Universitat de Girona

MULTI-OBJECTIVE OPTIMIZATION SCHEME FOR STATIC AND DYNAMIC MULTICAST FLOWS

Yezyd DONOSO MEISEL

ISBN: 84-689-3757-6
Dipòsit legal: GI-1013-2005

Universitat de Girona

Department of Electronics, Computer Science and Automatic Control

PhD Thesis

Multi-Objective Optimization Scheme for Static and Dynamic Multicast Flows

Author: Yezid Donoso Meisel

Supervisor: Ramon Fabregat Gesa

Thesis presented complying with the requirements for
the PhD degree in Information Technology
Girona, April 2005

Dr.	
President	

Dr.	
Secretary	

Dr.	
Vocal	

Dr.	
Vocal	

Dr.	
Vocal	

Date of presentation	
Qualification	

To my wife Adriana
for her love and tenderness and for our future together

To my children Andres Felipe and Daniella
…a gift of God to my life

Acknowledgements

Initially, I would like to thank my supervisor, Dr. Ramon Fabregat, for giving me the opportunity to do research in his department. As an expert, his technical and strategic advices have provided me clear solutions to the difficult situations encountered during my doctoral studies.

I welcome the opportunity to express my gratitude to Dr. Benjamin Barán for his guidance, commitment and collaboration, making it possible for me to advance and extend my knowledge within the fields of multi-objective optimization and evolutionary algorithms.

I would like to thank to Dr. Jose Luis Marzo by his corrections in the different important moments of this thesis.

I would like to thank to Dr. Ramón Puigjaner and Dr. Teodor Jové by his comments and suggestions about the different aspects of this thesis

I also wish to express grateful to Fernando Solano by the collaboration to finish my PhD thesis.

I also wish to express grateful to Carolina Alvarado by the collaboration with the results to finish my PhD thesis.

I wish to express my gratitude to all members of the Broadband Communications and Distributed Systems Group: Antonio, Cesar, David, Eusebi, Lluís, Liliana, Pere, who helped me in different situations.

Finally, I would like to thank the many people who have, in one way or another, made this thesis possible. I apologise for not listing everyone here.

This PhD was supported by the Universidad del Norte (Colombia) under contract G01 02-MP03-CCBFPD-0001-2001.

Abstract

Many new multicast applications emerging from the Internet, such as TV over the Internet, Radio over the Internet, Video Streaming multi-point, etc., need the following resource requirements: bandwidth consumption, end-to-end delay, packet loss ratio, etc. It is therefore necessary to formulate a proposal to specify and provide for these kinds of applications the resources necessary for them to function well.

In this thesis, we propose a multi-objective traffic engineering scheme using different distribution trees to multicast several flows. In this case, we are using the multipath approach to every egress node to obtain the multitree approach and of this way to create several multicast tree. Moreover, our proposal solves the traffic split ratio for multiple trees. The proposed approach can be applied in Multiprotocol Label Switching (MPLS) networks by allowing explicit routes in multicast events to be established.

In the first instance, the aim is to combine the following weighting objectives into a single aggregated metric: the maximum link utilization, the hop count, the total bandwidth consumption, and the total end-to-end delay. We have formulated this multi-objective function (MHDB-S model) and the results obtained using a solver show that several weighting objectives are decreased and the maximum link utilization is minimized.

The problem is NP-hard, therefore, an algorithm is proposed for optimizing the different objectives. The behavior we get using this algorithm is similar to what we get with the solver.

Normally, during multicast transmission the egress node can leave and enter of the tree and for this reason in this thesis we propose a multi-objective traffic engineering scheme using different distribution trees for dynamic multicast groups (i.e. in which egress nodes can change during the connection's lifetime). If a multicast tree is recomputed from scratch, it may consume a considerable amount of CPU time and all communication using the multicast tree will be temporarily interrupted. To alleviate these drawbacks we propose an optimization model (dynamic model MHDB-D) that uses a previously computed multicast tree (static model MHDB-S) adding new egress nodes.

Using the weighted sum method to solve the analytical model is not necessarily correct, because is possible to have a non-convex space solution and some solutions cannot be found. In addition, other kinds of objectives were found in different research works. For the above reasons, a new model called GMM is proposed and to find a solution to this problem a new algorithm using a Multi-Objective Evolutionary Algorithm (MOEA) is proposed too. This algorithm is inspired by the Strength Pareto Evolutionary Algorithm (SPEA).

To give a solution to the dynamic case with this generalized model a dynamic GMM model is proposed and a computational solution using Breadth First Search (BFS) probabilistic is also proposed to give a solution to the dynamic case in multicast.

Finally, in order to evaluate our proposed optimization scheme, we performed the necessary simulations and tests.

The main contributions of this thesis are the taxonomy, the optimization model and the formulation of the multi-objective function in static and dynamic multicast transmission (MHDB-S and MHDB-D), as well as the different algorithms proposed to give computational solutions to this problem. Finally, the generalized model with several functions found in different research works in static and dynamic multicast transmission (GMM and Dynamic GMM), as well as the different algorithms proposed to give computational solutions using MOEA and BFS probabilistic.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Glossary

ATM	Asynchronous Transfer Mode.
BC	Total Bandwidth Consumption.
B-ISDN	Broadband Integrated Service Digital Network.
BFS	Breadth First Search.
BGMP	Border Gateway Multicast Protocol.
CBT	Core Based Tree
CR-LDP	Constraint-based Routing LDP.
CRCDM	Controlled Rearrangement for Constrained Dynamic Multicasting.
CSPF	Constraint-based Shortest Path First
D GMM	Generalized model that optimize eleven different functions in multi-objective context in the Dynamic case.
DL	Total end-to-end delay.
DNLP	Nonlinear programming with discontinuous derivatives.
DVMRP	Distance Vector Multicast Routing Protocol.
DWDM	Dense Wavelength Division Multiplexing.
EA	Evolutionary Algoritms.
ECMP	Equal Cost Multiple Paths.
FEC	Forwarding Equivalent Class.
GMM	Generalized model that optimize eleven different functions in multi-objective context in the static case.
GMPLS	Generalized Multiprotocol Label Switching.
HC	Hop Count.
IP	Internet Protocol.
ISP	Internet Service Provider
LEF	Linear Energy Functions.
LDM	Load Distribution over Multipath.
LSP	Label Switched Path.
MEFPA	Multi-constrained Energy Function based Precomputation Algorithm.
MHDB-D	Model that optimize the following function: Maximum link Utilization, Hop count, Delay and Bandwidth consumption in the Dynamic case.
MHDB-S	Model that optimize the following function: Maximum link Utilization, Hop count, Delay and Bandwidth consumption in the Static case.
MLU	Maximum Link Utilization.

MMR-D	Dynamic Multicast Multitree Routing Algorithm.
MMR-S	Static Multicast Multitree Routing Algorithm.
MOEA	Multi-Objective Evolutionary Algorithms.
MOP	Multi-Objective Problem.
MOSPF	Multicast Open Shortest Path First.
MPLS	Multiprotocol Label Switching.
MP	Multipath.
MT	Multitree.
NP	Non-Polynomial
NSGA	Non-dominated Sorting based Genetic Algorithm.
OMS	Opportunistic Multipath Scheduling.
OSPF	Open Shortest Path First.
P2MP	Point to Multi-Point.
P2P	Point to Point.
PIM-DM	Protocol Independent – Dense Mode.
PIM-SM	Protocol Independent – Sparse Mode.
QoS	Quality of Service.
QoSR	Quality of Service Routing.
RSVP	Resource Reservation Protocol.
RSVP-TE	Resource Reservation Protocol with Traffic Engineering (MPLS)
SLA	Service Level Agreements.
SOP	Single-Objective Problem.
SPEA	Strength Pareto Evolutionary Algorithm.
SPT	Shortest Path Tree.
TCP	Transport Control Protocol.
TE	Traffic Engineering.
TEAM	TE using Adaptive Multipath-forwarding.
UP	Unipath.
UT	Unitree.
WFQ	Weighted Fair Queuing.

Chapter 1 Introduction

This chapter describes the motivation for this work in reference to certain problems and new trends we have detected in the field of network optimization in multicast transmission. From this starting point we out line the main aims of the thesis along with the expected achievements. The chapter ends by describing the structure and contents of the document.

1.1 Overview of the problem

Traffic engineering aims to optimize the performance of operational networks. The main objective is to reduce congestion hot spots and improve resource utilization. This can be achieved by setting up explicit routes through the physical network in such a way that the traffic distribution is balanced across several traffic trunks. This load balancing technique can be achieved by a multicommodity network flow formulation which leads to the traffic being shared over multiple routes between the ingress node and the egress nodes in order to avoid link saturation and hence the possibility of congestion, which is the inability to transmit a volume of information with the established capacities for a particular equipment or network.

When we translate this balancing technique into a mathematical formulation, the main objective is to minimize the maximum link utilization. When the network is congested, minimizing the maximum link utilization involves: 1) minimizing the congestion of links, 2) reducing the total packet delay, and 3) minimizing the total packet loss.

One solution is the multipath approach, in which the data is transmitted along different paths to achieve the aggregated, end-to-end bandwidth requirement. Several advantages of using multipath routing are discussed in the literature, such as: the links do not get overused and therefore do not get congested, therefore they have the potential to aggregate bandwidth, allowing the network to support higher data transfer than is impossible with just one path.

We can also have per-flow multipath routing where an originating node uses multiple paths for the same flow, i.e. each flow is split into multiple subflows. The split ratio is fed to the routers which divide the traffic of the same ingress-egress node pair into multiple paths. Several papers address this splitting multipath problem of unicast traffic, motivated by its importance in any complete traffic engineering solution. Traffic splitting is executed for every packet in the packet-forwarding path. However, in this research work a multicast approach is proposed.

Multicast connections are connections between one or more senders and a number of members of a group. The aim of multicasting is to be able to send data from a sender to the members of a group in an efficient manner. Many multicast applications, such as audio and videoconferencing or collaborative environments and distributed interactive simulation, have multiple quality-of-service requirements in relation to bandwidth, packet delay, packet loss, cost, etc. In multicast transmission, load balancing consists of traffic being split (using the multipath approach) across multiple trees, between the ingress node and the set of egress nodes. The multicast transmissions function has a particular way, in many cases the egress nodes can enter or leave the transmission tree and in this situation the connections are dynamic. Currently, multicast transmissions can be applied using switching technology and in this research work we use a MPLS technology for this function.

MPLS is a versatile solution addressing current problems at a network level such as velocity, scalability, quality of service, and applying traffic engineering. The central idea of MPLS is to add a label to each package to be sent. These packages are assigned a pair of short length values that synthesizes the source and egress of the package. At the end of the thesis, we will propose the use of MOEAs to give a computational solution to our optimization problem.

The term Evolutionary Algorithm (EA) refers to searching and optimizing techniques inspired by the evolution model proposed by Charles Darwin. The EAs are interesting given the fact that at first glance they seem especially apt for dealing with the difficulties posed by MOPs. The reason for this is that they can return an entire set of solutions after a simple run and they do not have any other of the limitations of traditional techniques. In fact, most recent publications on MOP resolutions using EAs, seem to consider this fact and they have opened the way to a whole new field of investigation: evolutionary algorithms applied to multi-objective optimization MOEA.

1.2 Objectives

In this section we outline the objectives of this thesis.

- To develop an optimization model for the transmission of multicast applications. The main objective function in the development of the model is to minimize the maximum link utilization. The solution obtained by this model can distribute traffic between multiple trees by using a multi-path scheme from the ingress node to each egress node. Using this scheme it is possible to transmit the information through more than one path between ingress node and egress node. For the multicast case, which is the topic of this thesis, it would consist in transmitting information flow through more than one tree. The contribution is that the current multicast routing protocols such as DVMRP [WAI88] [PUS00], MOSPF [MOY94], PIM-DM [DEE98], PIM-SM [EST98], CBT [BAL97] [BAL97a] and BGMP [THA00], transmit the information through just one tree.

