Computer Networks 56 (2012) 3136–3149

Contents lists available at SciVerse ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

The cost optimal solution of the multi-constrained multicast routing problem

Miklós Molnár [a,*], Alia Bellabas [b], Samer Lahoud [c]

[a] University Montpellier 2, IUT, Dep. of Comp. Sci., LIRMM, 161 rue Ada, 34095 Montpellier Cedex 5, France
[b] Université Européenne de Bretagne, INSA, IRISA, Campus de Beaulieu, 35042 Rennes, France
[c] University of Rennes 1, IRISA, Campus de Beaulieu, 35042 Rennes, France

ARTICLE INFO

Article history:
Received 14 October 2011
Received in revised form 2 April 2012
Accepted 17 April 2012
Available online 3 May 2012

Keywords:
Multicast routing
Quality of service
Multi-constrained Steiner problem
Hierarchy
Partial minimum spanning hierarchy

ABSTRACT

In this paper, we define the cost optimal solution of the multi-constrained multicast routing problem. This problem consists in finding a multicast structure that spans a source node and a set of destinations with respect to a set of constraints, while minimizing a cost function. This optimization is particularly interesting for multicast network communications that require Quality of Service (QoS) guarantees. Finding such a structure that satisfies the set of constraints is an NP-hard problem. To solve the addressed routing problem, most of the proposed algorithms focus on multicast trees. In some cases, the optimal spanning structure (*i.e.* the optimal multicast route) is neither a tree nor a set of trees nor a set of optimal QoS paths. The main result of our study is the exact identification of this optimal solution. We demonstrate that the optimal connected partial spanning structure that solves the multi-constrained multicast routing problem *always* corresponds to a hierarchy, a recently proposed generalization of the tree concept. We define the directed partial minimum spanning hierarchies as optimal solutions for the multi-constrained multicast routing problem and analyze their relevant properties. To our knowledge, our paper is the first study that exactly describes the cost optimal solution of this NP-hard problem.

1. Introduction

Quality of Service (QoS) multicast routing, known as the multi-constrained multicast routing problem, consists in computing a multicast structure that spans a source node and a set of destinations and respects the following constraints. For each destination node, the multicast structure should meet a set of QoS requirements such as delay, jitter, bandwidth, loss rate and cost. The most challenging QoS multicast routing techniques aim to support point-to-multi-point communications, which (i) satisfies the QoS constraints, and (ii) reduces the network resource consumption.

In a particular case, when there is only one additive constraint to satisfy and no cost function to minimize, a feasible solution[1] can be achieved by computing the shortest path tree.

To find a more appropriate solution within the feasible solutions, a cost function can be introduced. The cost can be an arbitrary metric and independent from the QoS metrics. For instance, the minimum cost solution can minimize the hop count. If we consider the case where the problem aims to construct the minimum cost multicast tree without QoS constraints, the solution corresponds to the NP-hard Steiner tree problem [1]. Moreover, when a QoS constraint is present in the Steiner tree problem, this corresponds to the constrained Steiner tree problem, which is also NP-hard [2].

* Corresponding author.
E-mail addresses: Miklos.Molnar@lirmm.fr (M. Molnár), alia.bellabas@irisa.fr (A. Bellabas), samer.lahoud@irisa.fr (S. Lahoud).

[1] A feasible solution contains at most a route satisfying the QoS constraints from the source to each destination.

1389-1286/$ - see front matter
http://dx.doi.org/10.1016/j.comnet.2012.04.020

In this paper, we consider the general case of the constrained multicast routing with multiple QoS constraints. We also suppose that an additive cost function is used to reflect the network usage. The two main reasons to couple cost optimization and multi-constrained routing are:

1. Real QoS requirements are often expressed in terms of multiple constraints and the multicast structure should contain a feasible path for each destination.
2. The interest of the network operators and thus implicitly of the users, is to minimize the network resource consumption.

It is indicated in [3] that guaranteeing QoS requirements and optimizing resource utilization are two conflicting interests and a trade-off should be achieved. We are interested in finding the best cost solution with respect to a set of QoS constraints. Thus, we investigate the multi-constrained cost optimization of multicast structures. Our objective is to find the minimum cost spanning structure, where the end-to-end paths, from the source to the destinations, satisfy the considered QoS constraints. In some cases, this optimal spanning structure is neither a tree nor a set of trees nor a set of optimal QoS paths. As is shown in [4], the multi-constrained routing problems are NP-hard even if there is only one destination.

In [5], the authors showed that a feasible partial spanning structure, that solves the multi-constrained spanning problem, may be different from a partial spanning tree. Indeed, they showed that the solution may be a sub-graph containing feasible paths and eventually minimizing a length function. Unfortunately, the sub-graph concept is not sufficient to define the (eventually optimal) solution.

The relevant criteria related to the multi-constrained multicast route are specified in the following. A multi-constrained multicast route should provide a feasible path toward each destination and be cost aware (cost optimal, if possible). This route may not correspond to a sub-graph. It can be a connected, graph-related structure, which is different from a sub-graph. To find the appropriate solution of the problem, we base our solution on recently proposed, graph-related structures, which can precisely describe multicast routes.

Our most important result, in this work, is the exact identification of the optimal solution. We show that a generalization of the tree concept that we call *hierarchy* always corresponds to the optimum. Furthermore, the hierarchies can also be used to describe interesting feasible solutions. Thus, we investigate on finding a partial minimum spanning hierarchy as the optimal solution of the multi-constrained partial spanning problem. The advantages of replacing the spanning tree concept by the spanning hierarchy are conclusive: the hierarchies enable the definition of the optimal solution. To the best of our knowledge, our study is the first one to formulate the optimal multicast routing problem using the hierarchy concept. This multi-constrained partial minimum spanning hierarchy problem is NP-hard.

In order to appropriately present our contributions, our paper is organized as follows. Section 2 specifies the multi-constrained partial spanning problems for multicast QoS routing and Section 3 provides an overview of the previously proposed approaches to solve these problems. Section 4 presents the hierarchy concept generalizing spanning trees as well as the basic properties of the hierarchies. In Section 5, we prove that the optimal solution of the multi-constrained partial spanning problem is always a hierarchy. The hardness of the problem is discussed, and the most relevant properties of the multi-constrained minimal cost partial spanning hierarchies are presented.

2. Problem formulations

At the routing level, several multimedia applications require multi-constrained multicast structures. In the literature, different objectives have been targeted and various solutions have been proposed. In this section, we present an overview of the most relevant formulations of the addressed problem. We show that the best solution corresponds to a minimum cost multicast structure with respect to the QoS constraints, which may be different from a tree. We propose the reformulation of the routing problem without the hypothesis that the solution is a sub-graph.

2.1. Previous problem formulations

Let $G = (V, E)$ be an undirected graph representing the network topology, where V is the set of nodes and E the set of edges. The source node and the multicast destination node set containing r destinations are denoted by $s \in V$ and $D = \{d_j \in V, d_j \neq s, j = 1, \ldots, r\}$ respectively. Each edge $e \in E$ is associated with m QoS weights given by a weight vector $\vec{w}(e) = [w_1(e), w_2(e), \ldots, w_m(e)]^T$. The end-to-end QoS requirements expressed as constraints from the source to the destinations, are given by an m-dimensional constraint vector $\vec{L} = [L_1, \ldots, L_m]^T$.

Recall that the QoS metrics can be roughly classified into additive metrics such as delay, multiplicative such as loss rate or bottleneck such as available bandwidth. As explained in [5], bottleneck metrics can easily be dealt with by pruning from the graph all links that do not satisfy the QoS constraints, while the multiplicative metrics can be transformed into additive metrics by using their logarithm. Therefore, and without loss of generality, we only consider additive metrics. The weight of a path $p(s, d_j)$ corresponding to the metric i is given by $w_i(p(s,d_j)) = \sum_{e \in p(s,d_j)} w_i(e)$. Thus, a path $p(s, d_j)$ is feasible if:

$$w_i(p(s,d_j)) = \sum_{e \in p(s,d_j)} w_i(e) \leqslant L_i, \quad \text{for} \quad i = 1, \ldots, m \qquad (1)$$

Unicast QoS routing consists in finding a feasible path $p(s, d_j)$, between a source node s and a destination node d_j. Often, a cost function c associated to the path should be optimized [6].

Multicast QoS routing aims to find a multicast structure (a multicast route) $M = (W, F)$, where W is an ordered set of node occurrences and F an ordered set of edge occurrences in the structure. Often, in constrained routing problems, M is not a sub-graph. As we will present here, to define this multicast structure it is not sufficient to determine the sub-graph containing it.

