Multi-constrained QOS multicast routing optimization

Naouel Ben Ali, Miklos Molnar, Abdelfatteh Belghith

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION INTERNE

N° 1882

MULTI-CONSTRAINED QOS MULTICAST ROUTING OPTIMIZATION

NAOUEL BEN ALI, MIKLÓS MOLNÁR AND ABDELFATTAH BELGHITH

ISSN 1166-8687

IRISA

CAMPUS UNIVERSITAIRE DE BEAULIEU - 35042 RENNES CEDEX - FRANCE

Institut de Recherche en Informatique et Systèmes Aléatoires
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 02 99 84 71 00 – Fax : (33) 02 99 84 71 71
http://www.irisa.fr

Multi-constrained QoS Multicast Routing Optimization

Naouel Ben Ali* , Miklós Molnár ** and Abdelfattah Belghith ***

Systèmes communicants

Publication interne n ° 1882 — Février 2008 — 48 pages

Abstract: In the future Internet, multimedia applications will be strongly present. When a group of users is concerned by the same traffic flow, the multicast communication can decrease considerably the network bandwidth utilization. The major part of this kind of multicast communication needs quality of service (QoS) specification. Often, the QoS is given as a set of QoS criteria and the computation of feasible or optimal routes corresponds to a multi-constrained optimization. Finding the multicast graph respecting the defined QoS requirements and minimizing network resources is a NP-complete optimization task. Exhaustive search algorithms are not supported in real networks. Greedy algorithms was proposed to find good multicast sub-graphs. The local decisions of greedy algorithms can lead to solutions which can be ameliorated. To improve greedy algorithm solution, we propose, first, ICRA algorithm which is an enhanced version of the well known Mamcra algorithm but is also limited. As Meta-heuristics are good candidates to find better solutions using a controlled execution time, we propose, secondly, Taboo-QMR algorithm which is a Taboo Search based algorithm to reduce the multicast sub-graph computed by the first step of the algorithm Mamcra. Simulations of all approaches are run based on random graphs and show that the application of Taboo-QMR algorithm presents a tangible enhancement in almost 32 per cent of the cases.

Key-words: Network, multicast routing, QoS, multi-constrained optimization, taboo search

(Résumé : tsvp)

Ce travail a été réalisé dans le cadre d'une collaboration entre l'Irisa de Rennes et l'ENSI de Tunis financée par la coopération INRIA - STIC - Universités Tunisiennes.

* naouel.benali@ensi.rnu.tn
** miklos.molnar@irisa.fr
*** abdelfattah.belghith@ensi.rnu.tn

Centre National de la Recherche Scientifique (UMR 6074) Université de Rennes 1 – Insa de Rennes
Institut National de Recherche en Informatique et en Automatique – unité de recherche de Rennes

Optimisation du routage multicast avec plusieurs contraintes de Qualité de service

Résumé : Les applications multimédias distribuées déployées dans l'Internet actuel ont fait natre un besoin urgent en infrastructure de communication offrant une qualité de service garantie et prévisible. Générallement, la qualité de service est exprimée sous formes de plusieurs critères à satisfaire et le calcul d'une structure de routage satisfaisant plusieurs critères correspond à un problème d'optimisation difficile. D'ailleurs, le calcul du graphe multicast satisfaisant les besoins requis par les applications et minimisant l'utilisation des ressources est un problème NP-complet. Comme les algorithmes de recherche exhaustive ne sont pas envisageables dans le cas de déploiement dans un réseau de communication, des algorithmes heuristiques et souvent gloutons peuvent tre adoptés pour trouver la solution à un tel problème. Sauf que, les décisions locales et irrévocables prises par ce genre d'algorithme peuvent mener à des solutions qu'on peut améliorer. Pour améliorer, Mamcra, l'unique approche gloutonne existant dans la littérature, nous proposons tout d'abord ICRA, qui est un algorithme incrémental qui corrige certains points faibles de Mamcra mais reste toujours limité. Puis, nous proposons, l'adoption de Taboo-QMR, qui est basée sur une approche métaheuristique, la recherche tabou. Taboo-QMR permet d'éviter les minima locaux dans lesquels sont piégés les algorithmes itératifs tels que ICRA et Mamcra et permet de trouver une meilleure solution la plupart des cas.

Mots clés : Réseaux, routage multicast, QoS, optimisation multicritère, recherche "tabou"

1 Introduction

Recently, the Internet has shown a tremendous growth. Emergent multicast applications like audio/video conferencing, video on demand, IP-telephony, etc. usually have Quality Of Service (QoS) requirements, which include bandwidth, bounded delay, jitter and loss rate. Several IP multicasting techniques have been proposed to support point-to-multipoint communications by sharing link resources leading to a reduction in network resource consumption. All these techniques are based on IP multicast routing protocols which use shortest path tree algorithm based on one single metric, typically delay or hop count. Multicast applications today needs to optimize more than one metric that's why multi-constrained QoS routing should be applied. QoS routing is a routing scheme under which paths for flows are attributes by taking into account flow requirements and are based on some knowledge of resource availability in the network. Multicast routing deployed in the Internet aims to use resources efficiently. In a point to multipoint session, p destinations will receive the same information. Sending p times over each shortest path to each individual multicast member is inefficient. Sending single packets through the shared links and duplicating them if it is necessary is more efficient. When we consider a single metric, multicast source routing can be achieved by forwarding packets over the shortest path tree for example. When the overall cost of the tree must be minimized, the problem must be tackled differently. Determining the minimal cost multicast tree for a multicast group corresponds to the Steiner Tree problem which is shown to be NP-complete [11]. An additional dimension to the multicast routing problem is to construct trees or sub-graphs that will satisfy multiple QoS requirements. Routing problem (in the rest of the paper, QoS routing refers to multi-constrained QoS Routing) even in the unicast case is known to be NP-complete problem and has been extensively studied by the research community. [20] gives an overview of the main proposed QoS routing algorithms which try to find a path between a source and a destination node that satisfies a set of constraints. For the multicast case, a number of QoS routing algorithms based on single, dual and multiple metrics have been proposed. Single metric QoS multicast routing algorithms have been proposed for cost [26, 2, 27] and for delay [6, 13]. Dual metric based routing algorithms have been formulated for the following combinations: cost-delay [18, 25, 10] and delay-jitter [24, 27]. For the general case of the multi-constrained multicast routing problem which involves multiple QoS metrics, only one algorithm has been proposed due to the complexity nature of this problem. Multicast Adaptive Multiple Constraints Routing Algorithm (Mamcra) [17] attempts to find multiple QoS constrained paths to the multicast members in an efficient but not always optimal manner. The main idea of Mamcra is to compute multi-constrained shortest paths from the source node to each destination using a unicast QoS routing algorithm, Samcra [22]. The set of obtained paths is then optimized to determine a multicast sub-graph that uses as few links from the first paths set as possible. Mamcra proposes a greedy heuristic approach to solve this second problem. The quality of the approximation isn't proved and the shortcoming of the proposed greedy algorithm can be improved. This paper deals with multiple constrained QoS multicast routing problem which constitutes one of the most interesting problems of multi-objective optimization in network field. In this paper, we will focus, essentially, on optimizing the set of shortest paths to solve the multiple constrained QoS multicast routing problem. The set of paths can be obtained by Samcra or any multiple constrained QoS unicast routing algorithm. Considering the drawbacks of Mamcra's greedy algorithm, we propose ICRA, an improvement version of the greedy algorithm and Taboo-QMR a global optimization based on a meta-heuristic approach, namely on the taboo search, Taboo-QMR provides a solution which can be close to the optimal solution.

This paper is organized as follows. Section 2 specifies the multiple constraint problems for unicast and multicast QoS routing and provide an overview of most proposed approaches to treat these problems. Section 3 presents Mamcra algorithm proposed to solve multiple constraint multicast routing problem. This section emphasizes its weak points and proposes some improvements without great changes. Section 4 proposes a formulation of a new problem aiming to optimize the multicast sub-graph, the OMS problem. Section 5 investigates how the problem of optimizing multicast sub-graph must be tackled if incremental search algorithms are adopted and it proposes ICRA, an incremental algorithm to solve the OMS problem. Section 6 describes how the technique of taboo search can be used to provide a solution to the multiple constraint optimal multicast routing problem . Section 7 presents simulation results.

2 Multi-Constrained Routing Problems

In this section, we will give formal definition of multi-constrained QoS routing problems. We start by the unicast case since existing approach for multiple constraint multicast routing problem resolution is based on the unicast multi-constrained routing solution.

2.1 Unicast QoS routing

QoS routing problem or constraint-based routing consists of finding path from a source node to a destination that satisfies multiple QoS constraints. Since this field is quite mature, we give here a formal definition of the problem and we describe a sample of proposed solutions namely the Self Adaptive Multi-Constraint Routing Algorithm (Samcra) [22] since it is used as a basis for multicast QoS routing algorithm proposed next. Samples of abundant work can be found in [2, 3, 12, 15, 16, 14] and their references.

2.1.1 Unicast QoS Routing Problem Specification

A QoS routing solution involves two components: the routing protocol and the routing algorithm. The objective of the routing protocol is to manage available resources dynamicity. All nodes must have a realistic view of available resources and network utilization of all links. That's why the routing protocol defines the mechanism used to distribute this information called link state information. So, a link-state routing such as in OSPF [30] or PNNI [31] is mandatory to make every node share a map of the network topology and the available resources. Using this link-state information, the routing algorithm computes paths between a source node and a destination node that are within defined constraints or optimizes a certain criterion. The unicast routing algorithms attempt to solve the Multi-Constraint Path (MCP) Problem and/or the Multi-Constraint Optimal Path (MCOP) Problem. In the following, we will first specify some hypothesis used to solve these problems and then we will explain the notation used throughout this section.

Hypothesis 1: Proposed solutions assume that the network-state information (a set of link values) is temporarily static and has been distributed in the network and is accurately maintained at each node using QoS link-state routing protocols.

Hypothesis 2: The most frequently used QoS metrics are categorized into additive and min\max metrics (bottleneck or concave metrics). A QoS metric is additive (e.g., delay, jitter, the logarithm of the probability of successful transmission) when the weight (of that metric) of a path equals to the sum of the QoS weights of the links defining that path. In the case of bottleneck metrics, the weight of a QoS measure of a path is the minimum (maximum) of the QoS weights along the path (e.g., available bandwidth). Constraints on min (max) QoS measures can easily be treated by omitting all links (and possibly disconnected nodes) which do not satisfy the requested min (max) QoS constraints. In contrast, constraints on additive QoS measures cause more difficulties. Hence, without loss of generality, all QoS measures are assumed to be additive.

Hypothesis 3: The network topology is modeled as an undirected graph $G=(V,E)$, where V is the set of nodes and E is the set of links. Each link $(u,v) \in E$ is characterized by m additive QoS metrics. So we associate to the link an m-dimensional link weight vector of m non-negative QoS weights $\vec{w}(u,v) = [w_i(u,v), fori = 1,2,\ldots,m]$. The m QoS constraints (which are the limits of the end-to-end values on the used paths) are represented by the constraint vector $\vec{L} = [L_1, L_2, ..., L_n]$.

Definition 1: Considering a path P of G composed of a set of links, the i^{th} weight w_i of P is defined as:

$$w_i(P) = \sum_{(u,v) \in P} w_i(u,v) \tag{1}$$

. We also define the weight of the path as

$$\vec{w}(P) = \sum_{(u,v) \in P} \vec{w}(u,v) \tag{2}$$

.

Multi-Constraint Path (MCP) problem In this case we consider the problem to find a path P from a source node s to a destination node d such that the QoS constraints are respected:

$$w_i(P) \leq L_i \text{ for } i = 1,2,\ldots m \tag{3}$$

This kind of paths is called feasible path. They may be many feasible paths; it might be interesting to find, from the set of feasible paths, the path minimizing a cost function l(P), l

refers to a length function, it can be any function of the weights w_i provided it obeys to the criteria for length or distance in vector algebra. Such a path is the solution of the MCOP problem which can be defined formally as follows.

Multi-Constraint Optimal Path (MCOP) problem

In this second case, the problem is to find a path P^* from a source node s to a destination node d such that:

$$w_i(P^*) \leq L_i \text{ for } i = 1, 2, \ldots m \tag{4}$$

$$l(P*) \leq l(P) \; \forall \; P, \; P* \text{ satisfying (4) where } l \text{ is a length function} \tag{5}$$

To illustrate the MCOP problem, let's consider the case of finding multi-constrained paths where the cost function l is given by:

$$l(P) = \max_{1 \leq i \leq m} \frac{w_i(P)}{L_i} \tag{6}$$

Note that the solution (or the solutions when there are multiple shortest paths) of the MCOP is not necessary element of the Pareto optimal set [1]. It depends on the length function adopted to evaluate multi-constrained solutions. (Figure 1) illustrates the relation between the set of possible paths and feasible paths in the case of two additive metrics ($m = 2$). The paths are represented in the plan. Each point corresponds to a path represented by its vector $l_i = w_i / L_i$. So the feasible solutions are inside the square *L(1,1)*. In this case, when using the length function defined in (6), the optimal solutions of the MCOP problem are in the perimeter of the minimal length square (indicated with dotted line) which contains at most one element of the Pareto optimal set.