- To broaden the possibilities of the above model with other objectives (total hop count, end-to-end delay and bandwidth consumption). As this model uses a load-balancing technique it is possible, in this case, that very long paths can be found.

- To make the model appropriate for dynamic connections. As in multicast transmissions, egress nodes can go in and out of the connection during the life-time connection, another objective is to present a new model solving the problem of dynamic nodes.

- To specify the way of creating paths between ingress nodes and egress ones by using MPLS technology, that is Label Switched Paths (LSPs) based on solutions found with the analytical model previously proposed.

- To propose a taxonomy to classify related works and this thesis contributions.

- To define a generalized model in order to be able to consider and cover most of different models and their objectives, found in bibliographical reviews.

- To find a solution to the problem by using MOEA.

- To analyze the validity of the proposed models through simulations.

1.3 Contributions

We list here the contributions of this thesis:

- The use of the "multipath approach" to solve the "load balancing" problem in multicast transmissions.

- An analytical model to solve the problem when the set of egress nodes of multicast transmissions is static.

- The use of four different objectives in the analytical model previously presented.

- As the static model has an NP complexity, a heuristic to solve this problem is proposed.

- An analytical model with four objective functions to solve the problem when the set of egress nodes of multicast transmission is dynamic.

- As the dynamic model has an NP complexity, a heuristic to solve this problem is proposed.

- As mapping the obtained results with the models (static and dynamic) and the explicit paths that must be established in an MPLS network is not "immediate", a model and a heuristic have been proposed to map sub-flows to point to multi-point (P2MP) LSP.

- A new general model considering most of the objectives found in the related works are analyzed. The variables obtained in this case are at a "sub-flow" level (a new index has been considered) and allow direct mapping between the results obtained by the models and the explicit paths that must be considered.

- An MOEA to solve the proposed general model as a system using multi-objectives considering the objectives previously mentioned in the literature. This model solves the problem even in the space of non-convex solutions.

- A new heuristic to solve the general model when the set of egress nodes of multicast transmissions is dynamic.

1.4 Outline of the Thesis

Chapter 2 is clearly divided into two main parts. In the first part we present the fundamental concepts. In this section we explain the concepts of Quality of Service (QoS), traffic engineering and load balancing of unicast flows. In addition, we present a taxonomy of our research and other related topics. In the second part, we present the related works with this research.

Once we have presented the fundamental concepts and the related works, in Chapter 3 we present our first analytical model (MHDB-S) and computational solution (MMR-S) as a proposal to solve static multicast transmission using load balancing. We finish with some conclusions, problems and motivations with respect to the presented proposal.

Since in some multicast transmissions the egress node can enter or leave the transmission tree, in Chapter 4, we present the Multi-Objective Optimization in Dynamic multicast Routing (MHDB-D). In this case an analytical model and algorithm are proposed. We finish with some conclusions, problems and motivations with respect to the presented proposal.

Since it is necessary to apply this proposal using MPLS technology, a mapping sub-flows to point-to-multipoint label switched path (P2MP LSPs) is proposed in Chapter 5. We present the specific problem and solution of mapping sub-flows (P2MP LSPs) for a MPLS network. We finish with some conclusions, problems and motivations with respect to the presented proposal.

Some problems and limitations are presented in Chapters 3, 4, and 5. To solve these drawbacks, in Chapter 6 we present a generalized model to static and dynamic case called the GMM-model and Dynamic GMM-model respectively and a computational solutions to both case using Multi-Objective Evolutionary Algorithms (MOEAs Previously, Multi-Objective optimization and Evolutionary Algorithms applied to Multi-Objective optimization are introduced.

Chapter 7 includes the analysis and evaluation of the different proposals. We present several results to demonstrate that the objectives have been achieved and that the different mechanisms operate correctly. We also evaluate how the system performs as a whole by means of simulation results in different scenarios.

In Chapter 8, we conclude this document and summarize the main contributions. We also list future work.

Finally, in Appendix A we are presenting other kind of model to give an analytical solution to the Multi-Objective Optimization in NonConvex Solution Spaces. In Appendix B we are presenting some Framework Considerations and in Appendix C we list the different publications of this research thesis.

Chapter 2 Background and Related Work

2.1 Overview

In this chapter we present the fundamental concepts that are necessary to develop this doctoral thesis. We will talk about Quality of Service (QoS) in Internet, Traffic Engineering, Load Balancing in Unicast Transmission, a taxonomy and related works with this research.

2.2 Fundamental Concepts

2.2.1 Quality of Service (QoS) in the Internet

As the use of the Internet spreads in both business and entertainment sectors, the notion of QoS has become more critical for network service providers. Today's standard IP (Internet Protocol) networks support only a single service level called best-effort service. For best-effort service, the network will try its best to forward the traffic without giving guarantees on the routing performance in terms of loss rate, bandwidth, delay, delay jitter, and so on. All packets are treated equally regardless of their source applications. In the current Internet, some of the applications have elastic requirements, e.g., they can tolerate packet losses and/or delays, or they can respond to the congestion by decreasing their transmission rates. Remote terminal (e.g., Telnet), file transfer protocol (e.g., FTP), and electronic mail are among the examples for elastic applications.

Although best-effort service is acceptable for elastic applications, it is not tolerable for real-time and multimedia applications such as Internet telephony and videoconferencing. Real-time applications have more difficult requirements than the elastic applications. Their performance is very sensitive to packet losses, delays and delay jitters throughout the network. Moreover, they can not reduce their transmission rates in case of congestion.

Under the hard requirements of some specific applications, the notion of QoS has been very popular in the Internet literature. QoS is defined as "a set of service requirements to be met by the network while transporting a flow", where flow implies a packet stream associated with a specific application. Alternatively, QoS can be defined as the level of service measured by the user(s) of the network. QoS requirements of a specific flow can be specified in terms of packet loss probability, bandwidth, end-to-end delay, reliability, etc. The customers of the network may agree with the service providers on the QoS requirements via Service Level Agreements (SLAs).

Quality of Services, in its simplest form, is defined as the mechanism that fulfils the requirements of the applications in a network, that is, one or some elements of the network that can guarantee, to a certain extent, that the traffic needs are fulfilled. All the layers of the network are involved in these elements of quality of service. QoS works by assigning priorities according to network traffic needs, and by managing in this way the network bandwidth [WAN01].

The following parameters have been widely used to describe the requirements of Quality of Service:

- **Minimum bandwidth**: the minimum amount of bandwidth requested by an application flow. The time interval for measuring the bandwidth must be specified because different intervals may give different results. The algorithms of packet scheduling guarantee bandwidth assignment.

- **Delay**: Delay requesting can be specified as the average delay or the delay in the worst case. The delay experimented by a package consists of three components: propagation delay, transmission delay, and queue delay. Propagation delay is caused by light speed and is directly related to the distance. Transmission delay is the time taken to send a package across a link. Queue delay is the time that packages have to wait.

- **Delay jitter**: a delay jitter request can be said to specify the maximum difference between the longest delay and the shortest one experienced by the packages. This should not be longer than the transmission in the worst case and the delay in a queue.

- **Loss rate**: This is the rate of lost packages and the total amount of transmitted packages. Package loss in Internet is often caused by congestion. By assigning enough bandwidth and buffers for traffic flow these losses can be prevented.

One of the causes of low performance in networks is congestion, which is the inability to transmit a volume of information with the established capacities for a particular equipment or network.

The following aspects are examples of this congestion:

- Congestion at port level: Multiple ingress flows compete for the bandwidth of the same egress port.

- Congestion in intermediate nodes: This occurs if the bandwidth of the backplane - switching matrix of a node is less than the total aggregate of its entries.

- Congestion in the network: This occurs if at some point between the ingress node and the egress node, one or several pieces of equipment or links of the network experiment become congested.

Congestion can also be caused by unbalanced traffic distribution, in this case when the traffic is distributed unevenly across the network some links in the network are overloaded while other links are underutilized.

Although the problem of inadequate network resources must be solved with new capacities or by reducing and controlling the demands, unbalanced traffic distribution can be addressed by managing the resources in the network better.

Basically, congestion problems can be solved by two options:

- Increasing the bandwidth in the network, which has its cost and, moreover, the bandwidth is not infinite.

- Managing the available bandwidth in an intelligent way. The network can monitor the use of its bandwidth, observe the symptoms of congestion, and reinforce the policies related with supplying, using and distributing the available bandwidth.

2.2.2 Traffic Engineering

To support QoS in today's Internet, several new architecture models have been proposed. Traffic engineering has become a key issue within these new architectures, as supporting QoS requires more sophisticated resource management tools. The goal of traffic engineering is the performance evaluation and optimization of operational networks. Traffic engineering has become an essential requirement for ISPs to optimize the utilization of existing network resources and it enables to maintain a desired overall Quality of Service (QoS) with fewer network resources. One of its key objectives is to balance loads across a network. Load balancing leads the number of overutilized links, which will have low QoS, and underutilized links, which represent a waste, to be reduced.

Traffic engineering aims to improve network performance by means of optimizing resource utilization in the network. One important issue that we need to address before we go any further is the objective of optimization. The optimization objective tends to vary depending on the specific problem that service providers are trying to solve. However, common objectives include:

- Minimizing congestion and packet losses in the network
- Improving link utilization
- Minimizing the total delay experienced by packets.
- Increasing the number of customers with the current resources.

With carefully arranged traffic trunks, service providers can spread traffic across the network to avoid congestion hot spots in the network. When we translate this into a mathematical formulation, the objective is basically to minimize the Maximum Link Utilization in a network.

When the Maximum Link Utilization (MLU) is minimized, MLU is called α in some cases, a new upper bound of the utilization in every link of the network is created (see Fig 2.1). In this way, when this new upper bound is exceeded, the information flow is transmitted by another different path. That is, all the traffic that exceeds the $(\alpha.u_{ij})$ value will be transmitted by other paths instead of using the total capacity of the links. If the traffic is multicast, instead of paths we consider trees.

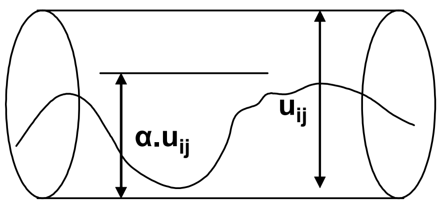


Fig 2.1. Meaning of α

Intuitively the hot spots are the points with the highest link utilization. Reducing the link utilization at these points balances the traffic distribution across the network. It turns out that when all links are utilized to the same degree, the network tends to perform at an optimal level in terms of packet losses, total delay, and bandwidth efficiency. Intuitively this is quite obvious. The queuing delay increases nonlinearly and indeed much faster as link utilization becomes higher. Thus the maximum queuing delay that packets experience will increase when traffic distribution is unbalanced.

This optimization objective of minimizing the MLU has a number of desirable features. First, as we have shown, minimizing the MLU can at the same time reduce the total delay undergone by the packets. Similarly it can also be shown that the total losses are minimized. Second, this optimization objective moves traffic away from congestion hot spots to less utilized parts of the network, so that the traffic distribution tends to be balanced. Finally, it also leaves more space for future traffic growth.

When the MLU is minimized, the percentage of the residual bandwidth in links (unused bandwidth) is also maximized. Therefore, the growth in traffic is more likely to be accommodated and can be accepted without requiring connections to be rearranged. If we assume that traffic grows in proportion to the current traffic pattern (i.e. scale up), this objective ensures that the extra traffic causes minimum congestion.