The multicast structure $M = (W, F)$ must contain at least one feasible path $p(s, d_j)$ from the source node s to each destination d_j, $j = 1, \ldots, r$. To solve the QoS multicast routing problem, some existing algorithms are focusing on finding a partial spanning tree, but the solution does not always correspond to a spanning tree. We will show that nodes and edges of the topology graph may be used multiple times for routing. Consequently, one node or one edge may be present multiple times in W and F respectively; this is why we propose to talk about occurrences of nodes and edges/arcs instead of talking about nodes and edges them selves.

Compared to the well known Steiner problem [1], where the minimum cost partial spanning tree is required, the construction of partial spanning structures satisfying multiple QoS requirements is even more complex.

To facilitate the comparison of the multi-constrained QoS paths, and to express the satisfaction of the QoS requirements, an interesting non-linear scalar length function has been introduced in [5], where the *length* of a path $p(s, d_j)$ is defined as:

$$l(p(s,d_j)) = \max_{i=1,\ldots,m}\left(\frac{w_i(p(s,d_j))}{L_i}\right) \tag{2}$$

In the following, we refer to this length as the non-linear length. If all of the constraints are satisfied, we trivially have $l(p(s, d_j)) \leqslant 1$. Thus, a feasible path can also be defined as a path whose non-linear length is less than 1, or by using the Pareto dominance:

$$\vec{w}(p(s,d_j)) \overset{d}{\leqslant} \vec{L} \tag{3}$$

For the multi-constrained multicast routing problem, three formulations have been proposed in [5]. The first formulation presents the Multiple Constrained Multicast (MCM) problem, in which a multicast sub-graph $M = (\{s, D\}, H)$ of the topology graph G is required, such that each destination node is connected to the source by the links in $H \subset E$. The authors state that M can obviously considered as a set of paths, from the source s to the destinations d_j, $j = 1, \ldots, p$. In our opinion, this definition is not sufficient to give the multicast route: the sub-graph cannot always exactly define the solution. The cost and QoS related end-to-end values associated with the sub-graph are not correlated with the values involved by the multi-constrained spanning objectives.

1. Fig. 1 shows an example of a sub-graph (presented with bold lines) spanning the source a and two destinations k and m. Trivially, the sub-graph contains several paths from the source a to the two destinations. For example, to reach the destination k, there are four paths in the given sub-graph: (a, b, c, e, f, g, i, k), (a, b, d, e, f, g, i, k), $(a, b, c, e, f, h, j, i, k)$ and $(a, b, d, e, f, h, j, i, k)$. Often, among the embedded paths, many of them can be feasible. Therefore, it is important to explicitly specify the paths in order to define the solution (the sub-graph as such cannot be configured for routing).
2. Furthermore, a sub-graph does not obviously reflect the end-to-end QoS requirements and the cost of the multicast route. Usually and supposing additive metrics, the cost (the length) of a sub-graph is equal to the sum of the values on the edges belonging to the sub-graph. This sum does not reflect either the end-to-end values of QoS metrics from the source to the different destinations or the overall cost (or length) of the multicast communication. Let us notice that even if the paths used for forwarding are made explicit, the cost of these paths may be different from the cost of the multicast communication. For example in a tree, the edges are used only once nevertheless that several paths traverse them. In Fig. 1, the edge (a, b) is used only once in any adequate multicast route embedded in the given sub-graph. Now, if the paths (a, b, c, e, f, g, i, k) and (a, b, d, e, f, g, i, k) are selected to connect the source a to k and m respectively, the edge (e, f) is used twice in this multicast route.

To conclude, in order to determine a solution, neither a sub-graph nor a set of paths from the source to the destinations is sufficient.

Unlike MCM, the Multiple Parameter Steiner Tree (MPST) problem searches for a partial spanning tree minimizing an arbitrary length function $l_{multicast}(T_M)$. For instance, the length function $l_{multicast}$ of the tree T_M can be proposed as follows:

$$l_{multicast}(T_M) = max_{i=1,\ldots,m}\frac{\sum_{e\in T_M} w_i(e)}{L_i} \tag{4}$$

Since the solution of this problem is a tree, it can be configured and used for multicasting. In the tree solution there is only one path from the source to any destination. The drawback of this problem formulation is that in some cases this kind of solution does not exist. The authors say that the MPST, although optimal in terms of resource utilization, does not always satisfy the constraints.

A third formulation is given by the combination of the two above cited ones; the Multiple Constrained Minimum Weight Multicast (MCMWM) problem consists in finding a sub-graph $M = (\{s, D\}, H)$ that contains a feasible path to each destination $d_j \in D$, such that the length $l_{multicast}$ of M is minimum. Trivially, the sub-graph minimizing the length can contain cycles and several feasible paths to the destinations. The definition is not accurate: we will show that the sub-graph corresponds to the image of an optimal solution and not to the solution itself. To better meet the needs, in the following, we reformulate the minimum cost multi-constrained multicast routing problem. In our proposition, a graph-related structure (and not necessarily a sub-graph) is needed for routing.

2.2. The cost minimum multi-constrained routing

Recall that the aim of routing is to find a structure containing at least one feasible path from the source to each destination, while minimizing the network resource usage expressed by an adequate cost function. For instance, the

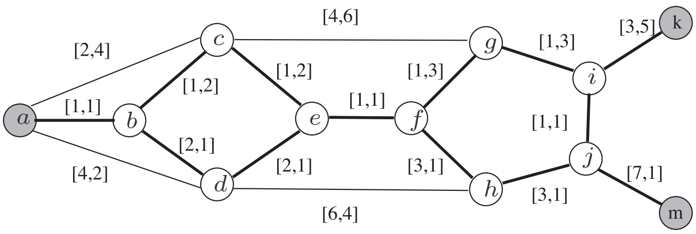


Fig. 1. A same sub-graph can be implicated in several multicast routes and with several set of paths.

cost may be an additive metric, which can be dependent or independent from the m QoS metrics expressed in the constraints [7].

However, the overall non-linear length $l_{multicast}(M)$ presented in Eq. (4) is not appropriate to characterize the quality of the multi-constrained multicast routing for two reasons:

1. First, this length and the non-linear length of the unicast paths are not obviously correlated. Indeed, the weight sum of the total links of the solution cannot indicate the end-to-end quality at the destinations. Moreover, if a given metric is not good for a given destination, this can be offset with another path, which is good for this metric and this destination. An example is shown in Fig. 2. Let us suppose that the source a should send messages to the destinations e and f with respect to a QoS vector $\vec{L} = [11, 11]^T$. The tree ($a(b(e), c(f))$ given by dotted lines has a minimal non-linear length. The overall length of the tree ($a(d(e, f))$) is higher but the QoS values at the destinations are lower than the maximal values on the first tree. If the cost of the routing corresponds to the hop count, then the second tree is also cheapest for multicasting.
2. Second, the cost and the length of the sub-graph do not always correspond to the cost and the cumulated length of the multicast route. Hereafter, we propose an adequate multicast routing structure, where the cost and the length correspond to the real values.

Therefore, we consider the cost of the multicast communication as the total cost of the forwarded data by using the edges (or arcs) of the multicast structure $M = (W, F)$:

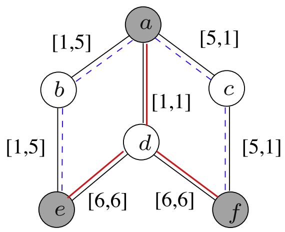


Fig. 2. The non-linear length of a multicast route is not appropriate.

$$c(M) = \sum_{e \in F} c(e) \quad (5)$$

As we will see, the structure M is different from a sub-graph. The sets W and F are not obviously exempt from repetition of graph elements. If an edge $e \in E$ of G is present twice in F (since it is used to forward the multicast message twice), its cost must be added twice to $c(M)$. The cost function $c(e)$ can be the frequently used hop-distance or any other positive additive cost expressed on the edges.

Taking into account the remarks, we propose the following formulation of the optimal QoS multicast routing.

Problem 1 (*Multi-Constrained Minimum Cost Multicast (MCMCM) Problem*). This problem deals with finding the structure $M^* = (W^*, F^*)$ with minimum cost $c(M^*)$, containing at least one path $p(s, d_j)$ from the source node s to each destination $d_j \in D$ that satisfies the given constraint vector $\vec{L}$ and minimizing the cost c:

$$\vec{w}(p(s, d_j)) \leqslant \vec{L}, j = 1, .., r \quad \text{and} \quad c(M^*) \text{is min} . \quad (6)$$

All of the above presented spanning problems with multiple constraints are NP-hard. For the three multicast routing problems presented in [5], the proof can be found in the same paper. The complexity of the MCMCM problem is discussed in Section 5. In the rest of this paper, we only consider the MCMCM problem, when referring to the multi-constrained multicast routing problem.