2.1.2 Unicast QoS routing problems resolution

The MCP and MCOP problems are NP-complete [7] and to solve them, heuristic algorithms are needed. [14] gives a survey of most proposed algorithms and compare them. The main idea used by most algorithms is to find out a length function which can be used to scan feasible solutions. So, the problem is simplified to a problem that is solvable by a shortest path algorithm such as Dijkstra or Bellman Ford algorithms [5, 4]. One of the most promising proposed algorithms is the Tunable Accuracy Multiple Constraints Routing Algorithm (Tamcra). To solve MCP problem, Tamcra [23] uses the non linear length combination weight given in (6).

The weight defined in (6), guarantees that P is feasible, if it verifies:

$$l(P) \leq 1 \tag{7}$$

In addition of the non linear weight, Tamcra uses the k-shortest path approach [1]. It is essentially a version of Dijkstra's algorithm that stores for each node more sub-paths

[1] the Pareto Optimal Set is the set of non dominated paths, knowing that a path P_1 is dominated by a path P_2, when $w_i(P_1) \leq w_i(P_2)$ for i=1,...,m

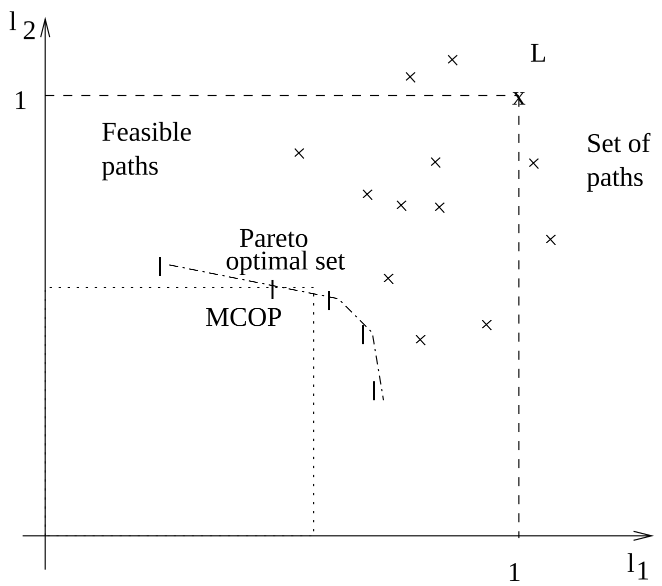

Figure 1: Plan illustration of MCP and MCOP problems (m=2)

than just the shortest one. Then, for each of these paths the path length given by (6) is calculated and the one with minimum length is considered. In Tamcra, k is pre-defined, while the extension of Tamcra algorithm, Self Adaptive Multiple Constraints Routing Algorithm, Samcra [22], controls the value of k self-adaptively. This means that Tamcra is of polynomial complexity, while Samcra is an exact algorithm and its complexity is exponential. The choice of k in Tamcra is a trade-off between performance and complexity. Samcra, on the other hand, guarantees to find a feasible path, if one exists and in that case it corresponds to the shortest path according to the adopted length function. To reduce the complexity of the algorithm, Tamcra and Samcra don't consider dominated paths. Moreover, in [21], the authors demonstrate that Samcra can be improved using a fourth concept of look-ahead. This concept can be considered as an additional mechanism to reduce the search space of possible paths by limiting the set of possible paths using information of the remaining sub-path toward the destination. The authors show in [21] that the incorporation of the look-ahead improvements leads to a gain particularly in large networks. The same authors in [19] evaluate the complexity of QoS routing. They attempt to show that the NP completeness of the MCP problem hinges on four factors, namely the underlying topology, link weights that can grow arbitrarily large or have an infinite granularity, a very negative correlation among the link weights, and the values of the constraints. They argue that in practice, these conditions are unlikely to occur simultaneously and therefore believe that exact QoS routing algorithm such as Samcra can be adopted. However, in [9], the authors assert that exact algorithms can not work in online Traffic Engineering environment (deployment field of QoS routing in practice). They show through simulations that heuristic algorithms are more suitable to such an environment. Such a debate can not be completed and can be much written about.

2.2 Multicast QoS routing problem specification

To specify multicast QoS routing, same hypothesis presented above are adopted. The constraint vector $\vec{L}$ represents the limits allowed for path weights from the source node to every member of the multicast group. It represents the limits for end-to-end values and not for the sum of values in the multicast structure. The constraint vector $\vec{L}$ is assumed to be the same for all multicast members. In the following, we present firstly the exiting problems specified in the literature then we formulate a new problem dealing with multicast QoS routing.

2.2.1 Existing multicast QoS routing problems

Multicast QoS routing problem consists on finding a set of paths from a node source s to p destination nodes d_j $(j = 1, \dots, p)$. In traditional multicast routing this set of paths corresponds to a tree but in multicast QoS routing it is not compulsorily a tree. In the general case, it corresponds to a sub-graph $M = (W, H), M \subseteq G$. M is regarded as a set of paths from s to d_j which use the links in H. The multicast group is given by a source and a destination set $\{s, D = \{d_1, d_2, \dots, d_p\}\} \subseteq W$. Under such hypothesis, three problems were formulated in [17].

Problem I: Multiple Constrained Multicast (MCM)

Given s and D, find $M(W, H)$ such that, for each path $P(s, d_j)$ from s to $d_j \in D(j = 1, \dots, p)$:

$$w_i(P) \leq L_i \text{ for } i = 1, \dots, m \tag{8}$$

Note that, if for a certain $d_j \in D$ no feasible path exists, the problem has not solution. To find a solution for the remaining sub-set of destinations, d_j should be removed from D. Since, the sub-graph or the routing structure solution of the MCM problem must fulfill (8), this structure composed of a set of feasible paths to destinations may contain cycles. These cycles lead to redundancies when the structure is used to achieve multicast routing. That is opposed to the multicast philosophy which aims the efficient use of multicast resources.

Problem II: Multiple Parameter Steiner Tree (MPST)

Given s and D, find $M(W, H), \{s, D\} \subseteq W$ for which $l(M)$ is minimum. l is a cost or length function and it can be the same length function which is used for QoS unicast routing given by 6.

$$l(M) = \max_{i=1,\dots,m}\left(\frac{w_i(M)}{L_i}\right)\text{where } w_i(M) = \sum_{(u,v)\in H} w_i(u, v) \ (i = 1, \dots, m\text{F}) \tag{9}$$

Note that some additive metrics as delay are not additive for multicast sub-graphs. The sum of the link delay values on the paths in a multicast sub-graph M does not characterize the QoS criteria for multicast routing. Only the end-to-end delays in the according paths are interesting. The MPST optimization can be used only in the case of cost-like metrics, when the overall metric describe the communication cost.

Problem III: Multiple Constrained Minimum Weight Multicast (MCMWM)

For s, D given, find $M(W, H)$ such that, for each path $P(s, d_j)$ from s to $d_j \in D (j = 1, \ldots, p)$:

$$w_i(P) \leq L_i \text{ for } i = 1, \ldots, m \text{ and } l(\mathrm{M}) \text{ is minimum.} \tag{10}$$

2.2.2 New proposal for multicast QoS routing Problem

The MCM problem defined previously aims to find a routing structure composed of feasible paths from the source node to each destinations nodes if they exists. The MPST attempts to minimize the presented non-linear length function l throughout a tree whereas the MCMWM minimizes l throughout a feasible routing structure. Minimizing such a function for a tree or routing structure does not match the original multicast philosophy. In fact, real multicast routing politics aims the minimization of allocated network resources. Without QoS criteria, the optimal solution for this problem is the Steiner tree which contains the minimal cost of network links. But, when QoS needs of multicast members must be considered, the goal of a multicast QoS routing approach can be to find a minimal hop count solution fulfilling the given QoS constraints. Respecting these considerations, we propose the formulation of the following optimization problem.

We adopt the same notations, hypothesis and definitions given in previous sections. Let $h(M)$ be the number of hops in the multicast sub-graph M. We introduce a new metric $cl(M)$, the critical length of the multicast sub-graph M. $cl(M)$ is given by:

$$cl(M) = \max_{d_j \in D} l(P(s, d_j)) \tag{11}$$

Problem IV: Multiple Constrained Minimum Length Multicast (MCMLM)

For s, D given, find $M(W, H)$ such that for each path $P(s, d_j)$ from s to $d_j \in D(j = 1, \ldots, p)$:

$$w_i(P) \leq L_i \text{ for } i = 1, \ldots, m \text{ and } h(M) \text{ is minimum.} \tag{12}$$

A feasible sub-graph should satisfy the constraints in all paths. Using the metric $cl(M)$ defined in (11), the MCMWM problem can be formulated as the determination of a multicast sub-graph M satisfying (13)

$$cl(M) \leq 1 \text{ and } l(M) \text{ is minimum.} \tag{13}$$

Using the cl metric, the MCMLM problem that we have just defined corresponds to find *M (V,H)* satisfying (14).

$$cl(M) \leq 1 \text{ and } h(M) \text{ is minimum.} \tag{14}$$

So, to summarize, we can deduce that solving the MCM problem results in satisfying the QoS requirement of multicast members. The MPST problem optimizes only cost functions.

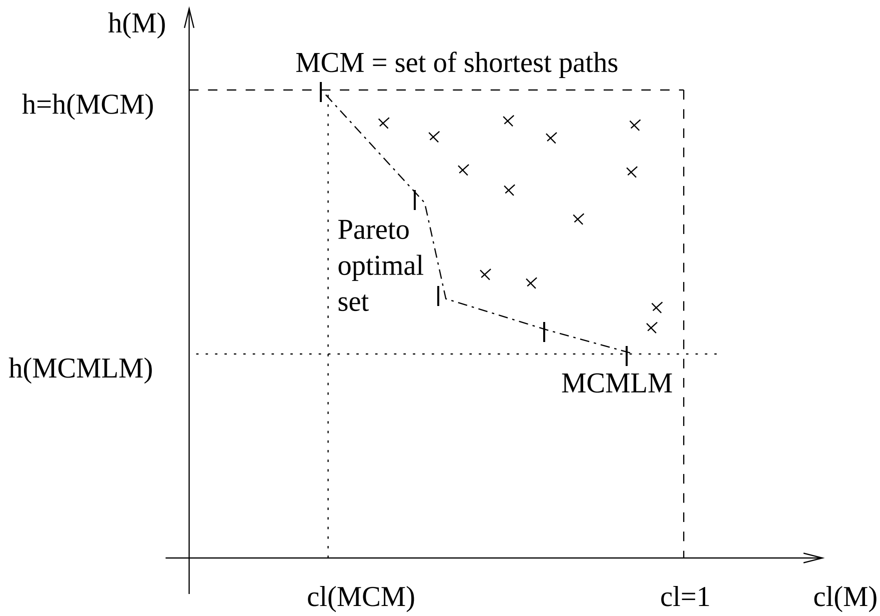

Figure 2: Relation between MCM and MCMLM Problems (m=2 weights)

The MCMWM optimizes the total weight of the multicast sub-graph while satisfying QoS requirements. MCMLM minimizes resource allocations (link utilization) while fulfilling the QoS needs.

To show relationship between the MCM and the MCMLM, we give in Figure 2 a plan illustration of the different solutions of these problems according to the critical length cl and to the hop count h. Solutions depicted by points in this space have different hop count and critical length values. Among these solutions, there are some solutions not dominated by others and constitute, so, the Pareto optimal set.

Let's consider one solution (MCM_S) of MCM problem having the minimal critical length ($cl(MCM_S) \leq cl(M)$ for all solutions M of the routing problem). Such a solution can be found by solving the MCOP for all destinations using a multi-constrained shortest path algorithm. Discovering solutions $MCMLM_S$ for MCMLM problem consists on finding a search algorithm that scans efficiently the research space. We can notice that this research space can be limited to the space containing sub-graphs M having a hop count less then $h(MCM_S)$. Interesting research space is then, limited by the solution MCM_S (corresponding to the minimal value of the critical length) and the solution $MCMLM_S$ (corresponding to the minimal hop count). Trivially these points belongs to the Pareto optimal set in this space (if there are multiple solutions for the MCM problem, one of them is in the Pareto optimal set). We think that the other solutions in the Pareto optimal set are also interesting compromising solutions but in the following we focus on approximated solutions of the MCMLM problem.

2.2.3 Multicast QoS routing problems resolution

In [17], the authors proved that the first three problems are NP-complete. They also show that that the sub-graph M solution of the MCM and MCMWM problems is not necessary a tree but the solution of the MPST is always a tree. In the same way, one can prove that the solution of the MCMLM problem is not necessary a tree. They propose an algorithm which solves exactly the MCM problem and approximates the MCMWM problem. The same algorithm can be considered as an algorithm given an approached solution of the MCMLM problem. This algorithm will be discussed in the next section.

3 Analyse of Mamcra algorithm

3.1 Overview of the Mamcra algorithm

Multiple Adaptive Multiple Constraints Routing Algorithm (Mamcra) provides solutions for MCM problem and can be considered as an algorithm which solve approximately MCMWM problem [17]. Mamcra computes the solution of the multiple constrained multicast routing problem by following two steps. In the first step, the set S of shortest paths from the source node s to all p multicast destination members is computed using the introduced non-linear length function l. The used algorithm corresponds to a lightly modified version of Samcra presented in section (2.1.2). The second step aims the reduction of the resulting sub-graph M such that the overall length function is reduced without violating the constraints. M is not necessary a tree. Even if M corresponds to a tree, this tree is not necessary a minimal length tree. This is due to the greedy approach adopted by the Mamcra reduction step. In order to decrease the network ressources use, our proposition concerns the improvement of the multicast-graph reduction. So we will deal, in the rest of this section, with the optimizing step.

