The identification of the solution as a spanning hierarchy facilitates future research on the computation of good directed multicast routes.

6 Conclusions and Perspectives

In this paper, the generalization of the spanning tree concept in graphs was proposed. Trees do not permit the repetition of graph elements but

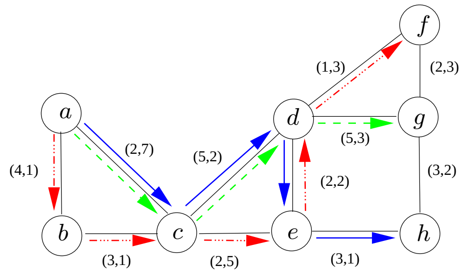

Fig. 11. Several paths can use the same edge

hierarchies obtained from trees by homomorphism permit. In the proposed more generic hierarchies, graph vertices or edges can be concerned several times. If there are constraints in a spanning problem, the question of finding minimum spanning structures satisfying the constraints often requires the determination of the minimum cost spanning hierarchy.

To illustrate the benefit of hierarchies, we discussed some special applications. In degree bounded spanning problems, we demonstrated that spanning hierarchies exist largely when spanning trees do not provide a solution and in some cases hierarchies provide better solutions even when spanning trees exist. When a size constraint is imposed on the spanning structures and a set of destinations should be connected to a source node, the optimal solution corresponds to a minimum cost hierarchy which can be decomposed into a set of trees rooted in the (central) source node. Our last example presented the case, where multiple QoS related constraints are given between a source vertex and some destinations. We mentioned that the optimal spanning structure solving this problem is also a hierarchy.

The analysis of minimum cost spanning problems under constraints promises further interesting challenges. Generally, these constrained spanning problems are NP-hard, and the solutions are spanning hierarchies. By reformulating the constrained spanning problems and proposing hierarchies instead of trees new conditions are obtained to resolve them. In future works, the different spanning hierarchy problems should be precisely analyzed from the point of view of their complexity, approximability, etc. In some cases and for real applications, the computation of optimal hierarchies is expensive and cannot be tolerated. Important research work should investigate the fast computation of advantageous spanning hierarchies for constrained spanning problems and related applications.

References

1. Ali, M., Deogun, J.: Cost-effective implementation of multicasting in wavelength-routed networks. IEEE J. Lightwave Technol., Special Issue on Optical Networks **18**(12), 1628–1638 (2000)
2. 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, p. 369. IEEE Computer Society, Washington, DC, USA (1995)
3. Boldon, B., Deo, N., Kumar, N.: Minimum-Weight Degree-Constrained Spanning Tree Problem: Heuristics and Implementation on an SIMD Parallel Machine. Parallel Computing **22**(3), 369–382 (1996)
4. Chen, S., Nahrstedt, K.: An overview of quality-of-service routing for the next generation high-speed networks: Problems and solutions. IEEE Network **12**(6), 64 – 79 (1998). DOI http://dx.doi.org/10.1109/65.752646
5. Cieslik, D.: The vertex degrees of minimum spanning trees. European Journal of Operational Research **125**(2), 278–282 (2000)
6. Costa, A.M., Cordeau, J.F., Laporte, G.: Fast heuristics for the steiner tree problem with revenues, budget and hop constraints. European Journal of Operational Research **190**(1), 68 – 78 (2008). DOI 10.1016/j.ejor.2007.06.012
7. Deo, N., Hakimi, S.: The shortest generalized Hamiltonian tree. In: 6th Annual Allerton Conference, pp. 879–888. IEEE, Monticello, Illinois, USA (1968)
8. Duin, C.W., Volgenant, A.: Some generalizations of the Steiner problem in graphs. Networks **17**(2), 353–364 (1987)
9. Fürer, M., Raghavachari, B.: Approximating the Minimum-Degree Steiner Tree to within One of Optimal. Journal of Algorithms **17**, 409–423 (1994)
10. 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)
11. Hell, P., Neetil, J.: Graphs and Homomorphisms. Oxford University Press, Oxford (2004)
12. Hwang, F.K., Richards, D.S.: The Steiner Tree Problem. Networks **22**(1), 55–89 (1992)
13. Könemann, J., Ravi, R.: Primal-dual meets local search: approximating mst's with nonuniform degree bounds. In: STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing, pp. 389–395. ACM, New York, NY, USA (2003). DOI http://doi.acm.org/10.1145/780542.780600
14. Konjevod, G., Krumke, S., Marathe, M.: Budget constrained minimum cost connected medians. In: U. Brandes, D. Wagner (eds.) Graph-Theoretic Concepts in Computer Science, *Lecture Notes in Computer Science*, vol. 1928, pp. 207–219. Springer Berlin / Heidelberg (2000)
15. Kuipers, F.A., Mieghem, P.V.: MAMCRA: a constrained-based multicast routing algorithm. Computer Communications **25**(8), 802–811 (2002)
16. Knemann, J., Ravi, R.: Quasi-polynomial time approximation algorithm for low-degree minimum-cost steiner trees. In: P. Pandya, J. Radhakrishnan (eds.) FST TCS 2003: Foundations of Software Technology and Theoretical Computer Science, *Lecture Notes in Computer Science*, vol. 2914, pp. 289–301. Springer Berlin / Heidelberg (2003). URL http://dx.doi.org/10.1007/978-3-540-24597-1_25
17. Lovász, L.: Operations with structures. Acta Mathematica Hungarica **18**, 321–328 (1967). URL http://dx.doi.org/10.1007/BF02280291