In [5], the authors state that the solution corresponds to a sub-graph, which may be different from a tree. This sub-graph contains a feasible path to each destination. It also may contain cycles that cannot be eliminated. As shown above, neither a sub-graph nor a set of QoS paths between the source and the destinations is sufficient to define the solution. Trivially, we raise the following important question: *What kind of structure corresponds to the optimal solution of the MCMCM problem?*

3. Related works

To formulate and solve (possibly approximatively) the optimal multi-constrained multicast routing problem, the different proposed approaches can be divided into three main classes, according to the objectives and interests of each class of problems: (i) the first class aims to compute a spanning tree that minimizes a given cost function without checking the feasibility of the solution, (ii) the second class computes a minimum spanning tree, with respect to a set of QoS constraints, (iii) and the third class computes a spanning structure called hierarchy introduced in the following section, with respect to a set of QoS constraints. Notice that in the two first classes of problems, the different proposed solutions aim to compute spanning trees. There are two reasons for basing multicast structures on multicast trees: (i) the data can be transmitted in parallel to various destinations along the tree links, (ii) and the tree structure avoids redundancies. We also notice that, if the addressed multi-constrained multicast problem has no cost function to minimize, any efficient multi-constrained unicast algorithm can be used, such as SAMCRA proposed in [8] and H_MCOP proposed in [9]. Therefore, we mainly focus the related works overview on the three above cited classes of problems.

In the first class of propositions, the solutions aim to minimize the cost of the multicast tree. The basic problem is known as the Steiner or Partial Minimum Spanning Tree (PMST) problem. This NP-hard problem has solicited a lot of interest, and a large number of exact and heuristic algorithms were proposed in the literature. Two good overviews are presented in [10,1].

In the second class, the addressed problems are known as the Constrained Steiner Tree (CST) problems that are also NP-hard [2]. To solve them, many of the proposed algorithms consider one QoS constraint as in [11], where the authors propose a branch-and-bound algorithm using the Lagrangian Relaxation and heuristics to get lower and upper bounds in the branch-and-bound tree. However, most of the proposed algorithms for the CST problem are heuristics and generally consider the end-to-end delay. In [2], the authors proposed a heuristic that constructs a low cost spanning tree, with respect to a bounded delay on each multicast destination. The proposed algorithm computes a delay-constrained closure[2] graph over the multicast group. Then, the algorithm constructs a constrained spanning tree of the closure graph using the well known Prim's algorithm [12]. Finally, the algorithm replaces the links in the spanning tree by the originally computed paths and removes the generated loops. In [13], the authors proposed an algorithm that approaches the minimum cost spanning tree solution with respect to the delay constraint. For that, the algorithm constructs two routing trees: a shortest path tree and an approached Steiner tree. Then, it identifies a given number of destinations k, where the difference between the delay observed in the Steiner tree and the delay in the shortest path tree for these destinations is large. For these destination nodes, the paths in the Steiner tree are replaced by the corresponding paths in the shortest path tree. The authors in [14] proposed a heuristic algorithm called the Bounded Shortest Multicast Algorithm (BSMA). This algorithm computes a least-delay tree that spans the source node and the destination nodes. Then, it iteratively replaces the links that can be replaced by other links that reduce the total cost of the tree, without violation of the delay constraint, until the total cost of the tree cannot further be reduced. The BSMA algorithm always finds a delay constrained tree, if such a tree exists, since it begins by the least-delay spanning tree.

When more than one QoS constraint are considered, the problem becomes more complex. In [15], the authors use the Lagrangian Relaxation Approach to construct special trees called LRATrees. Thus, the LRA algorithm relaxes the constraints and constructs a new problem with one objective function to minimize $(\max_\lambda L(\lambda) = \min c(p(s,d_j)) + \sum_{i=1}^{r} \lambda_i (w_i(p(s,d_j)) - L_i)$, for each destination $d_j \in D)$.λ_i determines how much the violation of the ith constraint should be penalized. The algorithm computes the minimum cost path between the source node and each destination node, then combines these paths to obtain an initial feasible tree. This tree is updated when the Lagrangian parameter $\lambda = [\lambda_1, \ldots, \lambda_m]^T$ is adjusted as follows: $\lambda^{k+1} = \lambda^k + \theta^k(\vec{w}(p(s,d_j)) - \vec{L})$, with $\theta^k = \frac{L(\lambda^{k+1} - \lambda^k)}{\|w_i(p(s,d_j)) - L_i\|^2}$. In [16], the author proposes an algorithm that searches for a feasible solution to the problem by finding, at first, a feasible tree that spans the source and some of the destinations. Then, it builds up the remaining destinations using a modified version of the H_MCOP algorithm [9]. This latter algorithm computes the shortest paths between two nodes by using the combined non-linear length function also presented in Eq. (2).

However, in some cases a partial spanning tree does not satisfy the required QoS constraints, while a set of unicast QoS paths does.

In the literature, few proposed algorithms allow solutions that are different from spanning trees. The Multicast Adaptive Multiple Constraints Routing Algorithm (MAMCRA) [5] is one of the most relevant algorithms that looks for structures that are different from spanning trees. In fact, the above cited algorithms aim to compute *a tree as the only allowed solution*, and this leads to the CST problem. MAMCRA is an algorithm that solves the multi-constrained multicast routing problem by computing *a special routing structure*. For this, MAMCRA proceeds in two steps:

- In the first step, the algorithm computes a set of optimal paths according to the non-linear length function defined in Eq. (2). The computation of the shortest paths uses a slightly modified version of SAMCRA [8], an exact multi-constrained unicast algorithm.
- In the second step, MAMCRA tries to eliminate the useless redundancies that are produced in the first step. For this, MAMCRA uses a greedy algorithm. The greedy algorithm iteratively compares two paths that share at least one node and deletes the longest prefix[3] from the source node to the farthest common node of one of them, if the resulting solution is still feasible.

4. Hierarchies as spanning structures in graphs

To identify the optimal solution of the multi-constrained multicast routing problem, we present simple examples, which illustrate the nature of this solution and we propose a brief overview of the hierarchy concept proposed in [17], which corresponds to the optimal structure.

[2] A closure graph on a set of nodes is a complete graph in which each link cost is equal to the cost of the shortest path between its nodes.

[3] The prefix of a path is the sub-path from the source node to an intermediate node.

Usually, *spanning trees* are considered as the only minimum cost partial spanning structures. They correspond to connected sub-graphs without cycles. Indeed, if a minimum cost structure is required to solve a partial spanning problem without any constraint, this structure always corresponds to a spanning tree called the Steiner tree.

To solve the optimal multi-constrained multicast routing problem, partial spanning trees have some limitations. For instance, the solution of the MPST problem corresponds to a partial spanning tree that minimizes a non-linear length function. The minimum tree solution always exists but it may not satisfy the end-to-end constraints as stated in [5]. Trivially, the minimum length tree does not always contain a feasible path to each destination. In some cases there is no tree solution for the constrained problem, and the minimum cost solution can be a different structure containing a feasible path for each destination. In the following, we consider all graph-related structures (not only sub-graphs) which contain a feasible path from the source to each destination as potential solutions of the multicast routing problem. The solution must be:

- a connected and oriented structure (containing a path from the source node to each destination)
- rooted in the source node.

For instance, a set of feasible paths as the result of the first step of MAMCRA [5] can be a simple solution. This set of paths is not cost optimal thus may lead to expensive multicast solutions Generally, among the feasible spanning structures, there is a solution with minimal cost.

In the following examples we illustrate the nature of the optimal solution in some simple cases. In the networks shown in Fig. 3, the multicast requests are given from a source to the destinations represented by grey nodes. The link values concerning two additive metrics are indicated and the end-to-end QoS requirements on these metrics are given by $\vec{L} = [15, 15]^T$ in the four cases. The only feasible (and so cost minimum) solution in the first figure is not a tree: it is a set of paths. In this case, the two paths are crossing at node *i*.

May the solution correspond to a set of paths? In the second topology, the cost minimum solution is not a set of paths: in this solution the link (*e*, *f*) is used only once, the node *f* is a branching node. It is a "virtual source" of two paths which are also crossing. The solution is no more a set of paths and it does not correspond to a tree.

After the first two examples, one can suppose that the solution is a directed acyclic graph (DAG, cf. [18]). The third and fourth figures show two cases where the solution crosses a link several times and in both directions and consequently it is not a directed acyclic graph. For example, in the last figure, the links (*f*, *h*) and (*h*, *i*) are used twice and the link (*i*, *j*) is used three times in the optimal solution. The presented solutions cannot be simplified without the violation of the QoS constraints at the destinations. Clearly, these solutions are neither trees nor set of paths from the source to the destinations nor other acyclic graphs but they *contain* feasible paths.