3.1.1 Properties used by Mamcra to reductions

To specify the optimizing procedure, the authors of [17] referenced some properties.
Authors consider that two paths $P_1(s, d_1)$ and $P_2(s, d_2)$ form a cycle, if both paths have two nodes in common. Trivially, the first node in common is the source node s. If the two paths have more than two nodes in common, there is a concatenation of cycles defined by a set of common nodes (nodes x_i such as it is depicted in Figure 3).

The authors proposed the two following properties (property 1 and property 2) wich are proved in [17] and are used to reduce the sub-graph S obtained by the first step of Mamcra.

Property 1: Consider two paths $P_1(s, d_1)$ and $P_2(s, d_2)$ forming a cycle with the common node x that is most hops away from s (Figure 4).

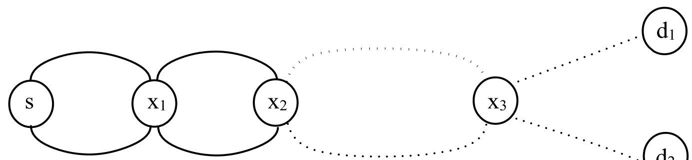

Figure 3: Concatenation of cycles

If $\vec{w}(P_2(s,d_2)) - \vec{w}(P_2(s,x)) + \vec{w}(P_1(s,x)) \leq^d \vec{L}$ then $P_2(s,d_2)$ may be rerouted to $P_1(s,x)P_2(x,d_2)$ without violating the constraints.

Here $P_1(s,x)P_2(x,d_2)$ indicates the concatenation of the two paths and the relation $\leq^d$ corresponds to the Pareto dominance.

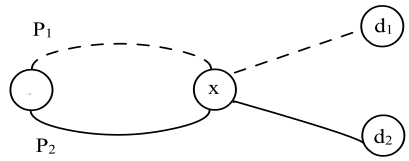

Figure 4: Re-routing path through other links of the multicast sub-graph

This property is used so to remove cycles of the resulting multicast sub-graph. When removing cycles is possible, the total weight vector is reduced.

Property 2 : Given a path $P(s,d)$ within the constraints which contains the sub-path $P(s,a)$, then $P(s,a)$ also lies within the constraints but it is not necessary the shortest path from node s to node a.

Property 2 can be applied to simplify the examination of destination nodes which are within a larger path toward a final destination. Let us suppose that the node a referenced here is a destination of the multicast group. So, the sub-path $P(s,a)$ fulfills the constraints and can be used to reach destination a. If the shortest path for a in S is a path different from $P(s,a)$, it can be removed from S without additional computations.

In the following, we describe the application of these properties in the reduction algorithm of Mamcra.

3.1.2 Mamcra reduction procedure

The goal of the reduction step is to obtain a sub-graph M from the set S by omitting as many cycles as possible without violating the constraints. The procedure is organized in a greedy manner. The paths in the set S are examined one after the other and are added to M. While S is not empty, the algorithm chooses the path which traverses most members.

If more than one maximum member paths are available, the path with smaller length is selected. Then the selected path is added to M after trying some possibilities to eliminate the eventual cycles. This path may form multiple cycles in M as showed in Figure 3. Mamcra first tries to optimize for all cycles on the path, beginning with the larger cycle (s, x_i, s) where the common node x_i is the most hop away from the source and by applying property 1 to eliminate the sub-path (s, x_i) of the newly added path. If this is not possible, the procedure is repeated without examining the last cycle. So, in the next step, only the cycle (s, x_{i-1}, s) is considered. When a cycle can not be removed, an additional constraint on the bandwidth must be checked. In fact, when considering such constraints and when overlaps can not be omitted, some links must be able to provide more bandwidth than the bandwidth required by the source. If n is the number of replicated packets on such a link, the link capacity must be equal or larger than n times the bandwidth required. At the end of the procedure, M contains all members for which feasible paths exist after eliminating some redundancies. This optimizing procedure in Mamcra can be summarized by the meta-code given in Algorithm 1.

Algorithm 1 Step B : Optimizing multi-constrained multicast structure

Reduction Step of Mamcra

Input: The network $G = (N, E)$, a group g with a source s, constraints L_i, the set S of optimal path computed by Samcra
Output: A set M of paths

```
While (S ≠ ∅) do
    add the path with the most members(d_j) to M;
    If (many) then
        choose the one with smallest length
    end If
    If (the added path forms a cycle in M) then
        optimize M by rerouting the new path through an already existing path
        without violating constraints;
    end If
    If (cycle is not removed) then
        Check if the new path does not violate the min/max constraints
    end If
    Remove from S all nodes that are already visited by M;
done
return (M);
```

Reduction step of Mamcra on an example

If we consider the topology presented in Figure 5 and a constraint vector equal to $[20, 20]$, execution of the first step of Mamcra will provide the set S of shortest paths to d_1 and d_2. Here $S = \{(s, b, c, e, d_1), (s, a, c, e, d_2)\}$.

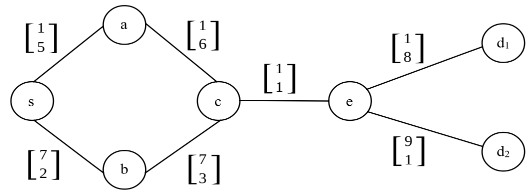

Figure 5: Depending on the constraints, the multi-constraint sub-graph covering the group of source s and of members d_1, d_2 can contain a cycle

During the optimizing step, the sub-graph S is reduced to the sub-graph M. In our case, we obtain $M = \{(s, b, c, e, d_1), (s, b, c, e, d_{,2})\}$. If the constraint limits are $(16, 16)$, the cycle (s, a, c, b, s) can not be removed, so the link (c, e) must have enough resources to transfer duplicated packets, that is explain the bandwidth constraint check described above.

3.2 Shortages of the reduction procedure of Mamcra

Before presenting our proposition, we show through some examples that the optimizing procedure like it is defined is not always efficient for cycle elimination. After showing its weak points,we propose some improvements that can be introduced without great changes in this optimizing procedure.

Problem 1: Mamcra eliminates redundancies only from newly added paths.

The optimization procedure of Mamcra adds the shortest paths from S to the final multicast sub-graph M one at a time. Let us suppose that P_1 is already in M and we are examining the adding of path P_2 to M. Let us suppose that adding P_2 implies cycles as it is illustrated in Figure 6. Mamcra propose the application of Property 1 to study rerouting P_2 through P_1. In our case, the sub-path (a, b, c) can be removed, if and only if, $\vec{w}(P_2(s, d_2)) - \vec{w}(P_2(s, c)) + \vec{w}(P_1(s, c)) \leq^d \vec{L}$. But, when having such cycles, two possibilities must be studied, Mamcra does not examine the possibility of removing (a, d, c), only rerouting new added path through already existing ones is allowed. So, if rerouting P_2 through P_1 violates the constraints, the cycle is not eliminated. But the fact that redundant links in the newly added path can not be removed does not imply that redundant links in the old path P_1 can not be removed. Trivially, the cycle (a, d, c) can be removed, if $\vec{w}(P_1(s, d_1)) - \vec{w}(P_1(s, c)) + \vec{w}(P_2(s, c)) \leq^d \vec{L}$. Studying the two cases can enhance considerably the overall performance of the reduction procedure: it can make some link eliminations possible and even if the two sub-paths can be deleted, the one that minimizes the length function must be chosen.

Problem 2 : Mamcra does not handle clearly some cycles.

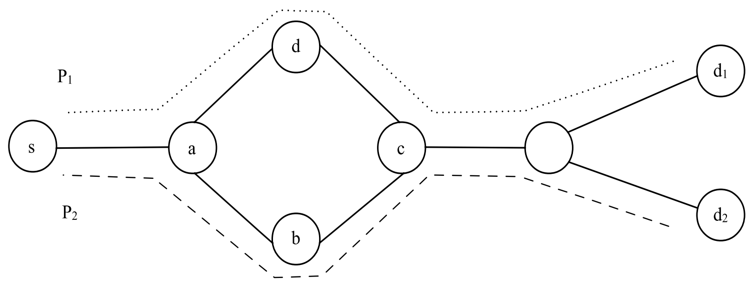

Figure 6: Union of two paths with one cycle

Mamcra proposes the computation of shortest paths from the source to the destinations. As Figure 7 illustrates it, two shortest paths can use the same links (edges) in the opposite direction. In this example, the shortest paths from the source s to the destinations d_1 and d_2 are the paths $P(s,b,a,d_1)$ and $P(s,a,b,d_2)$ respectively. They use the edge (a,b) in the opposite direction. When applying Mamcra algorithm, the larger cycle should be analyzed with the farthest common node x_i. In this particular case, there is only one cycle but two common nodes (a and b) between the paths and outside of the source node. Which node should be considered to reduction? Let us suppose that the constraint vector is $\vec{L} = [10,10]$. If the node a is examined at first, then the reduction is possible, since $P(s,\vec{a},d_1) = [2,9] \leq^d \vec{L}$. If the node b is examined at first, then there is no reduction. The path $P(s,b,d_2)$ violates the constraints because $P(s,\vec{b},d_2) = [11,2]$.

To use the Mamcra greedy algorithm, a precision is needed. For example, the algorithm may choose always the farthest common node from the source on the newly added path.

Similarly to Problem 1., as our example illustrate the case, the selection of the farthest node on the newly added path does not correspond necessarily to the best (or feasible) reduction.

Problem 3: Mamcra does not handle all destination nodes inside the cycles.

A second problem not clearly treated in Mamcra algorithm is raised when sub-paths candidate to the deletion contain destination members. To expose this problem, let's consider the example in Figure 6. Suppose that node d belongs to the multicast group. So, when adding the path P_1 and after applying Property 2 of Mamcra procedure, path in the set S from source node s to node d must be removed. Therefore, the only way to reach d from s is through path P_1. When adding P_2 and in the case of rerouting P_1 through P_2, the removed sub-path corresponds to (a,d,c). Removing such a sub-path isn't possible as it is used to reach node d. Only the part (d,c) of this sub-path can be omitted. The reduction possibilities which must be studied and compared are: deletion of (a,b,c) or deletion of (d,c), depending on whether Property 1 is respected or not.

Problem 4: The selection order of examined paths influences the resulting structure.

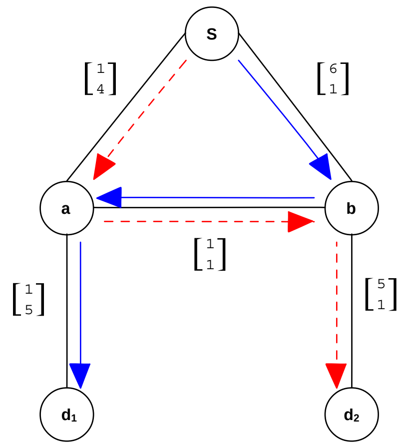

Figure 7: Two shortest paths use the same edge in the opposite direction

Another problem of the cycle elimination procedure of Mamcra is that the order of examined paths can influence the efficiency of the procedure. More precisely, the length of the finally found structure and the number of links in this structure depend on the selection order of paths in S.

Even if the selection respects the order defined in Mamcra, in cases when multicast destination number on the path and path length are the same for several paths, the selection order influences the obtained result. To illustrate, let us suppose that two paths P_1 and P_2 contain the same number of destination nodes (for example only one) and there are the same length too. Such an example is illustrated in Figure 8. Let us suppose that the constraint vector is $\vec{L} = [12, 12]$ in this case and that Mamcra chooses the path P_1 at first to add to M. When the add of P_2 is examined, since $\vec{w}(P_1(s, x_1)) + \vec{w}(P_2(x_1, d_2)) = [13, 7]$, the concatenation of $P_1(s, x_1)$ and $P_2(x_1, d_2)$ does not correspond to a feasible path. So the remove of $P_2(s, x_1)$ is not possible and M contains both P_1 and P_2. If Mamcra chooses P_2 at first, and the add of P_1 is examined at a second time, one can state that $\vec{w}(P_2(s, x_1)) + \vec{w}(P_1(x_1, d_1)) = [5, 11]$, and $P_1(s, x_1)$ can be removed.

This problem effects the resulting structure not only when examining cycle elimination of the two considered paths but also the elimination of all the cycles involved by all the shortest paths. To highlight this influence, let us consider the example given in Figure 9 where 3 paths (P_1, P_2, P_3) of identical length and with the same number of members are considered. We propose to compare two scenarii. In the first case, let us suppose that the selection order is P_1, P_2 and P_3. Let us suppose that $\vec{w}(P_2(s, d_2)) - \vec{w}(P_2(s, x_1)) + \vec{w}(P_1(s, x_1)) >^d \vec{L}$ and so cycle elimination is not possible when P_2 is added to P_1. At a second time, P_3 is added to M. Let us suppose that $\vec{w}(P_1(s, d_1)) - \vec{w}(P_1(s, x_2)) + \vec{w}(P_3(s, x_2)) \leq^d \vec{L}$ and

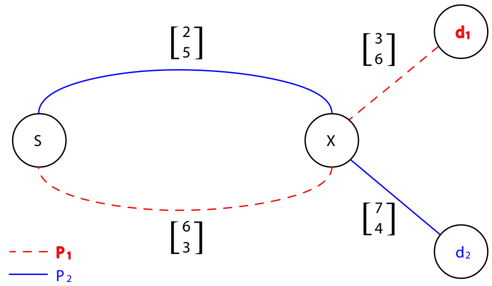

Figure 8: Case of two paths with the same condition for selection

$\vec{w}(P_3(s,d_3)) - \vec{w}(P_3(s,x_2)) + \vec{w}(P_1(s,x_2)) \leq^d \vec{L}$. Then $P_3(s,x_2)$ can be removed from the structure.