Load sharing can substantially improve the performance of a traffic engineering scheme. When traffic demands are large, any movement of a traffic demand may cause substantial shifting in the traffic between different links. When traffic demands can be split into smaller sizes, there is more flexibility in managing them. The use of load sharing in the Internet may increase substantially as Dense Wavelength Division Multiplexing (DWDM) becomes more widely used.

Traffic can be split equally among all outgoing links or in some specified proportion. The traffic splitting schemes must take into account a number of basic requirements. First, traffic splitting takes place in the packet–forwarding path and must be executed for every packet.

With multi-path forwarding, a router can have many paths to a destination. This paths can be used simultaneously to provide more bandwidth that the bandwidth of a single path. The optimized multi-path protocol is a traffic engineering extension of currently deployed link-state routing protocols, which aims at distributing load optimally based on each router having global knowledge about all link loads in the network. With this information the routers can shift traffic from congested to less congested paths and thus perform load balancing decisions. In a MPLS network MPLS ingress routers may establish one or more paths to a given egress to MPLS domain. Load can be balanced across a complex topology using MPLS

Traffic engineering in MPLS networks is similar to those arising in Asynchronous Transfer Mode (ATM) networks. To bring guaranteed QoS (lacking in connectionless IP networks), MPLS provides connection-oriented capabilities, as in ATM networks. LSPs coincide with the circuit-switched paths in ATM networks [CER04].

Several advantages of using multipath routing are discussed in [CHE01] and [IZM02]. Links do not get overused and therefore do not get congested, and so they have the potential to aggregate bandwidth, allowing a network to support a higher data transfer than is possible with any single path. Several paths can be used as backup paths for one primary path when paths are configured maximally disjoint. Each working path is protected by an alternative disjoint path. If the primary path goes down, alternative paths can quickly be deployed. Therefore, load balancing emerges as a fast response path protection mechanism. Furthermore, some authors have expanded this idea by proposing to split each flow into multiple subflows in order to achieve better load balancing [DON04], [DON03], [KIM04] and [KIM02]. The flow splitting approach can be used for the protection path. Splitting the working path has the advantage of reducing the amount to be protected [IZM02].

The per-packet overhead therefore has to be small, and to reduce implementation complexity, the system should keep no or little state information. Second, traffic–splitting schemes produce stable traffic distribution across multiple outgoing links with minimum fluctuation. Last but not least, the traffic–splitting algorithms must maintain per-flow packet ordering. Packet misordering within a TCP flow can produce a false congestion signal and cause unnecessary throughput degradation.

Simple schemes for traffic splitting are based on packet–by–packet round robin results in low overheads and good performance. They may, however, cause per–flow ordering. Sequence numbers or state may be added to reordering, but these additional mechanisms drastically increase complexity, and in many cases they only work in point–to-point links.

Hashing based traffic–splitting algorithms are stateless and easy to implement, particularly with hardware assistance. For hash functions that use any combination of the five-tuple as input, per–flow ordering can be preserved; all packets within the same TCP flow have the same five–tuple, and so the output of the hash function with the five–tuple as input should always be the same.

A simple method to divide the input traffic is on a per-packet basis, for example in a round-robin fashion. However, this method could result in excessive packet reordering and is not recommended in practice. [VIL99] tries to balance the load among multiple LSPs according to the loading of each path. In MPLS networks [ROS01] multiple paths can be used to forward packets belonging to the same "forwarding equivalent class" (FEC) by explicit routing. Once the explicit path is computed, the signaling protocol Constraint-Based Routing Label Distribution Protocol (CR-LDP) [ASH02] [ASH02a] or Resource Reservation Protocol with Traffic Engineering extension (RSVP-TE) [AWD01] is responsible for establishing forwarding state and reserve resources along the route in both last protocols. How the load is distributed between a set of alternate paths is determined by the amount of number space from a hash computation that is allocated to each path. Effective use of load balancing requires good traffic distribution schemes. In [CAO00] the performance of several hashing schemes for distributing traffic between multiple links while preserving the order of packets within a flow is studied. Although hashing-based load balancing schemes have been proposed in the past, [CAO00] is the first comprehensive study of how the schemes perform using real traffic traces. The current configurations in computer networks provide an opportunity for dispersing traffic over multiple paths to decrease congestion. In this work dispersion involves (1) *splitting* and (2) *forwarding* the resulting portions of aggregate traffic along alternate paths. The authors concentrate on (1), methods that allow a network node to subdivide aggregate traffic, and they offer a number of traffic splitting policies which divide traffic aggregates according to the desired fractions of the aggregate rate. Their methods are based on semi-consistent hashing of packets to hash regions as well as prefix-based classification [CAO00a]. The analysis of hashing methods is out of these thesis topics.

2.2.3 Load Balancing of Unicast Flows

In the Unicast case the network is modeled as a directed graph $G=(N,E)$, where N is the set of nodes and E is the set of links. We use n to denote the number of network nodes, i.e. $n=|N|$. Among the nodes, we have a source $s \in N$ (ingress node) and a destination $t \in N$ (egress node). Let $(i,j) \in E$ be the link from node i to node j. Let $f \in F$ be any unicast flow, where F is the flow set. We denote the number of flows by $|F|$. Let X_{ij}^{f} be the fraction of flow f assigned to link *(i,j)*. The problem solution, X_{ij}^{f} variables, provides optimum flow values. Let c_{ij} be the capacity of each link *(i,j)*. Let bw_f be the traffic demand of a flow *f*. The problem of minimizing $/F/$ unicast flows from ingress node s to egress node t is formulated as follows [WAN01]:

Minimize α (2.1)

Subject to

$$\alpha = \max\{\alpha_{ij}\}, \text{ where } \alpha_{ij} = \frac{\sum_{f \in F} bw_f . X_{ij}^f}{c_{ij}} \tag{2.2}$$

$$\sum_{(i,j) \in E} X_{ij}^f - \sum_{(j,i) \in E} X_{ji}^f = 1 \quad , f \in F, i = s \tag{2.3}$$

$$\sum_{(i,j) \in E} X_{ij}^f - \sum_{(j,i) \in E} X_{ji}^f = -1 \quad , i = t, f \in F \tag{2.4}$$

$$\sum_{(i,j) \in E} X_{ij}^f - \sum_{(j,i) \in E} X_{ji}^f = 0 \quad , f \in F, i \neq s, i \neq t \tag{2.5}$$

$$\sum_{f \in F} bw_f . X_{ij}^f \leq c_{ij} \quad ,(i,j) \in E \tag{2.6}$$

$$X_{ij}^f \in \Re, 0 \leq X_{ij}^f \leq 1, \alpha \geq 0 \tag{2.7}$$

The main aim is to minimize the Maximum Link Utilization (MLU), which is represented as α in (2.1). The α value is directly related to the utilization in each link (α_{ij}). The equation (2.2) is the same as the one that calculates the MLU *(α),* i.e. the maximum utilization of each of the links *(i,j)* through which a fraction of the unicast flow is being transmitted.

Constraints (2.3), (2.4) and (2.5) are flow conservation constraints. Constraint (2.3) ensures that the total flow emerging from the ingress node to egress node *t* in flow *f* is 1. It is possible to see in Fig 2.2. and Fig 2.3. that the sum of the X values leaving from the ingress node (Node 1) with a destination at the egress node (Node 4) is 1, i.e. $\left(x_{12}^f + x_{13}^f = 1\right)$.

In Figures 2.2 and 2.3 we show an example of the creation of two paths to transmit one unicast flow. This example shows an ingress node to node 1 and an egress node to node 4. In this example using the model presented in this chapter, it is possible to create two paths to transmit the unicast flow. In this case, the variable X represents the fraction of flow transmitted in each link with a destination at the egress node.

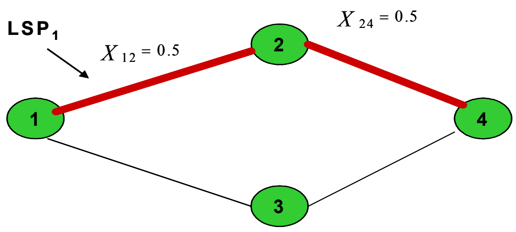


Fig 2.2. 1st Transmission Path in Unicast

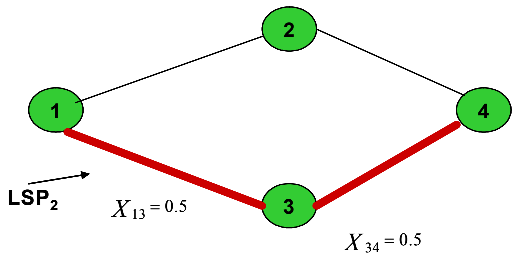


Fig 2.3. 2nd Transmission Path in Unicast

Constraint (2.4) ensures that the total flow coming from an egress node *t* of flow *f* is 1. Constraint (2.5) ensures that for any intermediate node that is different from the ingress node (i ≠ s) and egress nodes $(i \neq t)$, the sum of its output flows minus the input flows with a destination at egress node *t* of flow *f* is 0. i.e. $\left(x_{24}^{f} - x_{12}^{f} = 0\right), \left(x_{34}^{f} - x_{13}^{f} = 0\right)$.

Constraint (2.6) is the MLU constraint. The total amount of bandwidth consumed by all the flows in the link *(i,j)* must not exceed the maximum utilization (α) per link capacity c_{ij}.

Expression (2.7) shows that the X_{ij}^{f} variables must be real numbers between 0 and 1 because they represent the fraction of each flow that is transmitted. These variables form multiple trees to transport a multicast flow. The demand between the ingress node and egress node *t* may be split between multiple routes. When the problem is solved without load balancing, this variable is only able to take values 0 and 1, which show, respectively, whether or not the link *(i,j)* is being used to carry information to egress node *t*.

In [WAN01] and [LEE02] a solution for unicast transmission has been presented.

2.2.4 Taxonomy

In this section we present a taxonomy that helps to clarify and better understand the problems associated with multicast transmission, load balancing, traffic splitting and multi-objective optimization.

First, the taxonomy considers the flow type when classifying reviewed works into a traditional unicast flow type and a more general multicast flow type. Second, it categorizes load balancing techniques considering the number of paths/trees employed. For instance, if different flows going from a given source to the same set of destinations can be delivered (or not) through different paths/trees, a flow could use a path/tree while another flow could go through a different path/tree, both leave a given source node and go to the same set of destinations. Moreover, load balancing techniques are classified in relation to splitting, i.e. whether a given flow can be split into several subflows to balance loads using multiple routes. Note that splitting is only considered when a multipath / multitree is applied. Finally, optimization problems are also classified as a Single-Objective Problem (SOP) or Multi-Objective Problem (MOP) depending on how different objectives are treated, i.e. if the problem is treated with a final unique cost function, or as a set of simultaneous conflicting objective functions. A generic multi-objective optimization problem includes a set of *m* decision variables, a set of *o* objective functions and a set of *z* restrictions. The aim of multi-objective optimization is to obtain an efficient solution, in which any improvement in one objective can only be achieved at the expense of another.

However, the proposed *TE Load Balancing Taxonomy* (see Table 2.1) is very useful for presenting our model in the next chapters.

Table 2.1

Proposed TE Load Balancing Taxonomy

CLASSIFICATION PARAMETER		DESCRIPTION
Flow Types	Unicast	Transmission from one source to **one** destination using a path
	Multicast	Transmission from one source to a **set** of destinations using a tree
Number of flows	Unipath (UP) Unitree (UT)	All flows from a source to the same set of destinations travel through the **same** path / tree
	Multipath (MP) Multitree (MT)	Different flows from a source to the same set of destinations can travel through **different** paths / trees
Splitting	No	A flow always travels along the **same** path/tree
	Yes	A flow can be split into **several** subflows that can be delivered through different paths/trees
Objective Problem	Single-Objective Problem (SOP)	Only one generic cost function is considered
	Multi-Objective Problem (MOP)	Several (conflicting) objective functions may be simultaneously optimized in a multi-objective context.