Our main question is the following: *how to exactly define this kind of structures?* Hereafter, we demonstrate that this optimal solution corresponds to a generalization of trees.

4.1. Hierarchies as Tree-Like Structures

At first, we briefly present the basic idea of the generalization of the spanning tree concept. For this, we propose to study walks. Walks are not sub-graphs but graph-related structures. In a walk, the nodes of a graph can be visited several times and it can be seen as a "folded" path in a graph. In the following, we denote these graph-related concepts by *structures*.

As is known, *trees* are connected (sub-) graphs without cycles. For the source based multicast routing, the tree is rooted at the source node. A simple notation can be used for rooted trees, in which the children of a node are enumerated between parenthesis after the parent node, as is illustrated by the following expression corresponding to Fig. 4:

$$T = (a(b, c(d, e), f)) \tag{7}$$

In a rooted tree, as a sub-graph, each node is present only once and has one parent node, except the root which has

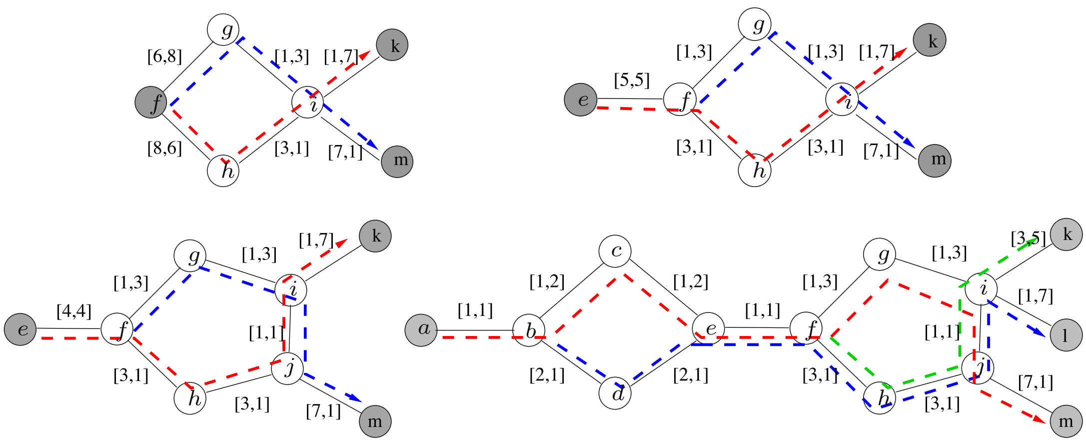


Fig. 3. Some examples of solutions of the multi-constrained multicast routing problem.

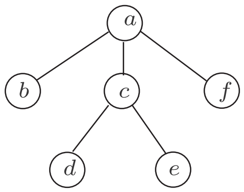


Fig. 4. Example of a rooted tree.

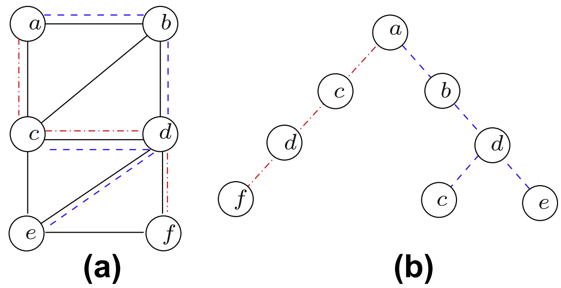


Fig. 5. Example of an undirected rooted hierarchy in an undirected graph.

no parent. Similarly to walks, "folded" trees can be imagined in graphs. In these structures, the graph nodes can be visited several times but the successive visits do not form a simple linear walk. The relations between the neighbor nodes of the structure keep the relations of a tree: a visited node can have more than one successors but only one predecessor. The corresponding *hierarchy* concept permits node and edge repetitions in the tree-based graph-related structures. An exact definition[4] of the concept can be found in [19]. In this paper, we only consider rooted hierarchies. In fact, rooted hierarchies are appropriate to describe the source-based multi-constrained multicast routing problem, since the source corresponds to a root node. The following definition can be considered as the extension of the rooted tree concept and gives useful and flexible hierarchical structures related to graphs.

Definition 1 (*Rooted hierarchy*). Related to a basic graph, a non-empty rooted *hierarchy* is a connected structure containing occurrences of the graph elements, where each *node occurrence* has at most one parent node.

A rooted hierarchy can be organized in levels. Even if the structure is not rooted, a "hierarchical walk" and levels can be organized from an arbitrary selected "root" node occurrence. That is why the name hierarchy was chosen for this structure.

A rooted hierarchy can be given by the hierarchical tree-like enumeration of node occurrences. For example, the hierarchy in Fig. 5 can be given by:

$$H = (a(c(d(f)), b(d(c, e)))) \qquad (8)$$

According to its definition, a hierarchy is not necessarily exempt from repetitions: nodes and edges of the basic graph may be present multiple times in a hierarchy. A hierarchy $H = (W, F)$ can be represented by the labeled set W of nodes and the labeled set F of edges, but these special sets may contain graph elements (labels) multiple times. (If the node and edge occurrences are distinguished in the sets, H corresponds to a tree.) A hierarchy is not a sub-graph, but (similarly to walks) a graph-related structure. In the example of Fig. 5, the nodes c and d are present twice in the previously described structure H. Since the different occurrences of the same element may play different roles in the hierarchy, the distinction and the identification of the occurrences is substantial. Indeed, a node occurrence can be an intermediate node in a hierarchy while another occurrence of the same node can be a leaf, like the node c in Fig. 5b. In the following, we will distinguish the occurrences of a node/edge x by different exponents x^1, x^2, ... when needed.

Even though a hierarchy is not a sub-graph, it generates a sub-graph in the basic graph. This sub-graph is called the *image* of the hierarchy and may contain cycles in the basic graph. The image of the mentioned hierarchy in Fig. 5a corresponds to the sub-graph generated by the nodes and edges used in the hierarchy. To facilitate the use of the hierarchies, we propose to retain some related terms:

- A *sub-hierarchy* of a (rooted) hierarchy H is a hierarchy that contains only elements of H.
- A *branching node occurrence* in a hierarchy H is a node occurrence that has at least two children.
- A *leaf* is a node occurrence that has no child in the hierarchy.

In Fig. 5, there are three leaves (one occurrence of c, e and f), the second occurrence of d is a branching node occurrence and $H' = (b(d(c, e)))$ is a sub-hierarchy of H rooted at b.

4.1.1. Some properties of hierarchies

Hierarchies can be directed or undirected. They also can span a set of nodes or all nodes. Moreover, a directed hierarchy may be related to an undirected graph. In this case, an arc can exist between two node occurrences in the hierarchy iff there is an edge between the related nodes in the basic graph. The direction of the arc can be arbitrary since the basic graph is not directed. For our routing problem, directed hierarchies are needed although the topology graph is not directed. Fig. 3 illustrates a directed hierarchy in an undirected graph.

Recall that trees are special hierarchies where each node has at most one occurrence. Thus, *a tree is a hierarchy*. Some properties of trees are true for hierarchies but not all, whereas all properties characterizing hierarchies are true for trees.

In a hierarchy, there is one and only one walk between two distinct node occurrences. Thus, there is only one walk from the root to an arbitrary node occurrence in a rooted hierarchy. This property is important for the optimal multicast routing. A connected graph-related structure containing

[4] The definition of the hierarchies can be based on graph homomorphism. A walk is a homomorphism of a path in a graph, which may return to a graph node several times. A hierarchy is a graph-related structure obtained by a homomorphism from a tree in a graph. A hierarchy, like walks, may return to a graph node several times.

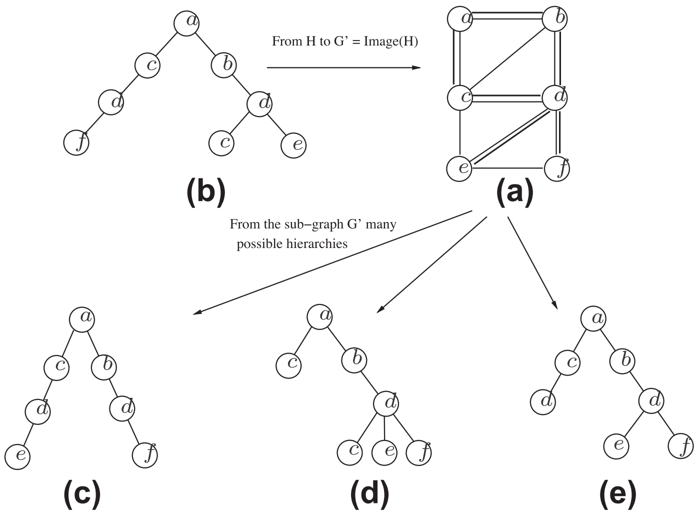


Fig. 6. A sub-graph and some corresponding generated hierarchies.

cycles can have several paths from the source to a destination. The hierarchy concept gives more precision: there is only one walk between the two node occurrences. Fig. 6 illustrates how from a given hierarchy we deduce its image $G' = Image(H)$. Corresponding to the same image G', we constructed three different hierarchies and can have more than these three. Indeed, a sub-graph does not exactly define a given hierarchy.