In the second scenario, let us suppose that the selection order corresponds to P_3, P_1 and P_2 then there is a reduction when P_1 is added to P_3 as $P_1(s,x_2)$ can be removed from the structure and d_1 is reached through the path $P_3(s,x_2)P_1(x_2,d_1)$. When adding P_2, a cycle (s,x_2,x_1,s) appears and the application of Property 1 can determine which part of the cycle can be removed. Let us suppose, for this case, that $\vec{w}(P_2(s,d_2)) - \vec{w}(P_2(s,x_1)) + \vec{w}(P_3(s,x_2)) + \vec{w}(P_1(x_2,x_1)) \leq^d \vec{L}$ and so $P_2(s,x_1)$ can be removed. We can notice that this reduction was not possible when P_2 was compared to P_1 and that's why the cycle can not be removed in the first scenario.

Generally, we can state that the selection order of paths in S influences the result when Property 1 and Property 2 are used to achieve reduction of the multi-constrained routing structure. Adding paths and removing redundancies incrementally may prevent some removal that can be achieved if other paths have been added.

3.3 Summary of our analysis

The second step of Mamcra aims to reduce the set of multi-constraint paths computed from the source node s to the p destination nodes in order to provide a feasible solution of MCM problem and an approximated solution to MCMWM and MCMLM problems. The greedy algorithm proposed to eliminate cycles presents some drawbacks that can be resumed as follows.

- The first problem is raised when cycles are removed. In Mamcra, only one possibility is checked to eliminate a cycle.
- The second problem is raised when some links are used by different paths in the opposite direction. The criterion choice of common nodes in the path reduction algorithm should be specified and this criterion influences the reduced structure.

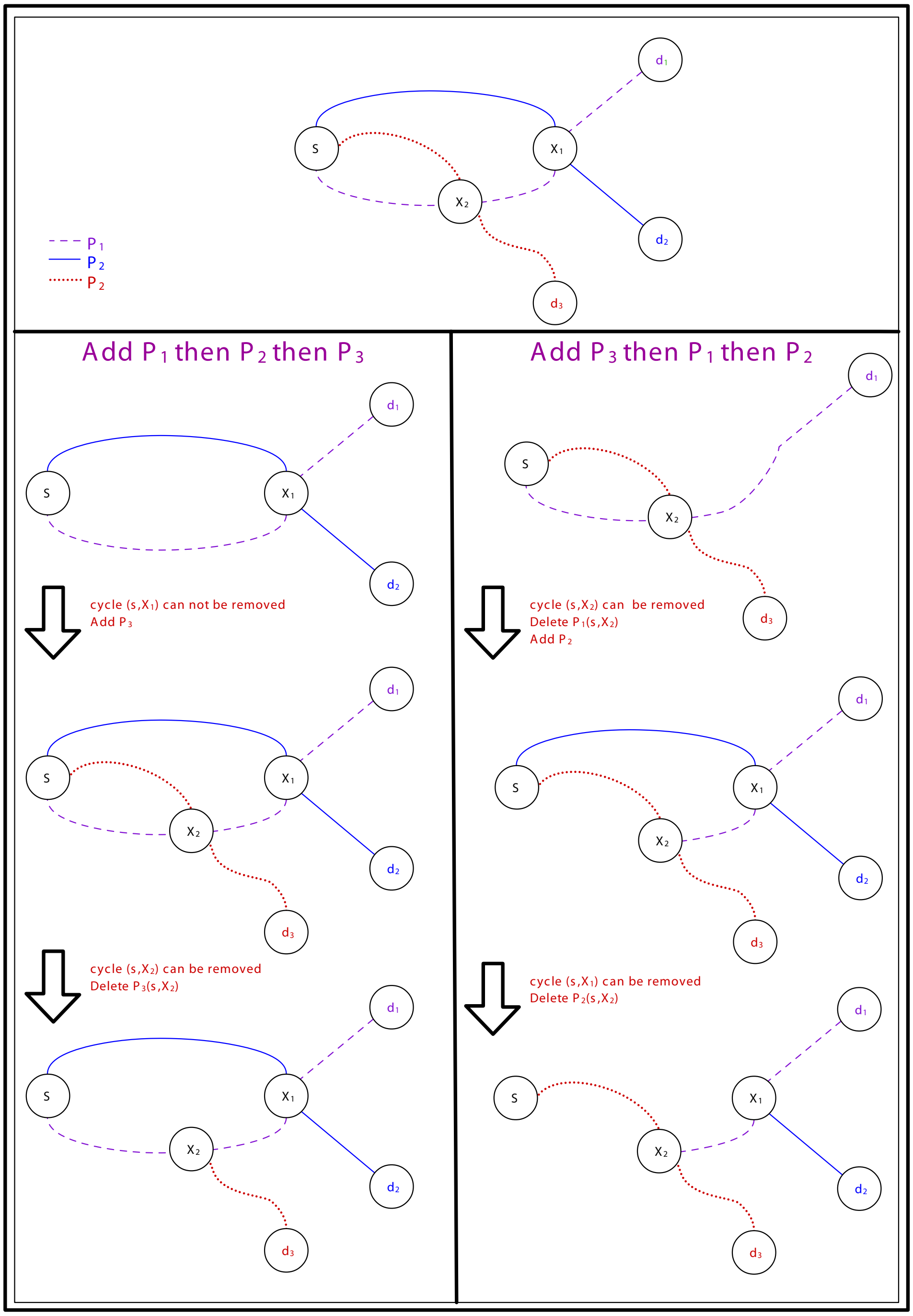

Figure 9: A simple case of three paths to add to M

- The third problem is the eventual existence of intermediate destination nodes in a path. Generally, only a part of the sub paths composing a cycle can be removed not to affect the reception of the intermediate nodes. Multiple possibilities must be examined.
- The last problem is the order adopted to remove cycles influences the number of cycles removed. That is due to the incremental approach of Mamcra used in removing cycles.

The first three problems can be easily resolved by making little modification to Mamcra procedure without changing Mamcra principles. But to tackle the last problem, the approach must be reviewed wholly and a global optimization procedure is needed.

Before proposing algorithms improving Mamcra reduction procedure, we must first formulate the problem of optimizing multicast sub-graph.

4 Optimizing Multicast Sub-graph (OMS) Problem

As it is presented in the previous section, Mamcra computes a set of feasible paths to the destinations of a given multicast group. This set can contain redundancies and the diminution of the set of used links is interesting to spare network resources. The second step of Mamcra executes a simple greedy algorithm to diminish the set of used links and so it is not optimal. Eliminating redundancies from the multi-constraint multicast routing structure can be formulated as an optimization problem.

In this section, we propose a formulation of the Optimal Multicast Sub-Graph Problem. As it will be discussed in the following, the problem is NP-complete, so we analyze different approaches to solve approximately this problem.

4.1 Optimal Multicast Sub-Graph (OMS) Problem: Problem formulation

Generally, let us suppose that a set of feasible paths to the destinations of a multicast group is available (for example: the set of "shortest paths" computed by Samcra). Our goal is the global optimization of this set: remove the more redundancies without violating the QoS constraints.

Optimal Multicast Sub-Graph (OMS) Problem:

Let us suppose that, for a source $s \in V_S$ and a set of destinations $D = \{d_i \in V_S, i = 1, ..., p\}$, a graph $G_S = (V_S, E_S)$ is given such as the union of feasible paths from the source to each destination. The goal is to find M a sub-graph of G_S optimizing (minimizing) an objective function $f(M)$. This objective function can be either the length function $l(M)$ specified in the MCMWM or the number of hops $h(M)$ defined for the MCMLM.

In this last case, ***the OMS problem consists to find*** $M \subseteq G_S$ ***such that*** $h(M)$ ***is minimal and there is a feasible path in*** M ***to each destination of*** D.

We can notice that the optimal solution of OMS problem does not correspond necessarily to the solution of the problems MCMWM or MCMLM even if the sub-graph G_S is created on the base of the shortest paths. To illustrate that, let the topology given in Figure 10 be used. If the shortest paths are computed from the source s to the destinations d_1 and d_2 by the unicast QoS routing algorithm then the sub-graph $G_S = \{P_1 = (s, a, d_1), P_2 = (s, c, d_2)\}$ is obtained as the set of feasible paths. The shortest multicast structure (solution of the MCMWM and also the MCMLM problem in this case) corresponds to the tree using the central node b. This tree is outside of the graph G_S, so it can not be found by any reduction from G_S.

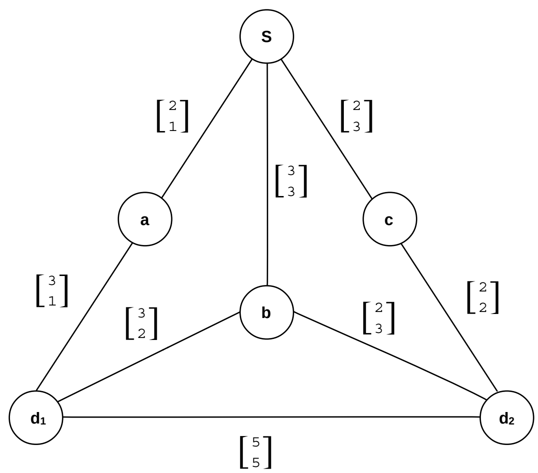

Figure 10: Shortest paths structure vs Shortest multicast structure

In our analysis, the start point is an arbitrary set S of feasible paths and the involved graph G_S. So, our objective can not be to find the optimal solution defined in the MCMWM and MCMLM problems but only finding the sub-graph with minimal hop count value.

To solve the OMS problem efficiently, some reductions are possible and we propose to adopt some definitions as follows.

A node $n \in G_S$ is a *significant node* if it is the source node or a destination node or a branching node (having a degree more than 2).

A *segment* is a path connecting two neighboring significant nodes of G_S. A segment is called *articulation segment*, if its deletion increase the number of connected components in the graph (the articulations are not redundant).

For example, we can consider the example in Figure 11 as a graph in which only the significant nodes are represented. (s, x_1) and (x_1, x_2) are segments containing intermediate nodes or not. (x_0, d_3), (d_5, d_2) and (x_3, d_5) are articulation segments in this graph.

Reduction 1: A link belongs to the optimal solution of the OMS problem if and only if the segment containing it is in the solution.

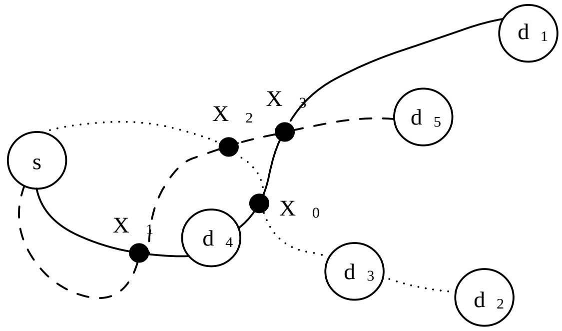

Figure 11: Decomposition of a set of paths in segments

So the set S of paths that we aim to optimize can be considered as a set of segments. Thus, optimizing multicast sub-graph consists of determining the set of segments of G_S that should be removed. In this way, a (probably reduced) graph G'_S containing the segments of G_S can be examined to solve the problem.

Another reduction of the search space is possible by detecting the segments which belong to all solutions. These segments can never be removed. For example, if we refer to the topology of Figure 11, the segments (x_0, d_3), (d_3, d_2), (x_3, d_5) and (x_3, d_1) can not be removed.

Reduction 2: The articulation segments of G'_S can not be removed and belong to all solutions.

Now, by removing from G'_S all articulation segments, the search space can be reduced to a graph G''_S. The solution M should be found by combining the remained segments of G''_S.

From the point of view of its complexity, the OMS problem is equivalent to the original MCMWM and MCMLM problem. find the optimal sub-graph of reduced graph G''_S (instead of S) spanning the multicast group with respect to the QoS constraints. In the worst case the sub-graph G_S corresponds to the whole graph G and if reductions are not possible, $G''_S = G'_S = G_S = G$. In this case OMS corresponds exactly to the original MCMWM/MCMLM problem which is NP-complete [17]. The interest of the reductions that generally the size of the search space can be reduced considerably.

To find the optimal solution of the OMS problem, exhaustive search algorithms can be imagined (enumeration of each combination of redundant segments for example). Let us consider the problem of the selection order when examining cycle elimination (denoted as problem 4 in Section 3.2). This problem can not be avoided when adopting an incremental greedy approach where each (local) decision is definitive. An algorithm which treats globally the redundancy problem is the only way to find the optimal solution. For network utilization purposes, limited execution time algorithms are needed. Improvement of the Mamcra algorithm, heuristic and metaheuristic search algorithms with limited execution time are candidate to find good solutions (with less redundancies) for the OMS problem. That's why in the next sections, we analyze, first, how the OMS problem can be handled in an incremental manner and then, we propose ICRA, a greedy algorithm that enhances some of the Mamcra drawbacks detailed above. The OMS problem is then tackled differently by proposing a metaheuristic based algorithm Taboo-QMR which will be described in section 6.