2.3 Related Works with this Research

Various traffic engineering solutions using techniques that balance loads by multiple routes have been designed and analyzed in different studies that attempt to optimize a cost function subject to constraints resulting from the application's QoS requirements. Later on, we present different characteristics of some works that are related to this research.

In [RAO98] the authors consider two generic routing algorithms that plan multipaths consisting of possibly overlapping paths. Therefore, bandwidth can be reserved and guaranteed once it is reserved in the links. The first problem deals with transmitting a message of finite length from the ingress node to the egress node within r units of time. A polynomial-time algorithm is proposed and the results of a simulation are used to illustrate its applicability. The second problem deals with transmitting a sequence of some units at such a rate that the maximum time difference between the two units received out of order is limited. The authors show that this second problem is computationally intractable, and propose a polynomial-time approximation algorithm. Therefore, a Quality of Service Routing (QoSR) routing along multiple paths under a time constraint is proposed when the bandwidth can be reserved.

[RAO98] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[RAO98]	1998	Delay Bandwidth		Flow Type:	Unicast	Ford-Fulkason method
				Number of flows:	Multipath	
				Splitting:	No	
				Objective problem:	Single-Objective Problem	

In [ABO98] the authors propose a fuzzy optimization model for routing in Broadband Integrated Service Digital Network (B-ISDN) networks. The challenge of the proposed model is to find routes for flows using paths that are not hideously expensive, fulfill the required QoS and do not penalize the other flows that already exist or that are expected to arrive in the network. The model is analyzed in terms of performance in different routing scenarios. The authors obtained good improvements in performance compared with the traditional single metric routing techniques (number of hops or delay based routing). This improvement was achieved while maintaining a sufficiently low processing overhead. Throughput was increased and the probability of congestion was decreased by balancing the load over all the network links.

[ABO98] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[ABO98]	1998	Link Utilization Hop Count Delay	Bandwidth	Flow Type:	Unicast	Fuzzy logic, Weighted Sum
				Number of flows:	Multipath	
				Splitting:	No	
				Objective problem:	Single-Objective Problem	

In [FOR02] the authors propose optimizing the weight setting based on the projected demands. They show that optimizing the weight settings for a given set of demands is NP-hard, so they resort to a local search heuristic. They found weight settings that performed to within a few percent of the optimal general routing, where the flow for each demand is optimally distributed over all paths between the source and destination. This contrasts with the common belief that Open Shortest Path First (OSPF) routing leads to congestion and shows that for the network and demand matrix studied it is not possible to get substantially better load balancing by switching to the proposed more flexible MPLS technologies.

[FOR02] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[FOR02]	2002	Link Utilization	Bandwidth	Flow Type:	Unicast	Linear programming and shortest path
				Number of flows:	Multipath	
				Splitting:	No	
				Objective problem:	Single-Objective Problem	

In [SRI03] the authors propose an approach that remedies two main difficulties in optimal routing. The first is that these protocols use shortest path routing with destination based forwarding. The second is that when the protocols generate multiple equal cost paths for a given destination routing prefix, the underlying forwarding mechanism balances the load across these paths by splitting traffic equally between the corresponding set of next hops. These added constraints make it difficult or impossible to achieve optimal traffic engineering link loads. It builds links by taking advantage of the fact that shortest paths can be used to achieve optimal link loads, but it is compatible with both destination based forwarding and even splitting of traffic over equal cost paths. Compatibility with destination based forwarding can be achieved through a very minor extension to the result obtained in [WAN01a], simply by taking advantage of a property of shortest paths and readjusting traffic splitting ratios accordingly. Accommodating the constraint of splitting traffic evenly across multiple shortest paths is a more challenging task. The solution we propose stems from the fact that current day routers have thousands of route entries (destination routing prefixes) in their routing table. Instead of changing the forwarding mechanism responsible for distributing traffic across equal cost paths, we plan to control the actual (sub)set of shortest paths (next hops) assigned to routing prefix entries in a router's forwarding table(s).

[SRI03] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[SRI03]	2003	Link Utilization Bandwidth	Bandwidth	Flow Type:	Unicast	Linear programming and shortest path
				Number of flows:	Multipath	
				Splitting:	No	
				Objective problem:	Single-Objective Problem	

In [SON03] the author proposes an adaptive multipath traffic engineering mechanism called Load Distribution over Multipath (LDM). The main goal of LDM is to enhance the network utilization as well as the network performance by adaptively splitting the traffic load among multiple paths. LDM takes a pure dynamic approach that does not require any previous traffic load statistics. Routing decisions are made at the flow level and traffic proportioning reflects both the length and the load of a path. Moreover, LDM dynamically selects a few good Label Switched Paths (LSPs) according to the state of the entire network.

[SON03] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[SON03]	2003	Hop Count Bandwidth	Hop Count	Flow Type:	Unicast	(Linear) multi-commodity network flow problem
				Number of flows:	Multipath	
				Splitting:	No	
				Objective problem:	Single-Objective Problem	

In [VUT00] the authors propose a traffic engineering solution that adapts the minimum-delay routing to the backbone networks for a given long-term traffic matrix. This solution is practical and is suitable to implement in a Differential Services framework. In addition, they introduce a simple scalable packet forwarding technique that distinguishes between datagram and traffic that requires in-order delivery and forwards them accordingly and efficiently.

[VUT00] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[VUT00]	2000	Delay	Bandwidth	Flow Type:	Unicast	Non linear programming
				Number of flows:	Multipath	
				Splitting:	Yes	
				Objective problem:	Single-Objective Problem	

In [CHE01] the authors propose an algorithm to carry out the unicast transmission of applications requiring minimum bandwidth through multiple routes. The algorithm consists of five steps: a) the multipath P set is initialized as empty, b) the maximum flow graph is obtained, c) the shortest route from the ingress node to the egress node is obtained, d) the bandwidth consumption obtained in the maximum flow of step b is decreased, and e) step (d) is repeated until the required bandwidth for transmission is reached. The results presented show very similar end-to-end delay values to those obtained independently whether the load balancing is applied or not. However, link utilization is improved when load balancing is applied.

[CHE01] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[CHE01]	2001	Delay	Bandwidth	Flow Type:	Unicast	Max-flow and shortest path
				Number of flows:	Multipath	
				Splitting:	Yes	
				Objective problem:	Single-Objective Problem	

In [WAN01a] the authors present a multi-objective optimization scheme to transport unicast flows. In this scheme they consider the MLU (α) and the selection of best routes based on the flow assigned to each link. In this paper the authors consider a new approach that accomplishes traffic engineering objectives without full mesh overlaying. Instead of overlaying IP routing over the logical virtual network traffic engineering objectives such as balancing traffic distribution are achieved by manipulating link metrics for IP routing protocols such as OSPF. In this paper, they present a formal analysis of the integrated approach, and propose a systematic method for deriving the link metrics that convert a set of optimal routes for traffic demands into the shortest path with respect to the link weights that pass through them. The link weights can be calculated by solving the dual of a linear programming formulation.

[WAN01a] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[WAN01a]	2001	Link Utilization Bandwidth Flow assignation	Bandwidth	Flow Type:	Unicast	Linear Programming, Weighted Sum
				Number of flows:	Multipath	
				Splitting:	Yes	
				Objective problem:	Single-Objective Problem	

In [LEE02] the authors propose a method for transporting unicast flows. The constraint of a maximum number of hops is added to the minimization of the MLU (α). Moreover, the traffic is divided between multiple routes in a discrete way. This division simplifies implementing the solution. The behavior of five approaches are analyzed: Shortest path based on non-bifurcation, Equal Cost Multiple Paths (ECMP), Traffic bifurcation, H Hop-constrained traffic bifurcation and H Hop-constrained traffic bifurcation with node affinity. Through the approaches of Hop-constrained traffic bifurcation, a minimum value of the MLU (α) is obtained.

[LEE02] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[LEE02]	2002	Link Utilization	Hop Count Bandwidth	Flow Type:	Unicast	Mixed-integer programming
				Number of flows:	Multipath	
				Splitting:	Yes	
				Objective problem:	Single-Objective Problem	

In [ABR02] the authors propose an intra-domain routing algorithm based on multi-commodity flow optimization which allows load sensitive forwarding over multiple paths. It is not constrained by weight-tuning of the legacy routing protocols, such as OSPF, and it does not require a totally new forwarding mechanism, such as MPLS. These characteristics are accomplished by aggregating the traffic flows destined for the same egress into one commodity in the optimization and using a hash based forwarding mechanism. The aggregation also reduces computational complexity, which makes the algorithm feasible for on-line load balancing. Another contribution is the optimization objective function, which allows precise tuning of the tradeoff between load balancing and total network efficiency.

[ABR02] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[ABR02]	2002	Link Utilization		Flow Type :	Unicast	(Linear) multi-commodity network flow problem
				Number of flows:	Multipath	
				Splitting:	Yes	
				Objective problem:	Single-Objective Problem	

In [CHO03] the authors propose two multi-path constraint based routing algorithms for Internet traffic engineering using MPLS. In a normal Constraint-based Shortest Path First (CSPF) routing algorithm, there is a high probability that it cannot find a feasible path through networks for a large bandwidth constraint. This is one of the most significant constraints of traffic engineering. The proposed algorithms can divide the bandwidth constraint into two or more sub constraints and find a constrained path for each sub constraint, providing there is no single path satisfying the whole constraint. Extensive simulations show that they enhance the success probability of path setup and the utilization of network resources.

[CHO03] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[CHO03]	2003	Bandwidth	Bandwidth Subflows	Flow Type:	Unicast	Max-flow and Shortest Path
				Number of flows:	Multipath	
				Splitting:	Yes	
				Objective problem:	Single-Objective Problem	

In [CET04] the authors introduce Opportunistic Multipath Scheduling (OMS), a technique for exploiting short term variations in path quality to minimize delay, while simultaneously ensuring that the splitting rules dictated by the routing protocol are fulfilled. In particular, OMS uses measured path conditions in time scales of up to several seconds to opportunistically favor low-latency high-throughput paths. However, a naive policy that always selects the highest quality path would violate the routing protocol's path weights and potentially lead to oscillation. Consequently, OMS ensures that over longer time scales relevant for traffic management policies, traffic is split according to the ratios determined by the routing protocol. A model of OMS is developed and an asymptotic lower bound on the performance of OMS as a function of path conditions (mean, variance, and Hurst parameter) for self-similar traffic is derived.

[CET04] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[CET04]	2004	Delay Queue Size	Bandwidth	Flow Type:	Unicast	Scheduling Algorithm
				Number of flows:	Multipath	
				Splitting:	Yes	
				Objective problem:	Single-Objective Problem	

In [KIM04] the author suggests a method to improve network performance by appropriately distributing traffic in accordance with the state of the paths in a dynamic traffic pattern occurring in a short time in a multipath environment. TE using an Adaptive Multipath-forwarding (TEAM) is a traffic engineering (TE) algorithm that aims to improve network performance by properly distributing traffic in dynamic traffic patterns occurring in a short time scale. This method monitors the state of the paths by using a probe packet in the ingress node of the network, and computes the cost of the paths with monitored values. Path cost consists of weights given in the paths, such as packet delay and loss rate, the number of hops and the number of LSPs. This enables it to adapt to the state of the network without a sudden change by tracing neighboring solutions from an existing solution. Therefore, it can be seen that network performance is improved when the total cost of paths of the whole network are minimized. In addition, by distributing traffic into each interface using a table-based hashing method, the problem of ordering packets is solved.