However, in the next section, we will see that the *directed rooted partial spanning hierarchies* allow an accurate and exact definition of the optimal solution of the multi-constrained partial minimum spanning problem. These rooted hierarchies are directed from the source to the destinations. They can precisely describe the multicast routes computed by MAMCRA.

4.2. Hierarchies used by MAMCRA

The hierarchy concept is useful to properly explain the structures computed by existing multicast routing algorithms. To solve the multi-constrained multicast routing problem, MAMCRA [5] is one of the most efficient heuristic algorithms known today. This algorithm that is briefly presented in Section 2 computes a set of optimal paths with minimum non-linear length. Since this set contains eventual redundancies, MAMCRA uses a greedy algorithm to eliminate some of these redundancies. In the following, we will prove that the set of paths computed in the first step and the final solution of MAMCRA are hierarchies. Notice that MAMCRA does not guarantee an optimal solution.

Lemma 1. *A set of paths from the same source node to different destinations corresponds to a rooted hierarchy.*

Proof. The lemma is trivial, since the set of paths is connected due to the common source node and each node occurrence has at most one parent in this set. □

Thus the first step of MAMCRA constructs a hierarchy. This set of paths may contain useless redundancies (loops). The second step of the algorithm tries to eliminate them.

Lemma 2. *The multicast routing structure obtained after the second step of MAMCRA is a rooted hierarchy.*

Proof. The first step of MAMCRA computes a set of paths that gives a hierarchy. The second step eliminates some redundancies based on the following operation. Let $p_1(s, x_1, d_1)$ and $p_2(s, x^2, d_2)$ be two paths sharing a common node x. Under some conditions as explained in [5], the part from s to x of one of the two paths is omitted: for example $p_1(s, x^1)$ is deleted and the concatenation $p_2(s, x^2) + p_1(x^2, d_1)$ is used for the destination d_1, if this new path is still feasible. To prove our lemma, it is sufficient to demonstrate that the redundancy elimination algorithm does not change the structure: the obtained solution is also a hierarchy. Since $p_2(s, x^2)$, $p_1(x^2, d_1)$ and $p_2(x^2, d_2)$ are paths, and they share only the node occurrence x^2, the children of x^1 will change their parent node to x^2, and they consequently still have one parent node. Therefore, MAMCRA returns a rooted hierarchy. □

However, it is important to state that the second step of MAMCRA gives a hierarchy but not a set of paths as its first step. Thus, a set of paths beginning at a common source form a hierarchy, but there are hierarchies that are not formed by such a set of paths. Fig. 7 shows the evolution of the multicast structure computed by MAMCRA, with five destinations c, d, e, g and h. Fig. 7a presents the five paths computed by SAMCRA in the basic graph (which are $p_1 = (s, a, c, d, f, g)$, $p_2 = (s, a, c, d)$, $p_3 = (s, b, c)$, $p_4 = (s, b, c, e, f, h)$ and $p_5 = (s, b, c, e)$), while Fig. 7b presents these paths as a hierarchy. This hierarchy enables to distinguish the different occurrences of the nodes that are shared by more than one path. As sub-paths $p_1(s, a^1, c^1, d^1)$ and $p_2(s, a^2, c^2, d^2)$ have the same sequence of nodes, one of them can be omit-

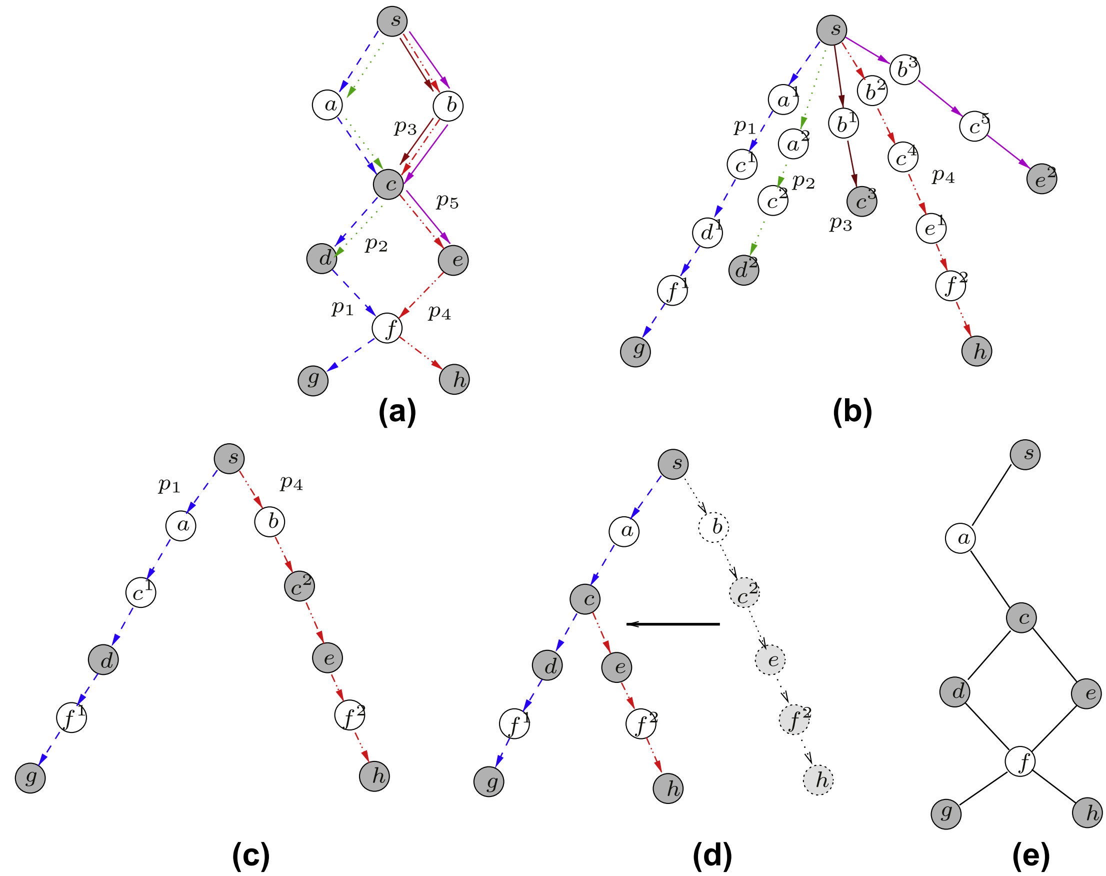


Fig. 7. The different phases of the hierarchies computed by MAMCRA and the image of the result.

ted. Therefore, d becomes an intermediate destination node in p_1. Similarly, the destinations c and e can be considered as intermediate nodes in p_4. These simplifications are detailed in [5]. We can consider that the five destinations are spanned by two feasible shortest paths as illustrated in Fig. 7c. According to Lemma 1, this set of paths corresponds to a hierarchy. Another occurrence of the destination c is a relay node in the path p_1. Let us suppose that the concatenation of $p_1(s, c^1)$ and $p_4(c^4, h)$ is a feasible path. In this case, the greedy algorithm in the second step of MAMCRA replaces the path p_4 by the concatenation as indicated in Fig. 7d. The resulting structure is a hierarchy and its image is also illustrated in Fig. 7e.

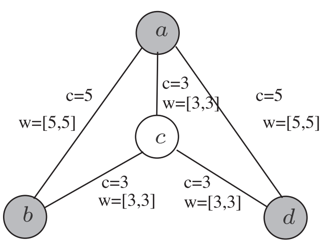


Fig. 8. The set of optimal paths does not contain the optimal spanning structure.

Concerning MAMCRA, it is important to emphasize that there is no obvious correlation between the optimal paths and the optimal solution. Consequently, even if an exact algorithm is used to eliminate the maximum of the redundancies from the set of optimal paths, the resulted hierarchy may be different from the optimal solution.