5 Incremental Cycle Reduction

Solving the OMS problem consists on finding a set of links from the multicast subgraph G''_S to be omitted while fulfilling the members requirements. In this section, cycle reduction will be considered in an incremental and greedy manner by adding the feasible paths of S one after the other to the multicast routing structure M already examined.

5.1 Cycle reduction when a new path is added to the routing structure

Contrarily to Mamcra algorithm, in our proposition the reduction is analyzed and executed regarding all the existing paths, with which the new path forms cycles. So, the reduction procedure is realized with a unique scan of the new path from the farthest common node to the one which is closest to the source. According to the presented reductions and to simplify the analysis, the paths in M and the new path denoted P_n are decomposed in segments. Each segment r is characterized by two information: the weight $\vec{w}(r)$ and the number of links (the hop count) $h(r)$. The new added path P_n forms a cycle in M, if it exists a set of paths P_o^i in M having common nodes with P_n other then the source node. Let $X = \{x_i, i = 1, ...\}$ be this set of common nodes between P_n and the paths already in M (other then the source node).

To simplify the analysis of the elimination of a given cycle $P_n(s, x_i)P_o^i(x_i, s)$, we propose some contractions.

- All the segments of the new path following the node x_i can be contracted in a single segment $r_{P_n^{i+1}}$ and the weight vector and hop count metric of this contracted segment correspond to the sum of weight vectors and hop counts of the concerned segments. In other worlds: to decide if a reduction is possible or not in x_i, only the farthest destination should be considered.
- In the same way, the sub-path of P_o^i rooted at x_i can be contracted in a single segment to analyze the possible reductions. The weight represent the weight of the critical path from x_i to the farthest destination on P_o^i and the hop count corresponds to the sum of hop counts in the sub-path.

Figure 12 illustrates the contraction in a simple case.

To enhance cycle reduction procedure, we propose to extend Property 1 (see section 3.1.1). A new added path P_n may have common nodes with many old paths in M. To examine cycle elimination, we study here three cases. The first case occurs when the examined common

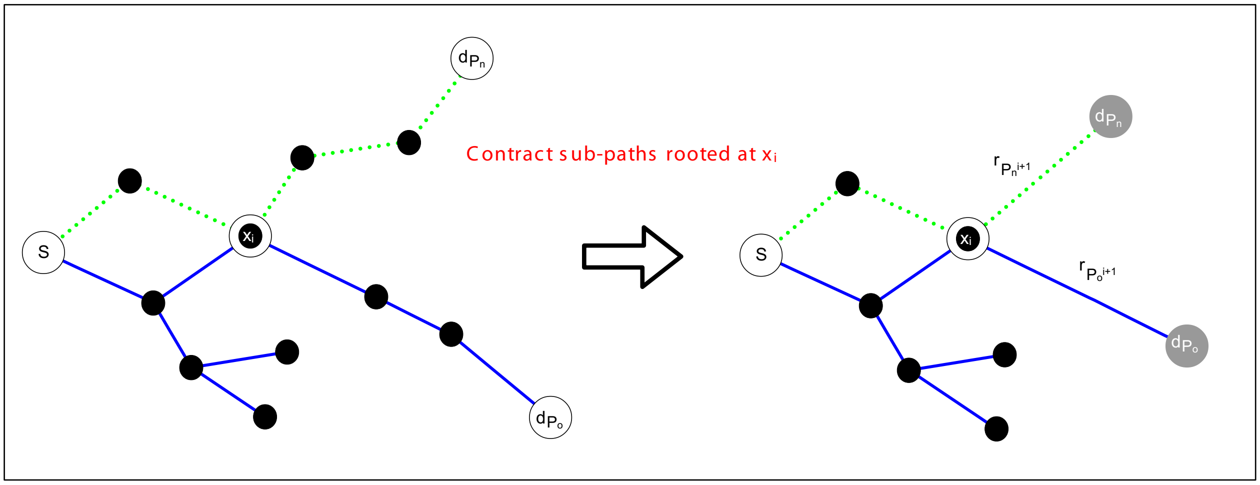

Figure 12: Contraction of components in the general reduction case

node x_i is in P_n and in only one old path. The seconde case is when x_i is in P_n and in at least two other old paths. The last case is when 2 common nodes in P_n are within an old paths in inverted order. We will study each case separately and we will propose an approach involving all these cases.

Case 1: New Path P_n has a common node with only 1 old path P_o

This case corresponds to the more frequent case. Without loss of generality, let us suppose that the two paths P_n and P_o form a cycle between the source s and another common node x_i. Let n_o and n_n be the last significant node before x_i, respectively, in the old path P_o and in the new one P_n . Eliminating the cycle $P_n(s, x_i)P_o^i(s, x_i)$ can be achieved by deleting one of the two segments $P_n(n_n, x_i)$ or $P_o^i(n_o, x_i)$ which precedes x_i respectively on P_n and on P_o^i. The segment $P_n(n_n, x_i)$ can be eliminated, if

$$\vec{w}(P_o^i(s, x_i)) + \vec{w}(r_{P_n^{i+1}}) \leq^d \vec{L}$$

Similarly, the segment $P_o^i(n_o, x_i)$ can be eliminated, if

$$\vec{w}(P_n(s, x_i)) + \vec{w}(r_{P_o^{i+1}}) \leq^d \vec{L}$$

If the two conditions are verified simultaneously and the two segments are candidates to the elimination, then the segment corresponding to a higher gain (for example, the segment with higher hop count) should be deleted. To illustrate such a case, let's consider the example presented in Figure 13. If P_o should be rerouted through P_n to eliminate cycle in x_i, only the segment preceding x_i can be removed to ensure forwarding data to nodes n_1, n_2

and n_3. We recall that these nodes can not be disconnected as they are whether destination nodes or branching nodes which forward data to other destination nodes. In the other case, eliminating the whole sub-path (s, x_i) from P_n is allowed even if nodes such as n_4 and n_5 exist. In fact, if these nodes are destination nodes, paths for these destinations exist in G_S as removing such paths from the initial graph G_S occurs only after adding the new path P_n.

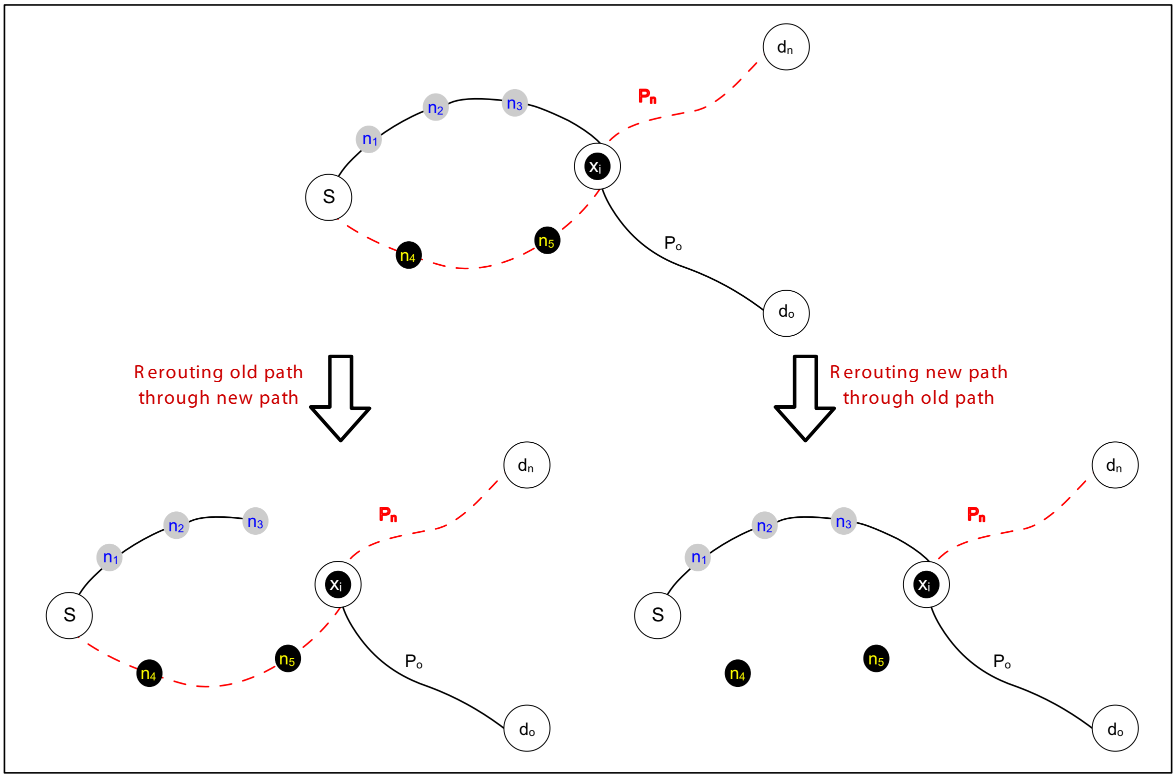

Figure 13: Cycle reduction when sub-paths contains intermediate destination(s)

Case 2: New Path P_n has a common node with a set of old paths
In a more general case, a cycle involves more than one old path. This case occurs when the common node is in the newly added path and in more than one old paths as it is depicted in Figure 14.

In this case, cycle reduction can be achieved if :

- rerouting the newly added path through one of the existing paths is possible due to a common node x_i: the sub-path $P_n(s, x_i)$ is eliminated and one path from the old paths can be used to forward data to the segment $r_{P_n}^{i+1}$.

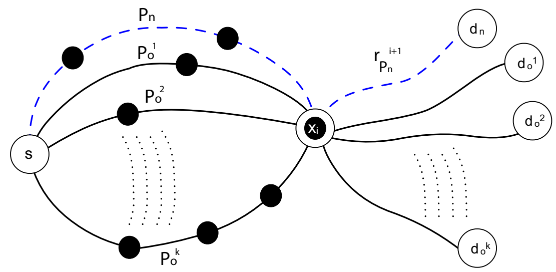

Figure 14: Contraction of components in the general reduction case

- rerouting some of the old paths through the newly added path. In that case, some of the segments preceding x_i and belonging to a set of old paths can be omitted. In the best case, all the old paths can be rerouted through P_n.

We should notice that the already exiting cycles formed by the old paths can not be eliminated without the adding of the new path. In fact, when adding these paths all elimination procedures have been tested and no elimination was possible as the cycles persist.

So, to achieve cycle reduction, whole sub-paths or a set of segments can be candidate to elimination. When more than one candidate to elimination exists, the candidate that optimizes an objective function g is deleted. Basically, for g specification, there are two extreme possibilities: minimize the number of links used for data forwarding or maximize the quality at the destination nodes. In order to minimize the number of used links, we propose the elimination of the segment with larger hop count. In the case of equality, the QoS criteria is discussed and the solution which guarantees the smaller critical value for QoS criteria is chosen. This decision can be seen as a multi-constrained optimization problem in the plan hop distance and critical length value. Other selection criteria can be also proposed but this is out of scope of our analysis.

Case 3: New Path P_n has 2 common nodes with at least one old path in inverted order

Another issue that must be not neglected is the order of considered common nodes in the old and in the new path. In some rare but possible cases, the order of common nodes in the old paths in M can be different from their order in the new path. It can be examined by detecting the position of the node x_i in the two paths compared to the preceding common nodes. In fact, two nodes may belong to two paths but they are used in inverted order. That is the case of nodes a and b in Figure 15: a and b are two common nodes of P_1 and P_2 but they are used from b to a in P_1 and from a to b in P_2. This special case has already been emphasized in Problem 2 in section 3.2. So, when considering 2 common nodes x_i and x_{i-1}, the order adopted here is the order of these nodes in the new path P_n.

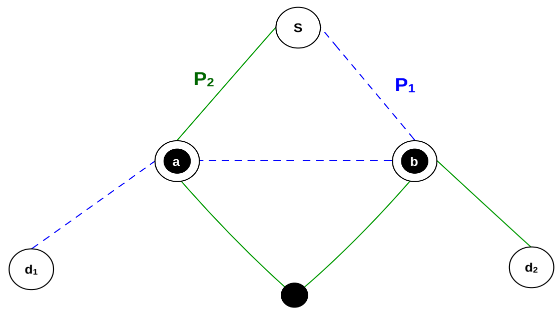

Figure 15: Decomposition of a set of paths in segments

Elimination strategy must involve all the above presented cases. In the following, we propose a new algorithm enhancing Mamcra mechanisms and considering all topology cases.

5.2 Improved Cycle Reduction Algorithm

Based on the preceding exploration of cycle reduction, we propose ICRA, Improved Cycle Reduction Algorithm which brings some improvements to Mamcra algorithm for optimizing QoS multicast routing structure.

ICRA algorithm supposes that a feasible path is known from the source node to each destination of the multicast group. The set S of these paths forms the graph G_S which can contain cycles. ICRA algorithms aims to eliminate redundancies from this set of QoS constrained paths. The result of the elimination procedure is a set of paths (a set of multicast trees) connecting the destinations to the source with less redundant links. Moreover, the algorithm must guarantee the satisfaction of the QoS constraints: the set of paths, result of the elimination procedure, must contain a feasible path for all destination nodes for which a feasible path exists in the initial set. In other worlds, the goal is to decrease the number of links in the resulting structure respecting the given constraints: find an approached solution of the MCMLM problem.