[KIM04] Characteristics

Reference	**Year**	**Objective functions**	**Constraints**	**Taxonomy**		**Heuristic**
[KIM04]	2004	Hop Count Delay Flow assignation Number of LSPs		Flow Type:	Unicast	Shortest path
				Number of flows:	Multipath	
				Splitting:	Yes	
				Objective problem:	Single-Objective Problem	

In [SEO02] the authors propose non-bifurcation and bifurcation methods to transport multicast flows with hop-count constraints. When analyzing results and simulations, they only consider the non-bifurcation methods. The constraint of consumption bandwidth is added to the constraints considered in [RAO98]. In [LEE02] a heuristic is proposed. The proposed algorithm consists of two parts: 1) modifying the original graph to the hop-count constrained version, 2) finding a multicast tree to minimize the MLU (α).

[SEO02] Characteristics

Reference	**Year**	**Objective functions**	**Constraints**	**Taxonomy**		**Heuristic**
[SEO02]	2002	Link Utilization Bandwidth	Hop Count Bandwidth Subflows	Flow Type:	Multicast	Mixed-integer programming, Weighted Sum
				Number of flows:	Unitree	
				Splitting:	Not Applicable	
				Objective problem:	Single-Objective Problem	

In [ROY02] the authors propose a new multicast tree selection algorithm based on a non-dominated sorting technique of a genetic algorithm to simultaneously optimize multiple QoS parameters. Simulation results demonstrate that the proposed algorithm is capable of obtaining a set of QoS-based near optimal, non-dominated multicast routes within a few iterations. In this paper, the authors use a Non-dominated Sorting based Genetic Algorithm (NSGA) technique to develop an efficient algorithm which determines multicast routes on-demand by simultaneously optimizing end-to-end delay guarantee, bandwidth requirements and bandwidth utilization without combining them into a single scalar objective function.

[ROY02] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[ROY02]	2002	Delay Bandwidth Flow assignation		Flow Type:	Multicast	MOEA based on NSGA
				Number of flows:	Unitree	
				Splitting:	Not Applicable	
				Objective problem:	Multiple-Objective Problem	

In [CUI03a] the authors propose algorithm MEFPA (Multi-constrained Energy Function based Precomputation Algorithm) for a multi-constrained QoSR problem based on analyzing linear energy functions (LEF). They assume that each node s in the network maintains a consistent copy of the global network state information. This algorithm fulfills each QoS metric to b degrees. It then computes B $\left(B = C_{b+k-2}^{k-1}\right)$ coefficient vectors that are uniformly distributed in the k-dimensional QoS metric space, and constructs one LEF for each coefficient vector. Then based on each LEF, node s uses Dijkstra's algorithm to calculate a least energy tree rooted by s and a part of the QoS routing table. Finally, s combines the B parts of the routing table to form the complete QoS routing table it maintains. For distributed routing, for a path from s to t, in addition to the destination t and k weights, the QoS routing table only needs to save the next hop of each path. For source routing, the end-to-end path from s to t along the least energy tree should be saved in the routing table. Therefore, when a QoS connection request arrives, it can be routed by looking up a feasible path satisfying the QoS constraints in the routing table.

[CIU03a] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[CUI03a]	2003	Cost	Hop Count Delay Bandwidth Subflows	Flow Type:	Multicast	Shortest Path Tree
				Number of flows:	Multitree	
				Splitting:	No	
				Objective problem:	Single-Objective Problem	

In [DON03] we presented several static models which are analyzed by comparing some particular one-objective optimization functions with the multi-objective optimization function.

[DON03] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[DON03]	2003	Link Utilization Hop Count Delay Bandwidth	Bandwidth Subflows	Flow Type:	Multicast	Non Linear programming, max-flow and shortest path tree, Weighted Sum
				Number of flows:	Multitree	
				Splitting:	Yes	
				Objective problem:	Single-Objective Problem	

In addition, some master or doctoral theses have worked in the same area, for example the thesis [CER04] focuses on the multi-objective optimization of Label Switched Path design problem in MPLS networks. Minimal routing cost, optimal load-balance in the network, and minimal splitting of traffic form the objectives. The problem is formulated as a zero-one mixed integer program and aims at exploring the trade-offs among the objectives. The integer constraints make the problem NP-hard. In this thesis first both the exact and heuristic multi-objective optimization approaches are discussed, and then a heuristic framework based on simulated annealing is developed to give a solution to this proposal.

[CER04] Characteristics

Reference	Year	Objective functions	Constraints	Taxonomy		Heuristic
[CER04]	2004	Cost Link Utilization Splitting (LSPs)	Bandwidth	Flow Type:	Unicast	Simulated Annealing
				Number of flows:	Multipath	
				Splitting:	Yes	
				Objective problem:	Multi-Objective Problem	

While [RAO98], [ABO98], [SRI03], [FOR02] and [SON03] consider unicast flow, in [CHE01], [WAN01a], [LEE02], [CHO03], [ABR02], [CET04] and [VUT00] this unicast flow is split, and in [SEO02], [ROY02] and [CUI03a] the flow is multicast but not split, thus our proposal solves the traffic split ratio for multicast flows. The major differences between our work and the other multicast works are:

- First, from an objectives and function point of view, we propose a multi-objective scheme to solve the optimal multicast routing problem with some constraints, while [ROY02] and [CUI03] propose a scheme without constraints.

- Second, in relation to how many trees are used, we propose a multitree scheme to optimize resource utilization of the network, while [SEO02] and [ROY02] only propose one tree to transmit the flow information.

- Third, in relation to traffic splitting, we propose that the traffic split can transmit the multicast flow information through load balancing using several trees for the same flow, while [SEO02], [ROY02] and [CUI03a] do not propose this characteristic.

It should be noted that the proposals in [SON03], [WAN01a], [LEE02], [CHO03], [VUT00], [KIM04] and [SEO02] can be applied to MPLS networks.

Table 2.2 classifies the papers that could be considered as the most inspired for this research area. Note that [XIA99] is repeated because it considers unicast and multicast flows. The last column of Table 2.2 shows the methodologies/heuristics proposed in publications. Clearly, most publications consider the load balancing problem as a SOP, therefore, they have mainly proposed using traditional SOP heuristics such as linear programming, shortest path, non-linear programming and even evolutionary approaches such as genetic algorithms. This last evolutionary approach is dominant when considering MOPs, where all works surveyed use some kind of MOEA.

It can be seen that, initially, most papers only considered multipath (and not splitting) for unicast traffic in a single-objective context. Immediately after, multicast flow was also considered in the same single-objective context (splitting was still not considered). In the year 2000 papers considering splitting began to appear. Lately, the multiobjective context of the TE problem has slowly been recognized [ROY02], with an increased number of publications since 2003 (see Table 2.2 in bold for the aims of this thesis).

Table 2.2

Proposed taxonomy applied to reviewed papers including objective functions, constraints and heuristics

Reference	Year	Objective functions (OF)										Constraint (C)				Taxonomy				Heuristic
		Link Utilization	Hop Count	Delay	Bandwidth	Flow assignation	Packet Loss	Queue Size	Number of LSPs	Jitter	Cost	Hop Count	Delay	Bandwidth	Subflows	Flow Type	Number of flows	Splitting	Objective problem	
[XIA99]	1999			X	X		X			X				X	X	Unicast	UP	NA	SOP	Genetic algorithms, Weighted Sum
[KOY04]	2004			X							X								MOP	MOEA
[RAO98]	1998			X	X												MP	No	SOP	Ford-Fulkason method
[ABO98]	1998	X	X	X										X						Fuzzy logic, Weighted Sum
[FOR02]	2002	X												X						Linear programming and shortest path
[SRI03]	2003	X			X									X						Linear programming and shortest path
[SON03]	2003		X		X							X								(Linear) multi-commodity network flow problem
[VUT00]	2000			X										X				Yes	SOP	Non Linear programming
[CHE01]	2001			X										X						Max-flow and shortest path
[WAN01a]	2001	X			X	X								X						Linear programming, Weighted Sum
[LEE02]	2002	X										X		X						Mixed-integer programming
[ABR02]	2002	X																		(Linear) multi-commodity network flow problem
[CHO03]	2003				X									X	X					Max-flow and Shortest Path
[CET04]	2004			X				X						X						Scheduling algorithm
[KIM02] [KIM04]	2002 2004		X	X		X			X											Shortest Path
[CER04]	2004	X							X		X			X					MOP	Simulated Annealing
[XIA99]	1999			X	X		X			X				X	X	Multicast	UT	NA	SOP	Genetic algorithms, Weighted Sum
[LEU98] [INA99] [LI99]	1998 1999 1999										X									Genetic algorithms
[SUN99]	1999										X		X							Genetic algorithms
[BAN01]	2001			X	X															Genetic algorithms, Weighted Sum
[SEO02]	2002	X			X							X		X	X					Mixed-integer programming, Weighted Sum
[ROY02]	2002			X	X	X													MOP	MOEA based on NSGA
[CUI03]	2003			X	X		X						X							MOEA
[ROY04]	2004			X	X		X													MOEA based on NPGA
[CRI04]	2004			X							X									MOEA based on SPEA
[CRI04a] [CRI04b]	2004	X		X							X									MOEA based on SPEA
[CUI03a]	2003										X						MT	No	SOP	Shortest Path Tree
[LAY04]	2004			X	X		X			X	X									Genetic algorithms, Weighted Sum
[POM04]	2004										X									DIMRO Heuristic
[FAB04]	2004	X	X	X	X														MOP	MOEA based on NSGA
[DON04] [DON03] [FAB04a] [DON04a] [DON04b]	2004 2003 2004 2004 2004	X	X	X	X									X	X			Yes	SOP	Non-linear programming, max-flow and shortest path tree, Weighted Sum
[BAR04]	2004	X	X	X	X	X		X	X	X				X	X				MOP	MOEA based on SPEA

UP: Unipath UT: Unitree NA: Not Applicable SOP: Single-Objective Problem

MP: Multipath MT: Multitree MOP: Multi-Objective Problem

TE Load Balancing Taxonomy proposed (see Table 2.1) could be extended with other kind of concepts like whether the connections are static or dynamic. All works presented previously (Table 2.2) consider static case. With respect to the dynamic case, some works have been realized.

One of the important distinctions between unicast and multicast connections is the possibility of connection dynamics. Now, two categories of multicast routing algorithms, static and dynamic are identified. If the network topology does not change and that group membership is fixed during a session, static algorithm is sufficient. However, in a real network environment, network links and nodes can fail (or be removed) or be recovered (or be added) frequently. In addition, the group membership can change dynamically during a multicast session. It is essential to design efficient dynamic multicast routing algorithm operating under dynamic network environment. This variability adds further complexity to the already difficult problem of traffic engineering. This kind of problem is known as a dynamic Steiner tree problem [IM95]. The main different between the static and dynamic cas in multicast transmission is that in the static case, the group of egress nodes is fixed during set up and the identities of all egress nodes are available simultaneously. Once a static group has been established, individual membership remains unmodified until it is discarded. Paths from the ingress node to all egress nodes are computed at the same time. In the dynamic case, the group of egress nodes can change during the connection and the identities of the egress nodes are revealed one by one. Note that the dynamic problem can be reduced to the static problem, if the multicast tree is recomputed from scratch each time there is a membership change. But, since the optimal solution obtained is ephemeral, because of the dynamic nature of multicast connections, the computation of an optimal tree for each membership group change may not be the best way forward. In the optimization process it is possible to have a solution like the best path to the new egress node is a direct connection between the ingress node to the egress node.

There are two main design objectives for the dynamic problem. The first is to minimize the computational complexity required to update a multicast tree. The second is to maintain routing stability by making minimal changes to the topology of an existing multicast tree, because multicast sessions cannot tolerate the disturbances and disruptions caused by excessive changes. As recomputing the multicast tree from scratch is computationally expensive, it makes sense to compute a near-optimal multicast tree which is minimally disturbed after each change in the membership group.