Lemma 3. *The optimal solution of the MCMWM problem does not necessarily belong to the set of shortest paths computed by the first step of MAMCRA.*

Proof. The proof is based on an example. Fig. 8 illustrates that the shortest paths, considering the non-linear length, are not necessarily included in the optimal solution. In the given graph, the cost c and the link weights w are indicated. Let us suppose that a is the source node, and there are two destinations: b and d. The QoS constraints are given by $\vec{L} = [7, 7]^T$. In Fig. 8, we show that the shortest paths (a, b) and (a, d), using the non-linear length function, do not contain the solution with minimum non-linear length. This optimal solution corresponds to the tree $(a(c(b, d)))$. □

It is well known that the cost of a shortest path tree can be arbitrarily far from the cost of the Steiner tree that spans the same set of target nodes. A similar observation can be made in our case between the minimum length paths computed by MAMCRA and the minimum length spanning hierarchy.

Lemma 4. *The overall non-linear length of the shortest paths computed by MAMCRA is arbitrarily far from the length of the optimal solution.*

Proof. Fig. 9 gives the proof. In the topology graph defined here, let s be the source node. The destinations are the nodes d_i, $i = 1, \ldots, k$. Using two metrics, the weight vector of each link is $\vec{w} = [1,1]^T$. Let us suppose that there are m links on the paths with minimum length between the source and each destination d_i (plotted in dotted lines). The QoS requirements are given by $\vec{L} = [m+2, m+2]^T$. The set of shortest paths form a feasible star S_1 and its overall non-linear length is $l(S_1) = k \cdot m$. Trivially, the minimum spanning hierarchy with the minimum non-linear length is the star S_2 plotted in bold lines. The length of this latter hierarchy is $lc(S_2) = k - 1 + m$. The ratio of the lengths is $R = \frac{k \cdot m}{k-1+m}$ and it tends to m when k tends to infinity. As m can be an arbitrary value, this ratio cannot be bounded. □

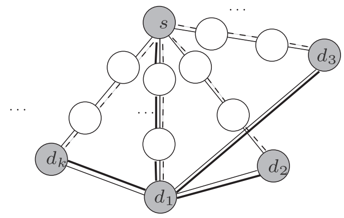


Fig. 9. The cost of the shortest paths cannot approximate the cost of the optimal solution.

The same proof can be made for the solution of the MCMCM problem using an arbitrary additive cost. The set of shortest paths does not approximate neither the multi constrained minimum cost nor the minimum length hierarchy.

5. The minimum partial spanning hierarchies to solve the multi-constrained multicast routing

The exact structure of the optimal solution of the multi-constrained multicast routing problem was not yet analyzed. The introduction of the hierarchies allows an accurate definition of this multicast structure.

5.1. The minimum cost solution

Let M be the minimum cost multicast structure solving the multi-constrained multicast routing problem from a source s to the destinations in a set D.

Theorem 1. *The optimal multicast structure M with respect to multiple constraints on positive additive metrics is always a directed partial spanning hierarchy.*

Proof. The optimal multicast structure must contain at most one directed feasible path from the source to each destination. Consequently, this structure is directed and connected. The structure does not necessarily span the entire node set, it is a partial spanning structure. It is sufficient to prove that it is a rooted hierarchy, *i.e.* in the minimum cost structure each node occurrence has at most one parent node and there is one arc between this parent and the node occurrence.

Let us suppose that a node occurrence v has two parent nodes (or two incoming arcs from the same parent node) in the minimum cost structure M spanning $\{s\} \bigcup D$. One of the incoming arcs of v can be dropped without loss of the connectivity and the remained structure covers $\{s\} \bigcup D$. Therefore, M cannot be the multicast structure with minimum cost. □

In the following, we refer to this solution as the Multi-Constrained Minimum Partial Directed Spanning Hierarchy, abbreviated by MC-MPDSH. Using the hierarchy concept, the MCMCM problem can be re-formulated as follows.

The MCMCM problem consists in finding the multi-constrained minimum partial directed spanning hierarchy containing at most one directed path $p(s, d_j)$ from the source to each destination $d_j \in D$, with respect to the given constraints: $\vec{w}(p(s,d_j)) \overset{d}{\leqslant} \vec{L}$.

Theorem 2. *The MCMCM problem is NP-hard.*

Proof. The minimum cost multi-constrained routing problem is NP-hard, even if there is only one destination (cf. the minimum cost constrained path problem in [20]). □

5.2. Properties of the MC-MPDSH

In the following, we review some particular properties of the MC-MPDSH. Indeed, these properties enable to (i) better identify the optimal solution (ii) demonstrate some limitations of known computational algorithms of spanning structures and (iii) design more efficient exact and heuristic algorithms.

Thus, we investigate the properties of the minimum cost solution $M = (W, F)$. The following two properties are always true for the optimal solution.

Property 1. *The leaves in the optimal spanning hierarchy M are destinations.*

Proof. Let us suppose that $t \in W$ is a leaf node but it is not a destination. In this case, at least t and its predecessor arc can be dropped and the destinations are still spanned with the residual connected structure. Therefore, M cannot be the minimum spanning structure. □

Note that a destination may also correspond to an arbitrary intermediate node occurrence. Due to this property, an MC-MPDSH has at most $|D|$ leaves.

Property 2. *In the optimal hierarchy M, the directed path from the source s to any arbitrary node occurrence v is a feasible path.*

Proof. M contains a feasible path from the source to each destination. For $v \in D$, the property is trivial. Let us suppose that $v \notin D$ (and thus, following Property 1, v is an intermediate node of a path toward a destination leaf). Let us also suppose that the path (s, v) is not feasible. Consequently, the paths from the source to the destinations which are extended from v are not feasible because the metrics are positive and additive. This is in contradiction with the fact that M contains a feasible path to all destinations. □

The solution is supposed to be a directed hierarchy even if the topology graph is undirected.

Property 3. *In the optimal hierarchy M, the edges of the topology graph G can be used multiple times and in both direction.*

Proof. The last example of Fig. 3 gives a case where an edge of the graph is used by the optimal hierarchy several times and in both directions. □

The nodes and the edges of the topology graph can be used multiple times in a hierarchy but, in the optimal solution, the node and arc occurrences are limited. In the following, we give limitations and upper bounds. These properties may be useful to design efficient hierarchy computation algorithms.

Recall that according to Property 1, a leaf of the MC-MPDSH is always a destination. Therefore, the MC-MPDSH has at most $|D|$ leaves. The following three properties are not trivial in MC-MPDSHs, but they considerably help the construction of the optimal solution.

Property 4. *In a directed path from the source s to an arbitrary destination d_j in the MC-MPDSH, a node $v \in V$ has at most one occurrence.*

Proof. Let us suppose that a node v has two occurrences v^1, v^2 in a directed "path" of the MC-MPDSH forming a cycle as illustrated in Fig. 10. If there is a destination d_1 between the two occurrences of v, the last segment (d_1, v^2) can be eliminated, and the obtained hierarchy has a lower cost and lower QoS weights at the destination nodes. If there are multiple destinations in the cycle, trivially, the last segment from the last intermediate destination node occurrence to v^2 can be eliminated. If there is no destination, then the entire cycle can be deleted. Thus, the optimal solution cannot contain two occurrences of the node v in the same directed path. □

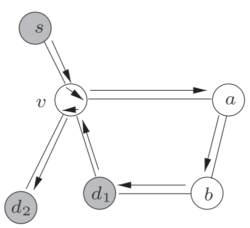


Fig. 10. A node is present twice in a path.

This property is also true regarding the source node itself: no directed path in an MC-MPDSH can return to the source node. Consequently, the source is present only once in an MC-MPDSH.