As for Mamcra algorithm, the inputs of the algorithm are the following. We suppose that the QoS constraints vector $\vec{L}$ is given and the m dimensional link value vectors are known for each link. Each path $P_k \in S$ is a feasible path from the source s to a destination $d_k \in D$ for $k = 1, ..., d$.

The output of our algorithm is a set M containing paths routed at s. Each path covers a sub-set of destination nodes and the union of the paths covers the totality of destinations. The result contain only feasible paths from the source node to the covered destinations as it will be done by the following algorithm.

Similarly to Mamcra, the algorithm ICRA works in a greedy manner and proceeds incrementally in two phases: first, selecting a path form S then attempting to add it to the final routing structure while removing as much redundancies as possible. These phases are detailed in the following.

5.2.1 Path selection

To process the input set S of paths, we propose the same procedure of selection as it is defined in Mamcra. For each path P_k, the end to end QoS length $l(P_k)$ and the destination nodes on the path are known. The selection of the next path to treat is based on the number of multicast members present in the path. Paths with large number of multicast members are examined at first. This implies that the selected path will cover as many destinations as possible. In the case of equality on this metric, the algorithm chooses the path with smallest QoS length. So, at a first time the set of paths grows with shortest paths and a short "kernel" is created to facilitate future path additions. If there is an equality on the two mentioned metrics, we propose to select randomly one of the equal paths. Even if the selection influences the result of the reduction, examining all possible orders is expensive that's why we propose the random selection. Algorithm 2 gives the meta-code of ICRA algorithm and details the path selection step.

Algorithm 2 Steps of ICRA (Improved Cycle Reduction Algorithm)

ICRA Algorithm

Input: The valuated network topology graph $G = (N, E)$, a group g with a source s, constraints L_i, the set S of feasible paths computed by a Unicast QoS routing algorithm (after the eventual reductions)

Output: A set M of paths

While $(S \neq \emptyset)$ **do**

Path Selection

Select the path P_n with the maximal number of members (d_j);
If (many) **then**
choose P_n the one with smallest length
end If
If (many) **then**
choose P_n randomly;
end If

Add P_n to the multicast sub-graph M

Adding-new-path(P_n,M);//see Algorithm 4
Remove from S all paths for destinations newly added to M;// return(M);
done

Let us suppose that a path P_n is selected from the set S. In the following, we discuss the addition of this path to the existing set M of paths.

5.2.2 Add a path to the set of multicast paths

The principal steps of path addition to the multicast sub-graph are as follows:

- At first, the common nodes $X = \{x_i, i = 1, ...\}$ between P_n and the paths already in M are determined, but the source node s is excluded from X.
- If there is no intersection, then the path P_n is added as a new path to the set M. In this case, the other paths from the source to the destination nodes in P_n can be deleted from S if they exist (using Property 2 given in section 3.1.1)

- If the path P_n forms cycles with some paths in M, then examining redundancies is achieved in a greedy manner. We propose a cycle reduction procedure based on selecting from a set of determined candidate segments which candidate to eliminate.

Computing the set of segments candidate to elimination. In Mamcra algorithm, removing cycles is processed by examining cycles formed between the new added path and older paths one at a time. For exhaustive examination, older paths are selected one after the other using the adding order. In ICRA algorithm, the reduction is analyzed and executed regarding all the existing paths, with which the new path forms cycles. Common nodes are scanned from the farthest common node to the one which is closest to the source according to the order in the new path. The first step of the cycle reduction procedure is to determine the set of segments candidate to the elimination. Investigation made in the previous section leads us to state the following reduction rules that must be used to determine segments candidate to the elimination.

Reduction Rule 1: A sub-path or segment s_{p^i} can be eliminated from a path P_k existing between s and d_k and it can be replaced by a sub-path s_{p^j} if the replacement results a continual path from s to d_k and if

$$\vec{w}(P_k) - \vec{w}(s_{p^i}) + \vec{w}(s_{p^j}) \leq^d \vec{L}$$

Reduction Rule 2: When a sub-path from source node of an old path in M is candidate to elimination, only the segment preceding the currently examined common node can be removed.

This last reduction rule is used to preserve intermediate significant nodes from deletion. In fact, when the old path is subject to elimination, only the segment preceding the common nodes can be omitted not to disconnect intermediate destination and significant nodes as it is explained in section 5.1.

Selection of segments that should be eliminated. In the following, each candidate to elimination is assimilated to a set of segments. If there is more than one candidate to elimination, the one that optimizes an objective function g is chosen. In ICRA algorithm, accordingly to MCMLM objective, we propose to define the function g as the simple sum of the hop numbers of the eliminated segments. This sum corresponds to the gain of the elimination and our objective is to maximize it.

Definition : Given a new path P_n and an old path P_o having a set of common nodes $X = \{x_i, i = 1, ...\}$, the elimination gain induced by the removal of a set of segments r_P^i from P_n or P_o when examining the common node x_i is

$$g_P = \begin{cases} g_{P_n} = h(r_P^i), & \text{if } r_P^i \text{ is in } P_n; \\ g_{P_o} & \text{if } r_P^i \text{ is in } P_o \end{cases}$$

and

$$g_{P_o} = \begin{cases} h(r_P^i), & \text{if } x_i \text{ and } x_{i-1} \text{ are in the same order in } P_o \text{ and } P_n; \\ h(r_P^i) + h(r_P^{i-1}) & \text{if } x_i \text{ and } x_{i-1} \text{ are in inverted order in } P_o \text{ and } P_n. \end{cases} \quad (15)$$

where r_P^j is the segment or the set of segments preceding x_j.

It is important to notice, that in (15) when r_P^j corresponds to a set of segments (case of elimination of a set of segments from P_o), $h(r_P^j)$ corresponds to the sum of the number of hops of each segments of the considered set of segments.

To better understand the gain definition, let us consider Figure where Case A shows that eliminating $r_{P_n^i}$ from P_n saves the use of $h(r_{P_n^i})$ hops. In the same way, eliminating $r_{P_o^i}$ from P_o induces a gain of $h(r_{P_o^i})$ hops, if the common nodes in P_n and in P_o have the same order in the two paths as it is in the Case B of the figure. Let us notice that the elimination extends on the whole sub-path in the new path, from the source to the last common node where the two reduction rules are respected, while the elimination concerns only the last segment (if the condition is verified for this elimination) on the old path. The reason is simple: the intermediate destination nodes on the new path can be reached using their own existing feasible paths in S. These paths will be examined later in the procedure. Regarding the intermediate destination present on the old path, there is no more path in S to reach them. So, only the last segment before the examined common node candidates to be eliminated.

The last case occurs when a segment $r_{P_o^i}$ of the old path P_o is candidate to elimination and when P_o and P_n have at least two common nodes (for example: x_i and x_{i-1}) in inverted order as depicted in Case C of Figure 16. Elimination of segments $r_{P_o^i}$ implies the elimination of the last segment $r_{P_o^{i-1}}$ preceding the node x_{i-1} as traffic to node d_{p_o} will be rerouted through P_n. In this case, chosen to eliminate the segments on the old path saves $h(r_{P_o^i}) + h(r_{P_o^{i-1}})$ hops. In general case, several common nodes can be in inverse order on the two paths. It is easy to show that the elimination of the last segment on the old path at a common node x_i implicates the elimination of all the last segments which are before the other common nodes going from x_i to the last destination on the old path.

Finally, let us consider the general case where the examined common node x_i belongs to a set of old paths $PO = P_o^i, i = 1, ...$ This set can be partitioned in 2 sets PO_{in} and PO_{out}. PO_{in} contains old paths having common nodes in the same order as in the new path. Unlike this, PO_{out} contains paths having some common nodes in inverted order compared to their order in P_n. The gain associated to the different elimination possibilities can be computed as follows:

- If rerouting P_n through one of the existing paths in PO is feasible, then the gain corresponds to the total length of the sub-path of P_n from the source to x_i and it is given by

$$g_{P_n} = h(r_{P_n^i})$$

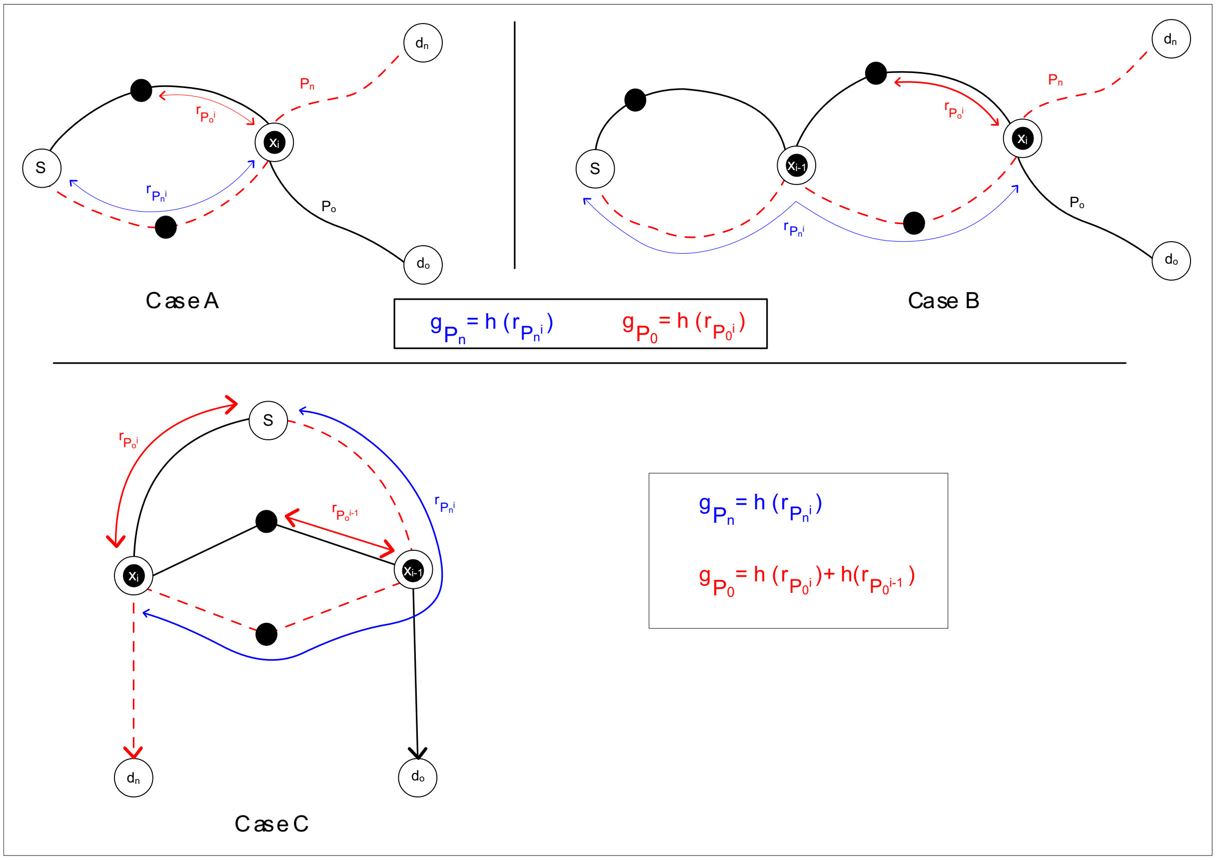

Figure 16: Computing gain associated to segment elimination

- If rerouting a set of old paths PR through P_n which is conserved, the gain is

$$g_{PR} = \sum_{P \in (PR \cap PO_{in})} g_P + \sum_{P \in (PR \cap PO_{out})} g_P$$

Reduction rule 3: To decide whether candidate to eliminate, compute the gain g associated to each candidate then chose the elimination that maximizes the gain.

The selection procedure is detailed with the Algorithm 3.

Algorithm 3 Segment elimination maximizing the gain

Selecting segments to eliminate

Input: A new path P_n, a set of common node X, a node x_i, a set PR of paths that can be rerouted through P_n to remove redundancies in x_i, g_{P_n} the gain induced by rerouting P_n through existing old paths, g_{PR} the gain induced by rerouting the paths in PR through P_n

Output: Paths P_n and paths in PR

If (g_{P_n}¿0) **then**
    If (g_{PR}¿0) **then**
        If ($g_{P_n} \geq g_{PR}$) **then**
            Delete segments preceding x_i from P_n;
        else
            Delete segments preceding x_i from all paths in PR;
            Delete segments preceding $x_{i-j}, j = 1, ...$ from paths in $PR \cap PO_{out}$;
        end If
    else
        Delete segments preceding x_i from P_n;
    end If
else
    If (g_{PR}¿0) **then**
        Delete segments preceding x_i from all paths in PR;
        Delete segments preceding $x_{i-j}, j = 1, ...$ from paths in $PR \cap PO_{out}$
    end If
end If

5.2.3 Formal description of ICRA algorithm

Using reduction rules, definitions and the algorithm given previously, ICRA mechanisms can be summarized as follows. When adding a new path to the multicast graph M, ICRA algorithm determines the set X=$\{x_i, x_i \neq s, i = 1, ...\}$ of common node that are in the new path and in other old paths. These common nodes are sorted in ascendant order of their distance from the source node in the new path. They are scanned node by node from the farthest one. Examining a node consists on determining the sub-paths and the segments that can be removed to eliminate redundancies in this node. Sub-paths are candidate to elimination when reduction rule 1 is respected to fulfill destination requirements. If reduction rule 2 is not respected, the examined sub-path must be reduced to the last segment preceding the examined node, as it is described above, to avoid the disconnection of intermediate destination nodes. When the set of segments candidate to elimination is not empty, ICRA applies the Algorithm 3 to chose the optimal set of segments. Paths subject of segment removals must be updated to ensure forwarding multicast traffic to the corresponding destination nodes. ICRA steps are described formally with the meta-code given by the algorithm 2 and the add procedure of a new path to the set of finally obtained paths is detailed by the Algorithm 4.