If designing an optimal tree is a complex problem, maintaining this tree optimality after changes in the membership group may be even more complex. The GREEDY algorithm is a simple, non-rearrangeable heuristic proposed by Vaxman, with the aim of minimizing the perturbation to the existing tree. To add a node, i.e. Nnew, to an existing multicast group, a closest node, i.e. Ntree, already in the tree is chosen and Nnew is attached to the Ntree via the least cost path. For a delete request, if the node being removed is a leaf node, then the branch of the tree supporting only that node is pruned. For the case of a nonleaf node, no action is taken.

In [STR02] present a multicast "life cycle" model that identifies the various issues involved in a typical multicast session. During the life cycle of a multicast session, three important events can occur: group dynamics, network dynamics and traffic dynamics. The first two aspects are concerned with maintaining a good quality (e.g., cost) multicast tree taking into account member join/leave and changes in the network topology due to link/node failures/additions, respectively. The third aspect is concerned with flow, congestion, and error control. In this paper they examine various issues and solutions for managing group dynamics and failure handling in QoS multicasting, and outline several future research directions.

The dynamic case in multicast transmission can be worked in different ways. In [STR02] the authors are identifying the different ways, in the Figure 2.4 are shown.

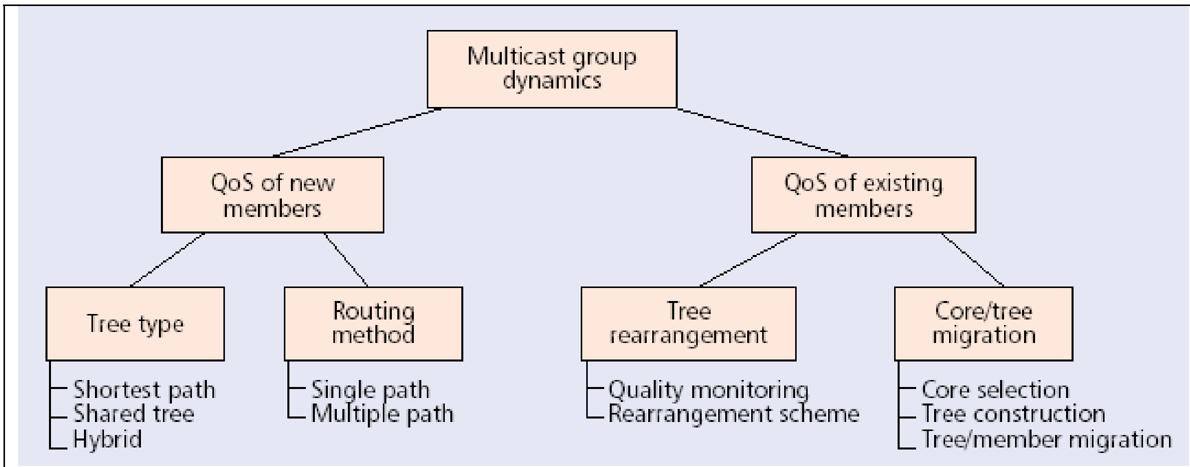


Fig 2.4 Issues in multicast group dynamics

In this thesis, in the dynamic case, we are working with QoS of new members. Te tree type are build through a combination of shortest path, maximum flow and breadth first search probabilistic algorithms. We respect to the routing method we are working with multiple path.

In [RAG99] the authors propose the problem of modifying such a delay-constrained multicast tree, when new nodes enter or existing members leave the multicast group. They present a new algorithm called controlled rearrangement for constrained dynamic multicasting (CRCDM) for on-line updating of delay-constrained multicast trees with the aim of minimizing the cost of constructing the trees.

In [TRA03] address the problem of rearranging a part of the existing multicast tree, so that the tree cost is reduced, while the source-to-destination delay and inter-destination delay constraints remain satisfied. They propose a genetic algorithm for which the tradeoff between tree cost and running time is tunable.

In [ALO02] the authors propose an efficient on-line estimation algorithm for determining the size of a dynamic multicast group. By using diffusion approximation and Kalman filter, they derive an estimator that minimizes the mean square of the estimation error. As opposed to previous studies, where the size of the multicast group is supposed to be fixed throughout the estimation procedure, they consider a dynamic estimation scheme that updates the estimation at every observation step. The robustness of our estimator to violation of the assumptions under which it has been derived is addressed via simulations. Further validations of our approach are carried out on real audio traces.

In [CHA03] the authors propose a QoS-based routing algorithm for dynamic multicasting. The complexity of the problem can be reduced to a simple shortest path problem by applying a Weighted Fair Queuing (WFQ) service discipline. Using a modified Bellman–Ford algorithm, the proponed routing builds a multicast tree, where a node is added to the existing multicast tree without re-routing and satisfying QoS constraints. With user defined life-time of connection this heuristic algorithm builds multicast tree which is near optimum over the whole duration of session. Simulation results show that tree costs are nearly as good as other dynamic multicast routings that does not consider QoS.

2.4 Conclusions and Motivations

In this section, we have presented some fundamental concepts and different related works with this research. The concepts presented in this chapter allow us to put our research into a context and understand the different related works helping us to make some proposals for multicast transmissions in chapters 3, 4, 5 and 6.

Chapter 3 Load Balancing of Static Multicast Flows

3.1 Overview

Although different models have been defined, which fulfill load balancing in unicast transmission, it is necessary to define other models that consider multicast transmission, because in this case, instead of creating multiple paths to transmit the flow from the ingress node to just one egress node it is necessary to create multiple trees to transport the flow from the ingress node to the egress node set of the multicast group.

In this chapter we propose a load-balancing scheme using several objectives to create multiple trees based on weighting methods. The scheme includes the maximum link utilization (MLU), the hop count (HC), the total bandwidth consumption (BC), and the total end-to-end delay (DL). In this model, we have included a constraint with the aim of solving the scalability problem, because without this constraint it would be possible to create too many trees.

3.2 Optimization Scheme

In this multi-objective proposal the network is modeled as a directed graph $G = (N, E)$, where N is the set of nodes and E is the set of links. We use n to denote the number of network nodes, i.e. $n = |N|$. Among the nodes, we have a source $s \in N$ (ingress node) and some destinations T (the set of egress nodes). Let $t \in T$ be any egress node. Let $(i, j) \in E$ be the link from node i to node j. Let $f \in F$ be any multicast flow, where F is the flow set and T_f is the egress node subset for the multicast flow f. We use $|F|$ to denote the number of flows. Note that $T = \bigcup_{f \in F} T_f$.

Let X_{ij}^{tf} be the fraction of flow f to egress node t assigned to link (i,j). Note that these variables include egress node t, which is not considered in previous works [KIM02] [LEE02] [ROY02] [SEO02] [WAN01]. Including the egress nodes permits us to control the bandwidth consumption in each link with a destination at the set of egress nodes. Therefore, it is possible to maintain the flow equilibrium constraint at the intermediate nodes exactly. The problem's solution, X_{ij}^{tf} variables, provides optimum flow values.

Let c_{ij} be the capacity of each link (i,j). Let bw_f be the traffic demand of a flow f from the ingress node s to T_f. The binary variables, Y_{ij}^{tf}, represent whether link (i,j) is being used (1) or not (0) for the multicast tree rooted at ingress node s and reaching egress node subset T_f. Let v_{ij} be the propagation delay of link (i,j). Let m be the number of functions in the multi-objective function. Let $connection_{ij}$ be the indicator of whether there is a link between nodes i and j. The Table 3.1 presents the variable definitions.

Table 3.1.
Variable Definitions

Terms	Definition
G	Graphs of the topology
N	Set of nodes
E	Set of links
s	Ingress node
T	Set of egress nodes
t	Any egress node
(i,j)	Link from node i to node j
F	The flow set
f	Any multicast flow
T_f	The egress node subset for the multicast flow f
X_{ij}^{tf}	The fraction of flow f to egress node t assigned to link (i,j)
c_{ij}	The capacity of each link (i,j)
bw_f	the traffic demand of a flow f
Y_{ij}^{tf}	Represent whether link (i,j) is being used (1) or not (0) to transmit the flow f to egress node t
m	The number of functions in the multi-objective function
$connection_{ij}$	The indicator of whether there is a link between nodes i and j
N_T	The maximum number of necessary links

The problem of minimizing $|F|$ multicast flows from ingress node s to the egress nodes of each subset T_f is formulated as follows:

(MHDB-S model)

Minimize

$$r_1.\alpha + r_2 \sum_{f\in F} \sum_{t\in T_f} \sum_{(i,j)\in E} Y_{ij}^{tf} + r_3 \sum_{f\in F} \sum_{(i,j)\in E} bw_f \max_{t\in T_f}\left(X_{ij}^{tf}\right) + r_4 \sum_{f\in F} \sum_{t\in T_f} \sum_{(i,j)\in E} v_{ij} Y_{ij}^{tf} \tag{3.1}$$

Subject to

$$\alpha = \max\{\alpha_{ij}\}, \text{ where } \alpha_{ij} = \frac{\sum_{f\in F} bw_f . \max_{t\in T_f}(X_{ij}^{tf})}{c_{ij}} \tag{3.2}$$

$$\sum_{(i,j)\in E} X_{ij}^{tf} - \sum_{(j,i)\in E} X_{ji}^{tf} = 1 \quad ,t\in T_f, f\in F, i = s \tag{3.3}$$

$$\sum_{(i,j)\in E} X_{ij}^{tf} - \sum_{(j,i)\in E} X_{ji}^{tf} = -1 \quad ,i,t\in T_f, f\in F \tag{3.4}$$

$$\sum_{(i,j)\in E} X_{ij}^{tf} - \sum_{(j,i)\in E} X_{ji}^{tf} = 0 \quad ,t\in T_f, f\in F, i\neq s, i\notin T_f \tag{3.5}$$

$$\sum_{f\in F} bw_f . \max_{t\in T_f}(X_{ij}^{tf}) \leq c_{ij} \quad ,(i,j)\in E \tag{3.6}$$

$$\sum_{j\in N} Y_{ij}^{tf}{}_{t\in T_f} \leq N_T, i\in N, f\in F \tag{3.7a}$$

$$\sum_{j\in N} Y_{ij}^{tf}{}_{t\in T_f} \leq \left\lceil \frac{bw_f}{\left\lfloor \sum_{j\in N} c_{ij} \Big/ \sum_{j\in N} connection_{ij} \right\rfloor} \right\rceil \quad ,i\in N, f\in F \tag{3.7b}$$

where

$$X_{ij}^{tf} \in \Re, 0 \leq X_{ij}^{tf} \leq 1 \tag{3.8}$$

$$Y_{ij}^{tf} = \left\lceil X_{ij}^{tf} \right\rceil = \begin{cases} 0 & ,X_{ij}^{tf} = 0 \\ 1 & ,0 < X_{ij}^{tf} \leq 1 \end{cases} \tag{3.9}$$

$$\sum_{i=1}^{m} r_i = 1, \quad r_i \in \Re, \quad r_i \geq 0, m > 0 \tag{3.10}$$

The Multi-objective function (MHDB-S model) (equation 3.1) defines a function and generates a single aggregated metric for a combination of weighting objectives.

The main objective consists in minimizing the maximum link utilization (MLU), which is represented by α in (3.1). The value of α is directly related to the utilization in each link *(i,j)*.

However with this objective the solution obtained may involve long routes. In order to eliminate these routes and to minimize hop count (HC), the term $\sum_{f \in F} \sum_{t \in T_f} \sum_{(i,j) \in E} Y_{ij}^{tf}$ is added. This is necessary because the objective function may report only the most congested link and the optimal solution may include unnecessarily long paths in order to avoid the bottleneck link [KIM02].