Property 5. *In the MC-MPDSH, the number of the occurrences of a node $v \in V$ is upper bounded by $|D|$.*

Proof. According to Property 4, a node has at most one occurrence in any directed path from the source. Let us suppose that the node v is present in all of the directed paths of the MC-MPDSH. In the worst case, there are $|D|$ directed paths such that any path (s, d_j) does not include any other destination d_k. Since an occurrence of v can belong to one of these paths, the upper bound is then equal to $|D|$. □

Property 6. *If a destination corresponds to a leaf node in the MC-MPDSH, then it has only one occurrence (the leaf occurrence) in the optimal solution.*

Proof. Let us suppose that the node d_i is present twice in the MC-MPDSH: a first node occurrence d_i^1 is a leaf in the path $p_1\left(s, d_i^1\right)$ and d_i^2 is another occurrence in an other path $p_2(s, d_j)$. Since the MC-MPDSH contains only feasible paths, the path $p_2(s, d_j)$ and its prefix $p_2\left(s, d_i^2\right)$ are both feasible. Thus, the path $p_1\left(s, d_i^1\right)$ is useless to reach d_i, and at least the last arc of this path can be deleted without affecting the feasibility of the solution. If there are intermediate destinations in $p_1\left(s, d_i^1\right)$, then the last segment from the last destination to d_i^1 can be deleted. Therefore, the two occurrences of d_i are not possible in the optimal solution. □

Property 7. *In the MC-MPDSH, a node can have at most $\left\lfloor \frac{|D|}{2} \right\rfloor$ branching node occurrences.*

Proof. Trivially the maximum number of branching node occurrences of a node v is reached when the occurrences of v are the only branching nodes in the hierarchy. Furthermore, each occurrence of v is a branching node with a minimum degree of three, where v has at least two outgoing arcs. Note that there are at most $|D|$ leaves in the hierarchy. If $|D|$ is even, the number of branching node occurrences of v is $\frac{|D|}{2}$. If $|D|$ is odd, the last occurrence of v is not a branching node, since it contains only one successor. Therefore, the upper bound of the branching node occurrences of v is $\left\lfloor \frac{|D|}{2} \right\rfloor$. □

Fig. 11 illustrates the worst case introduced in Property 7. The last two properties can be generalized in Property 8.

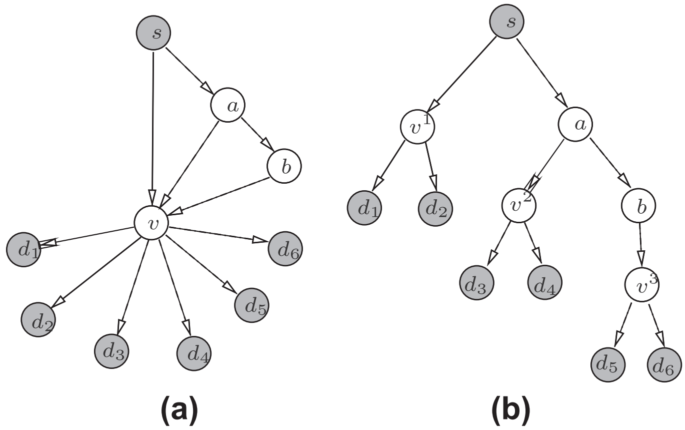


Fig. 11. A maximum number of branching node occurrences of a node in the hierarchy.

Property 8. *Let H = (W, F) be a sub-hierarchy of an MC-MPDSH with l_H leaves. Let v^j be the ith occurrence of the node v in the sub-hierarchy H and $d^+_H(v^i)$ its out-degree. For the occurrences of the node v, the following inequality always holds:*

$$\sum_{v^i \in W} d^+_H(v^i) \leqslant l_H$$

Proof. If v has a leaf occurrence in H, then according to Property 6, this occurrence is the only one of the node v in H, and v should be a destination. In this case, Property 8 is trivial.

If the different occurrences of v are not leaves, then each node occurrence has at least one child. More precisely, the occurrence v^j has $d^+_H(v^i)$ children and sub-hierarchies. In each sub-hierarchy there is at most one leaf node. According to Property 4, each path to a leaf contains at most one occurrence of v. □

Recall that the metrics in the graph are positive and additive. The following property enables to establish an important relation between the QoS constraints and gives a sub-optimality property.

Property 9. *Let M be an MC-MPDSH rooted at s and satisfying the constraints $\vec{L}$ in the leaves. Let M_v be a sub-hierarchy of M rooted at $v \neq s$. Let $\vec{w}(p(s,v))$ be the weight vector of the path p(s, v). The sub-hierarchy M_v is an MC-MPDSH from v to the destination occurrences that it contains with respect to the constraints $\vec{L_v} = \vec{L} - \vec{w}(p(s,v))$.*

Proof. A destination node has only one destination occurrence in the MC-MPDSH (other occurrences of the node can be used as intermediate relay node occurrences toward other destinations but the node should receive multicast messages as a destination only once). Therefore, we refer to destinations by destination occurrences. Let D_v be the set of destination node occurrences in M_v.

At first, we prove that $\vec{w}(p(v,d_j)) \overset{d}{\leqslant} \vec{L_v} = \vec{L} - \vec{w}(p(s,v))$ for all destination occurrences $d_j \in D_v$. Let us suppose that for the destination $d_j \in M_v$ the weight vector $\vec{w}(p(v,d_j))$ does not dominate $\vec{L_v}$. In this case, $\vec{w}(p(s,d_j))$ does not dominate $\vec{L}$ and thus M cannot be feasible.

Secondly, we prove that M_v is the minimum cost hierarchy in the set of the feasible hierarchies spanning the node v and D_v. Let us suppose that M_v is not a minimum cost solution but only a feasible solution spanning $v \bigcup D_v$. In this case, there is an MC-MPDSH $\widetilde{M}_v$ spanning $v \bigcup D_v$ with a lower cost than that of M_v. By replacing M_v with $\widetilde{M}_v$ in the hierarchy M, we obtain a feasible spanning hierarchy with a lower cost, which is in contradiction with the fact that M is an MC-MPDSH. □

This property may correspond to the well known Bellman's principle of optimality [21] and permits searching for the optimal solution with dynamic programming. Unfortunately, the computation of all smallest and possible sub-hierarchies is expensive. Recall that the problem is NP-hard.

As presented earlier, an edge can be used multiple times in both directions. Property 5 says that the number of the occurrences of a node in the MC-MPDSH is upper bounded by $|D|$. Trivially, the number of arc occurrences using the same edge is also bounded by $|D|$. However, the following additional questions can be raised. Some paths from the source to different destinations can use occurrences of the same arc. We say that two paths *share* an arc in a hierarchy, if the same occurrence of the arc belongs to both paths. When could the paths share a common occurrence of the arc? Trivially, the *shared* arc has only one occurrence in the optimal hierarchy. In any other case, when an arc cannot be shared by the paths, how many times is it present in the optimal hierarchy? The response is given by the following property.

Property 10. *Let $p(s, d_1)$ and $p(s, d_2)$ be two paths from the source to two distinct destinations both containing an arc a. The arc a is shared by $p(s, d_1)$ and $p(s, d_2)$ in the MC-MPDSH, iff the prefix of $p(s, d_1)$ and $p(s, d_2)$ from the source to the arc a is the same: $pref^{p(s,d_1)}(a) = pref^{p(s,d_2)}(a)$.*

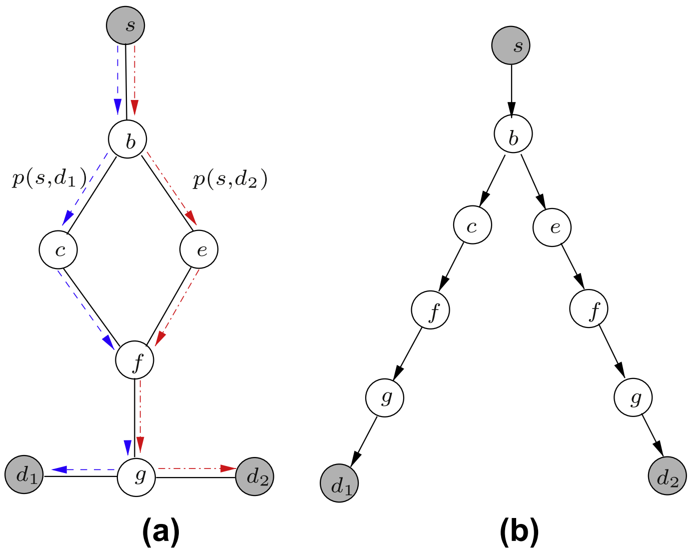


Fig. 12. Shared and unshared common arcs in the optimal hierarchy.

Proof. Let us suppose that$p(s, d_1)$ and $p(s, d_2)$ share the arc a but the prefixes $pref^{p(s,d_1)}(a)$ and $pref^{p(s,d_2)}(a)$ are different in the two paths. In this case, there is at least a node occurrence (the source of a or one of its common predecessor nodes) which has two predecessors in the spanning structure. This spanning structure cannot be a rooted spanning hierarchy (in a rooted hierarchy, each node occurrence has only one predecessor).

If the prefixes $pref^{p(s,d_1)}(a)$ and $pref^{p(s,d_2)}(a)$ are the same, then a is shared by$p(s, d_1)$ and $p(s, d_2)$ in the MC-MPDSH. Let us suppose that a is not shared by the two paths (*i.e.* there are two occurrences a^1 and a^2 of a in$p(s, d_1)$ and $p(s, d_2)$ respectively). In this case, another hierarchy can be constructed, in which a and its common prefix are shared. Trivially, the cost of this latter hierarchy is less than the cost of the hierarchy, which does not share a. The contradiction is trivial. □

Fig. 12 illustrates a graph that contains only one feasible path $p(s, d_1)$ and another $p(s, d_2)$ from the source s to the destinations d_1 and d_2 respectively. In the minimum cost spanning hierarchy, the two paths can share the arc (s, b) (the prefixes are the same, *i.e.* this prefix is empty), but they cannot share the arc (f, g). The corresponding minimum cost hierarchy is presented in Fig. 12b).