18. Mieghem, P.V., Kuipers, F.A.: Concepts of exact QoS routing algorithms. IEEE/ACM Trans. Netw. **12**(5), 851–864 (2004). DOI http://dx.doi.org/10.1109/TNET.2004.836112
19. Molnár, M.: Optimisation des communications multicast sous contraintes. University Rennes 1 (2008). URL http://ftp.irisa.fr/techreports/habilitations/molnar.pdf. Mémoire of Habilitation to advise doctoral theses (in French)
20. Molnár, M.: On the Optimal Tree-Based Explicit Multicast Routing. In: 2nd International Conference on Communication Theory, Reliability, and Quality of Service. Colmar, France (2009)
21. Molnár, M., Ballabas, A., Lahoud, S.: The cost optimal solution of the multi-constrained multicast routing problem. IEEE ??? **submitted** (2011)
22. Murthy, C.S.R., Gurusamy, M.: WDM Optical Networks: Concepts, Design and Algorithms. Prentice Hall, Upper Saddle River, NJ, USA (2001)
23. Papadimitriou, C.H., Yannakakis, M.: The Complexity of Restricted Minimum Spanning Tree Problems (Extended Abstract). In: H.A. Maurer (ed.) 6th Colloquium on Automata, Languages and Programming,ICALP, *Lecture Notes in Computer Science*, vol. 71, pp. 460–470. Springer, Graz, Austria (1979)
24. 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, pp. 438–447. ACM, New York, NY, USA (1993). DOI http://doi.acm.org/10.1145/167088.167209
25. 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**(1), 58–78 (2001). DOI http://dx.doi.org/10.1007/s00453-001-0038-2
26. Robins, G., Zelikovsky, A.: Improved Steiner tree approximation in graphs. In: SODA '00: Proceedings of the eleventh annual ACM-SIAM symposium on Discrete algorithms, pp. 770–779. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA (2000)
27. Sabidussi, G.: Graph derivatives. Mathematische Zeitschrift **76**, 385–401 (1961). URL http://dx.doi.org/10.1007/BF01210984
28. Voss, S.: Problems with generalized steiner problems. Algorithmica **7**, 333–335 (1992). URL http://dx.doi.org/10.1007/BF01758767
29. Zhou, F., Molnar, M., Cousin, B.: Is light-tree structure optimal for multicast routing in sparse splitting wdm networks? In: 18th IEEE International Conference on Computer Communications and Networks (ICCCN 2009). San Francisco, USA (2009)
30. Zhou, F., Molnar, M., Cousin, B.: Light-Hierarchy: The Optimal Structure for Multicast Routing in WDM Mesh Networks. In: Symposium on The 15th IEEE Symposium on Computers and Communications (ISCC2010), pp. 611 – 616. Riccione Italie (2010). DOI 10.1109/ISCC.2010.5643912