In addition, in the hop count function it is possibly to optimize other objective functions, for example, in order to minimize the total bandwidth consumption (BC) over all links, the term $\sum_{f \in F} \sum_{t(i,j) \in E} bw_f \max_{t \in T_f} \left(X_{ij}^{tf} \right)$ is also added. Remember that in a unicast connection, the total amount of bandwidth consumed by all the flows with a destination at egress node t must not exceed the maximum utilization (α) per link capacity c_{ij}, that is, $\sum_{f \in F} bw_f \sum_{t \in T} X_{ij}^{tf} \leq c_{ij} \cdot \alpha, (i,j) \in E$. However in a multicast connection it is necessary to consider only the maximum value because the same packet is never sent twice in a link.

Furthermore, in order to minimize the total end-to-end propagation delay (DL) over all links, the term $\sum_{f \in F} \sum_{t \in T_f} \sum_{(i,j) \in E} v_{ij} \, Y_{ij}^{tf}$ is also added. In [ABO98] the authors showed that the delay has three basic components: switching delay, queuing delay and propagation delay. The switching delay is a constant value and can be added to the propagation value. The queuing delay is already reflected in the bandwidth consumption. The authors state that the queuing delay is used as an indirect measure of buffer overflow probability (to be minimized). Other computational studies (e.g. [ABO98]) have shown that it makes little difference whether the cost function used in routing includes the queuing delay or the much simpler form of link utilization.

Equation (3.2) calculates the maximum link utilization (MLU), also called α*,* in function of the maximum utilization in every link *(i,j)* in which a fraction of the multicast flow is being transmitted.

Constraints (3.3), (3.4) and (3.5) are flow conservation constraints. Constraint (3.3) ensures that the total flow emerging from the ingress node to any egress node t at flow f is 1. In Fig 3.1

and Fig 3.2 we show that the sum of the X values leaving the ingress node (Node 1) with destinations at every egress node (Nodes 5 and 6) is 1. In the Figures 3.1 and 3.2 we show an example of the creation of two trees to transmit one multicast flow. In this example the ingress node is node 1 and the egress nodes are nodes 5 and 6. In this example using the proposal presented in this chapter, it is possible to create two trees to transmit the multicast flow. In this case, the variable X represents the fraction of flow transmitted in every link with a destination at one particular egress node.

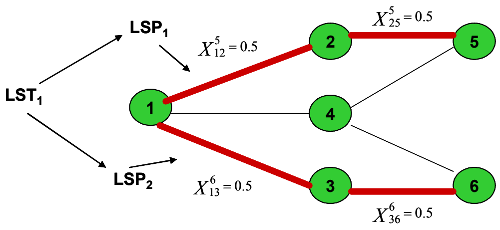


Fig 3.1. 1st Transmission Tree in Multicast

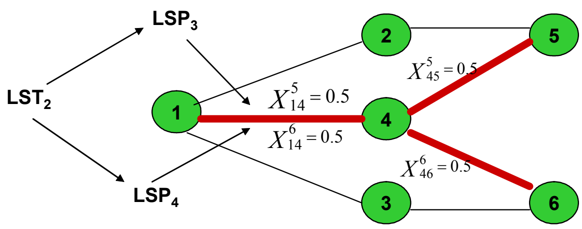


Fig 3.2. 2nd Transmission Tree in Multicast

Constraint (3.4) ensures that the total flow coming from an egress node t at flow f is 1. Constraint (3.5) ensures that for any intermediate node that is different from the ingress node (i ≠ s) and egress nodes $(i \notin T)$, the sum of its output flows to egress node t minus the input flows with a destination at egress node t at flow f is 0. i.e. $\left(x_{25}^5 - x_{12}^5 = 0\right), \left(x_{36}^6 - x_{13}^6 = 0\right), \left(x_{45}^5 - x_{14}^5 = 0\right), \left(x_{46}^6 - x_{14}^6 = 0\right).$

Constraint (3.6) is the maximum link utilization constraint. Remember that in an unicast connection, the total amount of bandwidth consumed by all the flows with a destination at egress node t must not exceed the maximum utilization (α) per link capacity c_{ij}, that is, $\sum_{f \in F} bw_f \sum_{t \in T} X_{ij}^{tf} \le c_{ij} . \alpha, (i,j) \in E \cdot$

Nevertheless, in constraint (3.6) only the maximum value of X_{ij}^{tf} for $t \in T_f$ must be considered. Though several subflows of flow f in link *(i,j)* with destinations at different egress nodes are sent, in multicast IP specification just one subflow will be sent.

The function *max* in expressions (3.1), (3.2) and (3.6) generates discontinuous derivatives. For this reason, the problem should be solved using a GAMS tool to solve DNLPs (Nonlinear programming with discontinuous derivatives) such as MINOS, MINOS5, COMOPT, COMOPT2, and SNOPT [GAM04]. The DNLP problem is the same as the NLP (Nonlinear Programming) problem, except that non-smooth functions (abs, min, max) can appear.

Our model is a nonlinear programming problem because some equations use the nonlinear function max (3.1), (3.2), (3.6). Usually the models can be relaxed by introducing a new constraint variable which reaches the greatest value of the *max* function. One way to give a solution to this kind of problem is the linearization. For example: min y, where y = max x_i, such model can be rewritten as: min y, where y ≥ x_i. But sometimes, such relaxation can introduce new variables in the model and this kind of solution (called e-constraint) has problems when in multi-objective optimization problem the solution space is not convex: Moreover, another difficult it is to know the maximum values of these new constraints. This kind of method to convert a nonlinear problem into a linear problem is good when the problem is a single-objective problem. And this kind of solution is good when it is impossible to have a nonlinear solver.

The constraints (3.7a) and (3.7b) limit the maximum number of subflows in each node by means of the capacity of each link and the traffic demand. This formulation represents the number of links necessary for a traffic demand, without this constraint the model could have scalability problems, i.e. the label space used by the LSPs would be too high.

A first approach towards achieving this aim is to use constraint (3.7a), in this case, the maximum number of necessary links is given by a constant value N_T [DON03]. However, this expression has a problem; what is the right value of N_T? To solve this drawback constraint (3.7b), which depends on network characteristics (flow demand, bandwidth in every link and number of connections in every node), is defined.

Expression (3.8) shows that the X_{ij}^{tf} variables must be real numbers between 0 and 1 because they represent the fraction of every flow that is transmitted. These variables form multiple trees to transport multicast flow. The demand between the ingress node and egress node *t* can be split over multiple routes. When the problem is solved without load balancing, this variable can only take the values 0 and 1, which show, respectively, whether or not link *(i,j)* is being used to carry information to egress node *t*.

Expression (3.9) calculates Y_{ij}^{tf} as a function of X_{ij}^{tf}. Note that the variables Y_{ij}^{tf} are integers.

Finally, expression (3.10) shows that the weighting coefficients, r_i, assigned to the objectives are normalized. These values are used to solve the optimization problem.

The problem presented is *NP-hard* because the problem of computing the minimum cost tree for a given multicast group is known as a Steiner tree problem (*NP-complete*) and this model includes constrained integers (Y_{ij}^{tf}) and real variables (X_{ij}^{tf}).

Therefore, it is necessary to develop an algorithm to solve the same problem in a considerable polynomial time. In the next sub-section, we present an algorithm that gives an approximate solution to the problem presented in this sub-section.

3.3 Static Multicast Multitree Routing Algorithm (MMR-S)

In this section we propose a heuristic algorithm called MMR-S to solve the multiobjective MHDB-S model proposed in equation (3.1). This algorithm, shown in Algorithm 3.1, has $G(N,E)$, s, Tf, bw_f and r as parameters, where G is the network topology, N is the node set, E is the links set, s is the ingress node for flow f, T_f is the egress node subset, bw_f is the transmission rate for flow f, and r is the weighting objectives vector. Let d_i^{tf} denote the length of a SPT from the ingress node to the egress nodes T_f. Let λf be a tree from s to t, $t \in T_f$, associated to flow f, which consists of several paths P_k^{tf} with $1 \le k \le h$, where h is the number of paths. This algorithm consists of two steps: 1) obtaining graph G' with the distance based on the hop count, bandwidth consumption and delay, and 2) finding the multicast tree.

Table 3.2.

Variable Definitions

Terms	Definition
d_i^{tf}	Denote the length of a SPT from the ingress node to the egress nodes T_f
P_k^{tf}	The paths found between ingress node s to the egress node t in the flow f
λf	A tree from s to t
G'	Graph with the distance based on the hop count, bandwidth consumption and delay

Step 1) Obtaining modified graph G'. In this step, all possible paths between ingress node s and every egress node t, $t \in T_f$ are looked for. This step consists of three nested loops which calculate (for each egress node, $t \in T_f$, for each path P_k^{tf} with a destination at node t and for

each node i, $i \in P_k^{tf}$) the distance value d_i^{tf} based on the hop count, bandwidth consumption and delay. In this case, a Breadth-First Search (BFS) is used to look for the different paths. In this type of search, the search proceeds by generating and testing each node that is reachable from a parent node before it expands any of these children. This search is said to be exhaustive because the search is guaranteed to generate all reachable states before it terminates with failure. By generating the all of the nodes at a particular level before proceeding to the next level of the tree (the Breadth First Strategy) this control regime guarantees that the space of possible moves will be systematically examined.

Step 2) Finding the multicast tree. The second step consists of finding for each egress node $t \in T_f$ the paths required to transmit the flow of information according to graph *G'* with its distance variables, d_i^{tf}, and the available capacity of each path. To find out the cost of the path, the r_i values obtained with SNOPT (see Table 3.3) in the solution of the multi-objective function of the MHDB-S model are taken into account. From among the paths that have a capacity greater than zero, the path with the lowest cost is selected and the maximum possible flow is applied in accordance with the maximum link utilization constraint (3.6).

For a simple case with just one egress node (as in the topology shown in Fig 3.3), Fig 3.4 shows the values of the distances d_i^{tf} obtained in the first run of the algorithm.

Table 3.3.

The weighting objective vector r

r1	r2	r3	r4
0.997925	0.001	0.001	0.000075

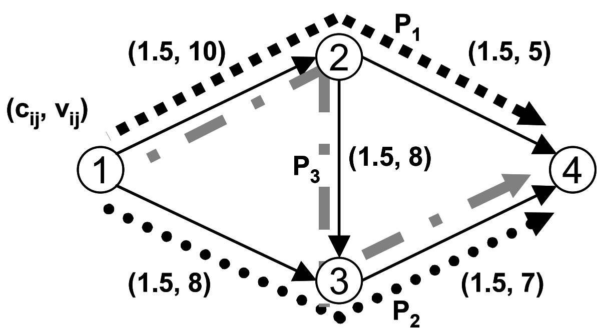


Fig 3.3. Graph G

Dist.	Path: nodes	HC (hops)	DL (mseg)	BC (Mbps)
d_1	----	0	0	0
d_2	P1: {1,2}	1	10	1.5
	P3: {1,2}	1	10	1.5
d_4	P1: {1,2,4}	2	15 = 10+5	1.5
	P2: {1,3,4}	2	15 = 8+7	1.5
	P3: {1,2,3,4}	3	25 = 10+8+7	1.5
d_3	P2: {1,3}	1	8	1.5
	P3: {1,2,3}	2	18 = 10+8	1.5

Fig 3.4. Distances labeled by paths

algorithm MMR-S (G(N,E),s,Tf,bwf,r):

begin

$X_{ij}^{tf} \leftarrow 0$

// Step1: (Obtain modified graph G') //

for each ($t \in T_f$) do

while (exists a path P_k^{tf} from node sf to node t) do

for each ($i \in P_k^{tf}$) do

$d_i^{tf} \leftarrow \{\alpha_p, HC_p, DL_p, BC_P\}$;

endfor

endwhile

endfor

α ← min αp

// Step2: (Finding multicast tree) //

for each ($t \in T_f$) do

while (bwf > 0 &&

X_{ij}^{tf} ← identify a path P_k^{tf} in d_i^{tf} from node sf to node t /

min (r1*α+r2*HCP+r3*DLP+r4*BCP) ≠ 0)) do

δ ← min{max_flowij, (i,j)Є P_k^{tf} } ;

increase δ units of flow along P_k^{tf} and update G';

bwf ← bwf – δ;

endwhile

endfor

end algorithm;

Algorithm 3.1. Algorithm MMR-S (Static Multicast Multi-Tree Routing)

The X_{ij}^{tf} values resulting from the algorithm allow us to calculate the values of MLU, HC, DL and BC according to the multi-objective (3.1) of the MHDB-S model.