Algorithm 4 Steps of ICRA (Improved Cycle Reduction Algorithm)

Adding a path P_n to the multicast sub-graph M

Input: A path P_n, the set M of the reduced paths
Output: the new set M of reduced paths

```
stop = false;
VX = ∅; //common nodes already scanned
X = M ∩ P_n; //common nodes in P_n and in old paths
While (stop is false) do
    If (X ≠ ∅) then
        Sort X in decreasing order of the distance to the source in P_n;
        x_i = first node of X;
        Add x_i to VX;
        PR=old paths in M containing the node x_i;
        candidates = ∅;
        If (P_n(s, x_i) can be removed from P_n (Reduction rule1)) then
            r_p = P_n(s, x_i);
            g_{P_n} = h(P_n(s, x_i)); // the gain of this candidate
        else
            g_{P_n} = 0;
        end If
        For (each path P_j in PR) do
            If (P_j can not be rerouted through PR (Reduction rule 2)) then
                remove P_j from PR;
                j++;
            end If
        end For
        If (PR ≠ ∅) then
            Compute the gain g_{PR} as given in reduction rule 3;
            Add corresponding sub-paths or segments to candidates; //taking
            into consideration reduction rule 2;
        end If
        If (candidates ≠ ∅) then
            Select from candidates the segments to remove; //apply algorithm 3
            Connect concerned destination nodes;
            Compute X = M ∩ P_n;
            Remove X_v from X; //remove already scanned nodes
        else
            Verify the min-max constraints; //cycle(s) in x_i can not be removed
        end If
    else
        Add P_n to M as a new path;
        stop=true;
    end If
done
return(M)
```

5.3 Conclusion

In this section, we adopt the same incremental approach proposed by Mamcra but we attempt to go beyond Mamcra problems. ICRA tackles the optimization of the multicast subgraph by takin into account different topology cases and by examining different elimination possibilities. But, ICRA is also an incremental algorithm that explores the solutions spaces and at each iteration, found solutions can not be revoked. So, the algorithm can be blocked in a local minimum which does not correspond to the optimal solution. That's why, metaheuristics can be considered as good candidates to avoid these local minima.

6 Taboo-QMR: Taboo QoS Multicast Routing

In this section, we propose Taboo-QMR algorithm for optimizing QoS multicast routing structure. Before presenting the algorithm mechanisms, we describe briefly the basic concepts of the taboo search.

6.1 Taboo Search overview

Taboo Search is a meta-heuristic approach proposed by Glover [8] to allow local search methods to overcome local optima. This method was applied successfully to solve different network problems. In fact, Taboo search has been used in [28, 29] for optimizing the link capacities in a dynamic telecommunication network.
The basic principle of Taboo Search is to pursue Local Search whenever it encounters a local optimum by allowing non-improving moves. Cycling back to previously visited solutions is prevented by the use of memories, called taboo lists, which record the recent history of the search. The two basic elements of any Taboo Search heuristic are the definition of its search space and its neighborhood structure. The search space is the space of all possible solutions that can be considered (visited) during the search. At each iteration of Taboo Search, elementary transformations are applied from the current solution, denoted X_i, to pass to the set of neighboring solutions in the search space, denoted $N(X_i)$ (the neighborhood of X_i). Then the "least wrong" solution is retained and will be used to compute the next neighborhood set.
The stopping criterion can be for example the maximal number of iterations specified by the user and the better visited solution is returned.

6.2 Taboo-QMR mechanisms

Taking into account that the OMS problem is NP-complete, that cycles are not always present in the set of paths S, and in other cases the number of cycles can be low, we propose the Taboo-QMR algorithm to find a good solution for the multi-constraint multicast routing problem using limited execution time. Taboo-QMR is based on the same concepts of Taboo search to solve the OMS problem. To apply the taboo search method, representation of the possible solutions of the problem and the applied elementary transformation should be defined. So, given S the set of path we must determine which elementary segments of S must be removed.

Representation of a solution: The search can be realized in the reduced graph S" (as it described in section 4.1). A solution corresponds to a combination of the used segments. To represent this solution, we associate to the reduced set of segments S" a bitmap of k bits, where k is the number of segments in S". Thus, each bit corresponds to a segment and is equal to 0 if the segment is removed from S" and it will be coded by 1 if it is kept.

Elementary transformation: An elementary transformation in our Taboo Search corresponds to the remove or re-establishment operation of a segment. Using the chosen representation, it corresponds to the modification of only one bit of the bitmap.

More precisely, let X_i be a solution of OMS problem. The bitmap of this solution is (x_1^i ,x_2^i ,..., x_k^i) where k=N(S_R) and x_j^i =1 or 0 $\forall$ j=*1...k*. We define the transformation t_l (l=1...k):

$$t_l(X_i) = X_{i+1} \text{ with } x_j^i = \begin{cases} x_j^i, & \text{if j} \neq \text{l;} \\ \\ x_j^i + 1, & \text{if j = l.} \end{cases} \tag{16}$$

After given these defintions, Taboo-QMR algorithm consists on two steps presented in Algorithm 5. First, it computes the set S of feasible paths, than it tries to optimize redundancies. According to the number of cycles in G_S, the graph obtained from the set S, exhaustive search algorithm or taboo search algorithm is adopted to compute from G_S, the optimized graph solution of the OMS problem. If the number of cycles exceeds a T_c threshold, then the taboo search algorithm is adopted else an exhaustive search can easily find the best solution. Taboo-QMR meta-code is given in Algorithm 5.

Algorithm 5 Steps of Taboo-QMR Algorithm

Taboo-QMR Algorithm

Input: The network $G = (N, E)$, a group g with a source s, constraints L_i, the set S of QoS unicast paths, T_c the threshold cycle number
Output: M : a set of paths

Step 1:Computing paths

Compute G_S: set of feasible paths for all members of the group

Step 2:Optimizing procedure

```
If (number of cycle = 0) then
    stop;the solution corresponds to G_S
else
    Reduce G_S to G_S"
    If (number of cycle¡threshold) then
        Execute exhaustive search algorithm
    else
        Execute Taboo Search algorithm
    end If
end If
```

Executing exhaustive algorithm consists on examining which segments can be omitted from to eliminate the maximum of redundancies without violating the constraints. An exhaustive enumeration of all combinations of segments should be performed to determine which subset of segments can be eliminated with maximal gain. Such algorithm is possible when the number of cycles is very low ($\approx 2or3$) but when there are more cycles, the exhaustive search is expensive to achieve. In that case, taboo based search is adopted. Taboo search algorithm tries to find the optimal solution by beginning from an initial solution X_0, the one given by the unicast QoS routing, and then explore the research space. At each iteration, this exploration is assured by generating a neighborhood set of the actual solution X_i by transforming 1 bit of the bitmap using the transformation defined in (16). The best feasible solution, according to the length function l or to the hop count h, of this set is retained and will be used as the start point for the next iteration. A solution is feasible if there is a feasible path for each destination in the graph containing the articulation segments and not the reduced graph. Naturally, all the segments having their coding bit equal to 0 are omitted for this verification. A solution X_2 of the MCMLM problem is better then an already retained solution X_1, if the two solutions correspond to feasible sub-graphs (without reduction 2) and the number of segments in X_2 is less then that in X_1. When X_1 and X_2

have the same number of segments, X_2 is better when it has the least critical length (diameter) ($cl(X_2) \leq cl(X_1)$) that we already defined according to the length function l defined in [17]. At the end of a fixed number of iterations, the best solution of the retained solutions of each iteration corresponds to the proposed solution for the MCMLM or the MCMWM according to the cost function used.

Algorithm 6 Taboo search algorithm used in Taboo-QMR

Taboo Search Algorithm

Input: a bitmap representing the reduced graph G", an objective function f
Output: X_{opt} : a bitmap corresponding to the optimal solution

Initializations

X_0=(x_1^0,x_2^0 ,..., x_k^0) with $x_j^0 = 1$ for ($j = 1 \dots k$)
X_{opt}← X$_0$;
f_{min}← f(X$_0$);
TabooList←∅ ;

Taboo search iterations

```
While (not stop) do
    For i from 1 to k do
        R = t_1(X_{i-1});
        If (t_l ∉ TabooList) then
            X_l ← t_l(X_{i-1})
        end If
        If (((Feasible(X_l) and f(X_l) < f(R)))) then
            R ← X_l; //choose the solution minimising f
            T ← l; //to memorize the transition t_T
        end If
    end For
    Add t_T to TabooList
    (Delete the oldest transformation eventually)
    If (f(R) < f_min) then
        X_opt ← R;
        f_min ← f(R);
    end If
    X_{i-1} ← R;
done
return (X_opt);
```

We notice that like it is defined Taboo-QMR attempts to optimize the graph G_S by considering all paths found by the unicast QoS routing and not like Mamcra does by constructing the sub-graph solution M incrementally. An incremental approach should be used to achieve join of new members. In that case, an optimized multicast sub-graph M exists to route multicast traffic to the destinations. When a set of destination nodes D want to join the multicast session, the unicast multi-constraint paths from s to every node of D is computed. If S_D is the set of these paths, the optimizing procedure of Taboo-QMR can be applied to optimize $M \cup S_D$.

7 Simulations

In this section, we present and discuss our simulation results for optimizing multi-constraint multicast routing. Since the set of shortest paths applying the presented non-linear length function is a good start point, we implemented Samcra to generate the set S of multi-constraint shortest paths from source node to each destination node. Then for comparison of the efficiency of our algorithm to greedy optimization, we tested two algorithms: Mamcra algorithm and Taboo-QMR. These algorithms were run on different realization of Internet like random graphs implementing degree constrained Waxman topologies [32]. The multicast groups were randomly chosen in the graphs. One simulation test consisted of generating 100 topologies. The values of the m link weights were sampled from independent uniform distributions in the range (0, *Maximum link Value*). To show the overall performance of the proposed algorithm, we adopt three tests scenarii. In the following, we first analyze the result set returned by the multi-constraint unicast routing algorithm to evaluate the size of the second optimization/reduction problem. The complexity of the optimization can be characterized by the number of redundancies that is why we analyze the number of cycles in the set S. Then, we compare the results of the optimization of the obtained sub-graph S by executing Mamcra and the proposed Taboo-QMR.

7.1 Results on the number of cycles

The first set of tests concerns the number of cycles in the multicast sub-graph before any optimization. (Figure 17) represents the maximum number and the average number of cycles of the multicast sub-graph according to the number of nodes of the Waxman topology when the group size is fixed (40 nodes) and the Maximum link value is fixed (all link weights are into [0,10]). The constraint limits are constant and equal to (40, 40). Each value in the figure corresponds to the average obtained on the base of 100 simulations with the same parameters in the described random graphs. We state than the number of cycles increases when the graph size grows. In Figure 18, the same values are represented as a function of the group size for topologies having a fixed size (70 nodes) and for the same Maximum link value (10,10).

We can deduce from these results, that number of cycles that appears when computing multi-constraint multicast sub-graph in the studied graph isn't important, the average num-

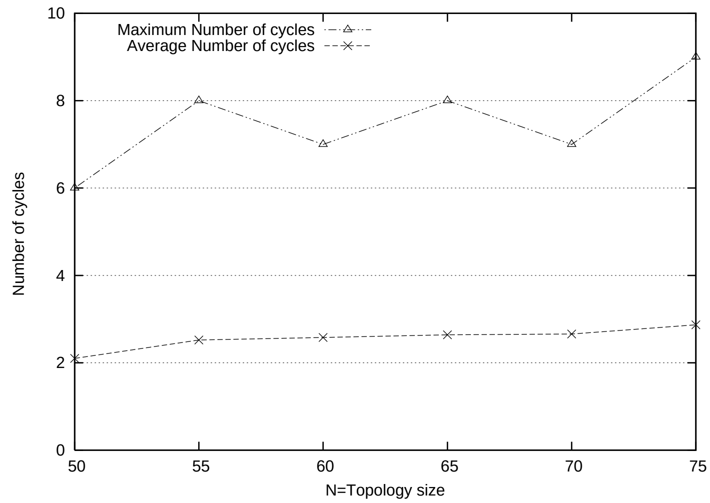

Figure 17: Number of cycles as a function of topology size: waxman graph

ber is around 3 cycles per topology but in some cases it can reach more important values (7 to 10) cycles.