Property 11. *An MC-MPDSH h can be decomposed in a set of paths. (These paths are not necessarily originated at the source.)*

Proof. Let us suppose that there are h sub-hierarchies at the source. These sub-hierarchies are arc disjoint. Let us suppose that each sub-hierarchy can be decomposed in a set of paths. Then the decomposition of H is the union of the decompositions of the sub-hierarchies. A sub-hierarchy H_i of the source can be decomposed in a set of paths. Let $p(s, d_i)$ be a path from the source to a leaf node d_i in H_i. By deleting $p(s, d_i)$ from H_i a set of sub-hierarchies is obtained. The root node of these sub-hierarchies are some nodes in $p(s, d_i)$ different from the source. Each sub-hierarchy can be recursively decomposed in the same manner. □

Fig. 3 illustrates several examples of this decomposition.

To our knowledge, our study is the first one to analyze the structure of the optimal solution for the multi-constrained multicast routing problem. This study is needed for the computation of the optimal solution with the help of exact algorithms and can help the design of heuristics, which is an important challenge. These algorithms can be based on the search space reductions according to the given properties of the optimal solution. For instance, the sub-optimality of the hierarchies expressed by Property 9 permits construction of exact (but ex pensive) solutions based on dynamic programming. The bounds on the number of occurrences of graph elements in the optimal solution and established by the analyzed properties can help the design of efficient brunch-and-cut algorithms. The possible decompositions of the hierarchies give ideas for heuristic solutions. Important future investigations need to design efficient algorithms to solve this reformulated routing problem.

6. Conclusions and perspectives

The main result of our investigations on the multi-constrained multicast routing problem is the exact definition of the structure solving this problem. This structure always corresponds to a directed rooted hierarchy. However, finding the minimum cost hierarchies is an NP-hard optimization problem. The algorithms computing heuristic solutions, like MAMCRA, manipulate hierarchies such as the sets of paths rooted at the source and more or less redundancy-free spanning structures. In our paper, we proposed a first study of the properties of the hierarchy-type solutions. We argue that it is vital to identify the optimal solution structure of the addressed problem. An additional result of our analysis is that the optimal solution does not obviously belong either to the set of minimum cost paths or to the set of shortest paths computed by using the well known non-linear length. Since hierarchies may contain multiple occurrences of a node or an edge/arc, most of the enumeration algorithms are not appropriate to compute the optimal solution. Following this first study on multi-constrained minimum partial spanning hierarchies, important future works on exact algorithms and heuristics are needed to find more efficient algorithms.

References

[1] F.K. Hwang, D.S. Richards, Steiner tree problems, Networks 22 (1992) 55–89.

[2] V.P. Kompella, J.C. Pasquale, G.C. Polyzos, Multicast routing for multimedia communication, IEEE/ACM Trans. Netw. 1 (3) (1993) 286–292.

[3] X. Masip-Bruin, M. Yannuzzi, J. Domingo-Pascual, A. Fonte, M. Curado, E. Monteiro, F. Kuipers, P.V. Mieghem, S. Avallone, G. Ventre, P. Aranda-Gutierrez, M. Hollick, R. Steinmetz, L. Iannone, K. Salamatian, Research challenges in QoS routing, Comput. Commun. 29 (1) (2006) 563–581 (NoE E-Next special issue).

[4] P.V. Mieghem, F.A. Kuipers, Concepts of exact QoS routing algorithms, IEEE/ACM Trans. Netw. 12 (5) (2004) 851–864.

[5] F.A. Kuipers, P.V. Mieghem, MAMCRA: a constrained-based multicast routing algorithm, Comput. Commun. 25 (8) (2002) 802–811.

[6] G. Feng, On the Performance of Heuristic H MCOP for Multi-Constrained Optimal-Path QoS Routing, Proceedings of the 18th International Conference on Advanced Information Networking and Applications, vol. 2, IEEE Computer Society, Washington, DC, USA, 2004, pp. 50–55.

[7] N. Ben Ali, M. Molnár, A. Belghith, Multi-constrained QoS Multicast Routing Optimization, Tech. Rep. 1882, IRISA (February 2008).

[8] P. Van Mieghem, H. De Neve, F. Kuipers, Hop-by-hop quality of service routing, Comput. Networks 37 (3-4) (2001) 407–423.

[9] T. Korkmaz, M. Krunz, Multi-constrained optimal path selection, in: INFOCOM, 2001, pp. 834–843.

[10] P. Winter, Steiner problem in networks: a survey, Networks 17 (1987) 129–167.

[11] V. Aggarwal, Y.P. Aneja, K.P.K. Nair, Minimal spanning tree subject to a side constraint, Comput. Oper. Res. 9 (4) (1982) 287–296.

[12] R.C. Prim, Shortest connection networks and some generalisations, Bell Syst. Tech. J. 36 (1957) 1389–1401.

[13] S. Kumar, P. Radoslavov, D. Thaler, C. Alaettinoglu, D. Estrin, M. Handley, The MASC/BGMP Architecture for Inter-Domain Multicast Routing, in: SIGCOMM, 1998, pp. 93–104.

[14] Q. Zhu, M. Parsa, J.J. Garcia-Luna-Aceves, A Source-Based Algorithm for Delay-Constrained Minimum-Cost Multicasting, in: INFOCOM, 1995, pp. 377–385.

[15] D. Wang, F. Ergun, Z. Xu, Unicast and Multicast QoS Routing with Multiple Constraints, in: e.a. Marson (Ed.), Quality of Service in Multiservice IP Networks, vol. 3375, Springer, Berlin/ Heidelberg, 2005, pp. 481–494.

[16] G. Feng, A multi-constrained multicast QoS routing algorithm, Compt. Commun. 29 (2006) 1811–1822.

[17] M. Molnár, Optimisation des communications multicast sous contraintes, University Rennes 1, mémoire of Habilitation to advise doctoral theses, 2008. URL <http://ftp.irisa.fr/techreports/habilitations/molnar.pdf> (in French).

[18] K. Thulasiraman, M.N.S. Swamy, Graphs: theory and algorithms, John Wiley & Sons, Inc., New York, NY, USA, 1992.

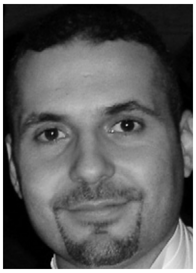
[19] M. Molnar, Hierarchies to Solve Constrained Connected Spanning Problems, Tech. rep., Laboratoire d'Informatique de Robotique et de Microélectronique de Montpellier – LIRMM (September 2011). URL <http://hal-lirmm.ccsd.cnrs.fr/lirmm-00619806>.

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

[21] R. Bellman, Dynamic Programming, Princeton University Press, 1957.

Miklós Molnár is with the University of Montpellier 2, laboratory LIRMM and he is a full professor in Computer Science. He was graduated at the Faculty of Electrical Engineering, University BME of Budapest in 1976 and received the Ph.D. degree in Computer Science from the University of Rennes 1 in 1992 and the French HDR degree in 2008. His main research activity is in combinatorial optimization to solve network related problems. His results are related to NP-hard optimization problems, constrained spanning and Steiner problems, routing algorithms for unicast, in-cast and multicast communications in optical and in wireless networks, dependable communications, energy aware protocols and optimization.

Alia Bellabas is with IRISA (Institut de Recherche en Informatique et Systèmes Aléatoires) laboratory in Rennes (France). She received her Engineer Degree in Computer Science in 2007 from the High School of Computer Science (Algeria), and a master degree from the University of Versailles (France) in 2008. She received the Ph.D. degree in Computer Science at INSA of Rennes in 2011. Her main research topics and interests include multicast routing, quality of service in networks, and combinatorial optimization. She is involved in many research programs at the national level.



Samer Lahoud received the Ph.D. degree in Computer Science from Telecom Bretagne, Rennes. After his Ph.D. he spent 1 year with Alcatel-Lucent Bell Labs Europe working as a research engineer. Since 2007, he has been with the University of Rennes 1, where he is working as assistant professor, and with IRISA of Rennes, where he is taking part in the research activities. His main research activity is in network design, combinatorial optimization and engineering algorithms for communication networks. He has been involved in many research programs at the national and the European level.