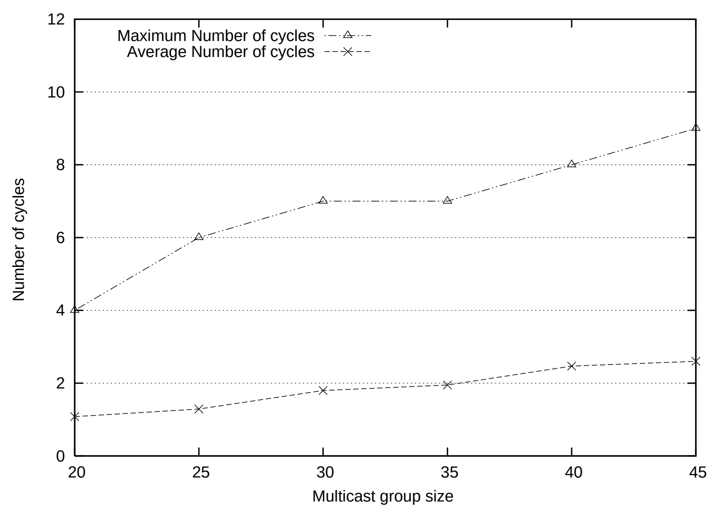

Figure 18: Number of cycles as a function of multicast group size:waxman graph

When we run the tests on planar graph composed of 100 nodes and with constraints equal to 90 , the number of cycle experienced is more important as it is shown on figure 19. Thus, the complexity of the optimization problem increases when the network topology size is important, when the group size increase. It depends also on topology's type. That's why optimizing such a sub-graph makes QoS multicast routing more efficient.

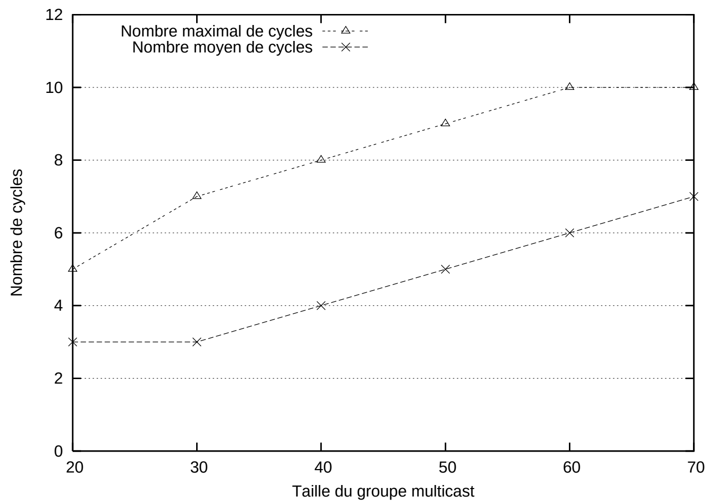

Figure 19: Cycles number according to group size: planar graph

7.2 Results on optimizing multicast sub-graph

In this section, we compare the Mamcra algorithm to Taboo-QMR algorithm for optimizing multicast sub-graph S. We have executed the two algorithms on the same topologies described above. In the presented example, the network topology graph was composed of 100 nodes and the multicast group contained 70 members. The constraint limits are fixed to 40 for Waxman graphs and to 90 for planar graphs. These values are chosen in order to adapt constraints to the graph characteristics. In fact, if constraints are two hard to satisfy, few feasible paths are found and if constraints are two loose, many feasible paths exists and so cycles can be easily omitted. That's why we adjust constraint limits value according to the used topology. In the following M_{Mamcra} corresponds to the optimized multicast sub-graph obtained by Mamcra and M_{Taboo} corresponds to the optimized multicast obtained by Taboo-QMR algorithm.
Table (1) gives the average cycle number in S, M_{Mamcra} and M_{Taboo}. $\mathcal{C}_{moyen}(G)$ corresponds to the number of cycle existing in the graph G. So, $\mathcal{C}_{moyen}(S)$ corresponds to the initial average number of cycle, $\mathcal{C}_{moyen}(M_{Mamcra})$ and $\mathcal{C}_{moyen}(M_{Taboo})$ correspond to the

Network	$\mathcal{C}_{moyen}(S)$	$\mathcal{C}_{moyen}(M_{Mamcra})$	$\mathcal{C}_{moyen}(M_{Taboo})$
Waxman	3,61	0,0426	0,0107
Planaire	6,59	0,3939	0,0221

Table 1: Number of cycles in the optimized graph : Taboo-QMR vs Mamcra

Graph topology	$cl(S)$	$cl(M_{Mamcra})$	$cl(M_{Taboo})$
Waxman	0,7610	0,7820	0,7656
Planaire	0,8531	0,8598	0,8539

Table 2: Average critical length: TabooQMR vs Mamcra

average number of cycle in the routing structure obtained after executing the corresponding algorithm. We notice that Mamcra gives generally good results for optimizing cycles. The results illustrate that Taboo-QMR removes, approximatively, all cycles when Mamcra can do it and removes other cycles that Mamcra can't remove.

Furthermore, if we compare, solutions obtained by Mamcra and Taboo-QMR according to the critical length cl(M), on average, Taboo-QMR find solutions having a better diameter than solutions found by Mamcra as it is shown in Table 2. We also notice during simulations that the iteration number where Taboo-QMR found the better solution depends of the number of cycles that must be removed. In fact, if the sub-graph S contains k cycles, Taboo-QMR execution requires k iterations.

To confirm the improvement brought by Taboo based algorithm compared to Mamcra algorithm, we compare routing structure obtained by Taboo-QMR to ones obtained by Mamcra. We notice that in 96.3% of the cases, Taboo-QMR gives solutions having a critical length shorter or equal to Mamcra solutions and in 32,45% of the cases Taboo algorithm gives solutions having a critical length cl(M) shorter than the one obtained by Mamcra algorithm. Few are the cases (3,7%) where Mamcra gives better results according to the critical length cl(M).

With cycles	No cycle	Taboo solutions better than Mamcra ones	Taboo solutions less than Mamcra ones	Same critical length
95	5	23	4	68
94	6	37	3	54
97	3	32	5	60
93	7	31	2	60
Average		32,45%	3,69%	63,85%

In nearly all cases, solutions obtained by Taboo based algorithm reduces the critical length compared to Mamcra algorithm. Taboo based algorithm succeeds to find improving

Table 3: Taboo solutions compared to Mamcra solutions according to critical length

solutions in few iterations. All these results make taboo based algorithm a good candidate to optimize multicast sub-graphs and so to achieve multi-constraint QoS routing.

8 Conclusion

In this paper, we discussed improvements of the multi-constraint multicast routing algorithm based on a taboo search algorithm. The QoS routing problem is known as a NP-complete multi-constraint optimization problem. Exhaustive search algorithms and a greedy algorithm were proposed to find multicast sub-graphs for multicast communications requiring QoS. The exhaustive algorithm is not scalable and can be applied only for limited sizes. The greedy algorithm Mamcra proposed for this routing problem does not take advantage of simple but possible reductions of the multicast sub-graph. We propose ICRA an incremental algorithm to solve the OMS problem. Even if ICRA attempts to avoid some of Mamcra weakpoints, it can not avoid problems of iterative algorithms. To obtain a good trade-off, we proposed a taboo search based algorithm, Taboo-QMR, for the cases when exhaustive search is not possible. Simulation results show that a firstly proposed, shortest path based multicast graph is often irreducible. If redundancies exist in the multicast graph, the taboo search based algorithm eliminates more cycles than the Mamcra algorithm. In randomly generated graphs, the solution obtained by the taboo algorithm has a less diameter than the solution found by Mamcra. Generally, a loop-free solution was found in time $O(k)$ in the cases where k cycles were detected in the set of shortest paths.

References

[1] E.I. Chong, S. Maddila, S. Morley, On finding single-source single destination k shortest paths, J. Comput Inf., Special Issue ICCI'95, pp. 40-47(1995).

[2] S. Chen, K. Nahrstedt, An Overview of Quality-of-Service Routing for the Next Generation High-Speed Networks: Problems and Solutions, in IEEE Network, Special Issue on Transmission and Distribution of digital Video, pp. 64-79,(1998).

[3] S. Chen , K. Nahrstedt, On Finding Multi-Constrained paths, IEEE ICC, (1998).

[4] T. H. Cormen, C. E. Leiserson , R. L. Rivest, Introduction to Algorithms, MIT Press, Cambridge, MA, (2000).

[5] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik, no. 1, pp. 269-271, (1959).

[6] M. Faloutsos, A. Banerjea, R. Pankaj, QoSMIC: Quality of Service sensitive Multicast Internet protoCol, in ACM SIGCOMM, pp. 144-153,(1998).

[7] M.R. Garey , D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-completeness, Freeman, San Francisco, (1979).

[8] F. Glover, Future Paths for Integer Programming and Links to Artificial Intelligence, Computers and Operations Research 13, pp. 533-549, 1986.

[9] W. Goodridge, W. Robertson, W. J. Phillips , S. Sivakumar, Heuristic Constraint-Path Routing Decision System, in Proceedings of the 3rd Annual Communication Networks and Services Research Conference (CNSR'05), IEEE,(2005).

[10] L. Henrique, M. K. Costa, S. Fdida, O.Carlos, M. B. Duarte, Hop-by-Hop Multicast Routing Protocol, In ACM SIGCOMM, pp. 249-259, (2001).

[11] F. Hwang , D. Richards, Steiner Tree Problem, Networks, vol. 22. no. 1, pp.55-89, (1992).

[12] A.Juttner, B. Szviatovszki, I. Mecs , Z. Rajko, Lagrange Relaxation Based Method for the QoS Routing Problem, IEEE INFOCOM, (2001).

[13] X. Jia, Y. Zhang, N. Pissinou, K. Makki, A distributed multicast routing protocol for real-time multicast applications, in Computer Networks 31, pp.101-110, (1999).

[14] F.A. Kuipers, T. Korkmaz, M. Krunz , P. Van Mieghem, An Overview of Constraint-Based Path Selection Algorithms for QoS Routing, IEEE Communications Magazine, vol. 40, no. 12, (2002).

[15] T. Korkmaz, M. Krunz , S. Tragoudas, Multiconstrained Optimal Path Selection IEEE INFOCOM, (2001).

[16] T. Korkmaz, M. Krunz, S. Tragoudas, Efficient Algorithm for Finding a Path Subject to Two Additive Constraints, In Computer Communications Journal, Vol. 25, No. 3, February , pp. 225-238,(2002).

[17] F. Kuipers, P. V. Mieghem, Mamcra: A Constrained-Based Multicast Routing Algorithm, In Computer Communications, Vol. 25, No. 8, pp. 801-810,(2002).

[18] V. P. Kompella, J. C. Pasquale, G. C. Polyzos, Multicast Routing for Multimedia Communication_, in IEEE/ACM Transactions on Networking, Vol. 1, No. 3, pp. 286-292,(1993).

[19] F. A. Kuipers , P. Van Mieghem, Conditions that Impact the Complexity of QoS Routing,IEEE/ACM Transactions on Networking, vol 13/4, (2005).

[20] P. Van Mieghem (ed.), F.A. Kuipers, T. Korkmaz, M. Krunz, M. Curado, E. Monteiro, X. Masip-Bruin, J. Solé-Pareta, , S. Sánchez-López, Quality of Service Routing, Chapter 3 in Quality of Future Internet Services, EU-COST 263 Final Report, edited by Smirnov et al. in Springer LNCS 2856, pp. 80-117, (2003).

[21] P. Van Mieghem , F. A. Kuipers, Concepts of Exact QoS Routing Algorithm-s,IEEE/ACM Transactions on Networking, vol. 12, no. 5, (2004).

[22] P. Van Mieghem, H. De Neve , F.A. Kuipers, Hop-by-Hop Quality of Service Routing, Computer Networks, Vol. 37, No. 3-4, pp. 407-423, (2001).

[23] H. De Neve , P. Van Mieghem, Tamcra: A Tunable Accuracy Multiple Constraints Routing Algorithm, Computer Communications, Vol. 23, pp.667–679, (2000).

[24] G. N. Rouskas, I. Baldine, Multicast Routing with End-to-End Delay and Delay Variation Constraints, in IEEE JSAC, Vol. 15, No. 3, pp. 346-356,(1997).

[25] H. F. Salama, D. S. Reeves, Y. Viniotis, Evaluation of Multicast Routing Algorithms for Real-Time Communication on High-Speed Networks, in IEEE JSAC, Vol 15, No. 3, pp.332-345, (1997).

[26] L. Wei, D. Estrin, The Trade-offs of Multicast Trees and Algorithms, in Proceedings of the 1994 International Conference on Computer Communications and Networks, pp. 17-24, (1994).

[27] B. Wang, J. C. Hou, Multicast Routing and its QoS Extension: Problems, Algorithms and Protocols, in IEEE Network, pp. 22-36, 2000.

[28] J. Xu, S. Y. Chiu, F. Glover Tabu search for dynamic routing communications network design, Telecommunication Systems, vol. 8. pp 55-77, (1997).

[29] J. Xu, S. Y. Chiu, F. Glover Optimizing a Ring-Based Private Line Telecommunication Network using Taboo Search, Management Science, vol. 45, No 3, (1999).

[30] G. Apostolopoulos, D. Willams, S. Kamat, R. Guerin, A. Orda , T. Przygienda, QoS routing mechanisms and OSPF extensions, RFC 2676, (1999).

[31] The ATM Forum, Private network-to-network interface specification version 1.1 (PNNI 1.1),af-pnni-0055.002, (2002).

Contents