3.4 Conclusions and Motivations

Although different models have been defined to carry out load balancing in unicast transmission, it is necessary to define models that consider multicast traffic because in this case, instead of creating multiple paths to transmit flow from an ingress node to a single egress node, it is necessary to create multiple trees to transmit flow from the ingress node to the egress node set of the multicast group.

In this section, we have presented multi-tree routing in order to develop multicast transmission with load balancing using multiple trees. We have employed a multi-objective, load-balancing scheme to minimize: the maximum link utilization (α), the hop count (HC), the total bandwidth consumption (BC), and the total end-to-end delay (DL). By introducing HC, lengthy paths are eliminated. By introducing BC, the bandwidth consumed by links is minimized. Using DNLP mathematical programming, we obtained the optimal set of X_{ij}^{tf} (the fraction of flow *f* with a destination at node *t* assigned to link *(i,j)*) for the problem (MHDB-S). The optimization variables (MLU, HC, DL and BC values) are calculated using a GAMS solver called SNOPT and the results will be compared with the heuristic algorithm presented.

The optimization scheme proposed in section 3.2 has the following limitations:

1. In multicast transmissions the egress nodes can enter or leave the group in transmission time.
2. It is necessary to define the label assignation for the information transmission in MPLS.
3. The solution to the multi-objective problem is obtained through a lineal combination of the different objectives and therefore it is transformed into a single-objective function to be optimized.
4. The method used in section 3.2 to solve the optimization problem would present some problems when the feasible solution set is a non-convex set, since some solutions cannot be found.

To solve these drawbacks several proposals are presented in the following chapters. A solution to dynamic nodes in multicast transmission is presented in chapter 4. A solution to the label assignation problem in MPLS is presented in chapter 5. In addition, in chapter 6 a new scheme for finding a solution to points 3 and 4 using Multi-Objective Evolutionary Algorithms (MOEA) is presented.

3.5 Papers Published with this Chapter

International Journals

- [DON05] Y. Donoso, R. Fabregat, JL. Marzo. "Multicast Routing With Traffic Engineering: A Multi-Objective Optimization Scheme And A Polynomial Shortest Path Tree Algorithm With Load Balancing." Annals of Operations Research. Kluwer Publisher. 2005.

- [DON04c] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Algorithm for Multicast Routing with Traffic Engineering." Telecommunication Systems Journal. Kluwer Publisher. 2004.

International Conference

- [FAB04a] R. Fabregat, Y.Donoso, JL. Marzo, A. Ariza. "A Multi-Objective Multipath Routing Algorithm for Multicast Flows." International Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'04). San Jose, USA. July 2004.

- [DON04] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Algorithm for Multicast Routing with Traffic Engineering". IEEE 3rd International Conference on Networking ICN'04, Guadeloupe, French Caribbean, March 2004. **Selected as Best Papers**.

- [DON04d] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Model and Heuristic Algorithm for Multipath Routing of Static and Dynamic Multicast Group." III Workshop on MPLS Networks. Girona, Spain. March 2004.

- [DON04e] Y. Donoso, R. Fabregat, JL. Marzo. "Multicast Routing with Traffic Engineering: a Multi-Objective Optimization Scheme and a Polynomial Shortest Path Tree Algorithm with Load Balancing". Proceedings of CCIO -2004 Cartagena de Indias, Colombia, March 2004. **Operation Research National Award given by the Colombian Operations Research Society. March 2004.**

- [DON03] Y. Donoso, R. Fabregat. "Multi-Objective Scheme over Multi-Tree Routing in Multicast MPLS Networks". IFIP/ACM Latin America Networking Conference 2003 (LANC03), IFIP-TC6 and ACM SIGCOMM, La Paz (Bolivia). October 2003.

In the next paper a simplified model for solving the SOP (single-objective problem) for minimizing α is presented.

- [DON03a] Y. Donoso, R. Fabregat. "Ingeniería de Tráfico aplicada a LSPs Punto-Multipunto en Redes MPLS". CLEI (Congreso Latinoamericano de Estudios en Informática) 2003. La Paz, Bolivia. October 2003.

Chapter 4 Load Balancing of Dynamic Multicast Flows

4.1 Overview

In this chapter, we propose a multi-objective traffic engineering scheme using different distribution trees for dynamic multicast groups (i.e. in which egress nodes can change during the connection's lifetime). If a multicast tree is recomputed from scratch, it may consume a considerable amount of CPU time and all communications using the multicast tree will be temporarily interrupted. To overcome these drawbacks we propose an optimization model (dynamic model MHDB-D) that uses a previously computed multicast tree (static model MHDB-S) adding new egress nodes. Similarly to the MHDB-S model, our aim in the MHDB-D model is to combine the following weighting objectives into a single aggregated metric: maximum link utilization, hop count, total bandwidth consumption and total end-to-end delay. In the same way, our dynamic proposal (MHDB-S model) also solves the traffic split ratio for multiple paths. In this case they are multiple paths and not multiple trees because a new node is added only in the flow transmission. Finding a solution for the analytical MHDB-D model using the different equations is very complicated. A multicast multi-objective dynamic routing algorithm called MMR-D is proposed for optimizing the different objectives.

4.2 Optimization Scheme

In this section, the MHDB-S model presented in the previous section is extended to the case of dynamic membership because using the MHDB-S model to obtain the optimal multicast tree after each change is computationally expensive. The solution is called the MHDB-D model. This model is based on adding the new egress nodes one by one to the tree. In this case, in order to connect the new egress node to the MHDB-S tree solution, only paths from the ingress node or the egress nodes are considered. The nodes that can become ingress nodes of the new node in the dynamic model must receive 100% of the flow information f. On the other hand, when the new node is connected to the existing tree the MHDB-D model considers the multipaths found previously, firstly by using the MHDB-S model and subsequently by using the previous MHDB-D model. In both cases, the multipath that optimizes MLU, HC, DL and

BC is chosen. With this scheme, the existing paths to the egress nodes can be used from that point to continue creating or extending new paths to connect to the new egress node.

For the MHDB-D model the network is modeled as a directed graph $G=(N,E)$, where N is the set of nodes and E is the set of links. We use n to denote the number of network nodes, i.e. $n=|N|$. Among the nodes, we have a source $s \in N$ (ingress node) and some destinations T (the set of egress nodes in the initial transmission). Let $(i,j) \in E$ be the link from node i to node j. Let $f \in F$ be any multicast flow, where F is the flow set and T_f is the egress node subset for the multicast flow f. We use $|F|$ to denote the number of flows. Let $T=\bigcup_{f \in F} T_f$. Let bw_f be the traffic demand of a flow f from the ingress node s to T_f. Let v_{ij} be the propagation delay of link (i,j). Let m be the number of variables in the multi-objective function. Let $connection_{ij}$ be the indicator of whether there is a link between nodes i and j. These variables that have been presented are the same as the MHDB-S model but have been included for completeness.

We will now extend the variables and functions defined in the previous section to define the new dynamic model. Let t' be the new egress node of the egress node subset (the membership group) for the multicast flow f. Let $t' \in T_f'$, $T_f' = T_f \cup \{t'\}$. Let $Xd_{ij}^{t'f}$ be the fraction of flow f to the new egress node t' assigned to link (i,j). The binary variables, $Yd_{ij}^{t'f}$, represent whether link (i,j) is being used (1) or not (0) for the new egress node t' in flow f. Let c_{ij}' be the residual capacity of each link (i,j), which is calculated by decreasing the used bandwidth in the static model through X_{ij}^{tf} variables and flow demand f.

The Table 4.1 presents the variable definitions.

Table 4.1.

Variable Definitions

Terms	Definition
G	Graphs of the topology
N	Set of nodes
E	Set of links
s	Ingress node
T	Set of egress nodes
t	Any egress node

t'	The new egress node of the egress node subset (the membership group) for the multicast flow *f*
(i,j)	Link from node *i* to node *j*
F	The flow set
f	Any multicast flow
T_f	The egress node subset for the multicast flow *f*
X_{ij}^{tf}	The fraction of flow *f* to egress node *t* assigned to link *(i,j)*
$Xd_{ij}^{t'f}$	The fraction of flow *f* to the new egress node *t'* assigned to link *(i,j)*
c_{ij}	The capacity of each link *(i,j)*
$c_{ij}^{'}$	The residual capacity of each link *(i,j)*
bw_f	the traffic demand of a flow *f*
Y_{ij}^{tf}	Represent whether link *(i,j)* is being used (1) or not (0) to transmit the flow *f* to egress node *t*
$Yd_{ij}^{t'f}$	Represent whether link *(i,j)* is being used (1) or not (0) for the new egress node *t'* in flow *f*
m	The number of functions in the multi-objective function
$connection_{ij}$	The indicator of whether there is a link between nodes *i* and *j*

The problem of minimizing the dynamic membership group is formulated as follows:

(MHDB-D model)

Minimize

$$r_1^{'}.\alpha + r_2^{'}.\sum_{(i,j)\in E} Yd_{ij}^{t'f} + r_3^{'}.\sum_{(i,j)\in E} bw_f.Xd_{ij}^{t'f} + r_4^{'}.\sum_{(i,j)\in E} v_{ij} Yd_{ij}^{t'f} \quad (4.1)$$

Subject to

$$\alpha = \max\{\alpha_{ij}\}, \text{ where } \alpha_{ij} = \frac{bw_f.Xd_{ij}^{t'f}}{c_{ij}} \quad (4.2)$$

$$\sum_{(i,j)\in E} Xd_{ij}^{t'f} = 1, \quad i \in T_f \vee i = s \quad (4.3)$$

$$\sum_{(i,j)\in E} Xd_{ij}^{t'f} - \sum_{(j,i)\in E} Xd_{ji}^{t'f} = -1 \quad , j = t' \quad (4.4)$$

$$\sum_{(i,j)\in E} Xd_{ij}^{t'f} - \sum_{(j,i)\in E} Xd_{ji}^{t'f} = 0 \quad , i \neq t', i \notin T_f \quad (4.5)$$

$$bw_f . Xd_{ij}^{t'f} \leq c'_{ij} \quad ,(i,j) \in E, t' \in T'_f, f \in F \tag{4.6}$$

$$\sum_{j \in N} Yd_{ij}^{t'f} \leq \left\lceil \frac{bw_f}{\left\lfloor \sum_{j \in N} c'_{ij} \Big/ \sum_{j \in N} connection_{ij} \right\rfloor} \right\rceil \quad , \tag{4.7}$$

$i \in N, f \in F, t' \in T'_f$

where

$$Xd_{ij}^{t'f} \in \Re, 0 \leq Xd_{ij}^{t'f} \leq 1 \tag{4.8}$$

$$Yd_{ij}^{t'f} = \left\lceil Xd_{ij}^{t'f} \right\rceil = \begin{cases} 0 & , Xd_{ij}^{t'f} = 0 \\ 1 & ,0 < Xd_{ij}^{t'f} \leq 1 \end{cases} \tag{4.9}$$

$$\sum_{i=1}^{m} r'_i = 1, \quad r'_i \in \Re, \quad r'_i \geq 0, m > 0 \tag{4.10}$$

The Multi-objective function (MHDB-D model) (4.1) defines a function and generates a single aggregated metric through a combination of weighting objectives, but in this case, the transmission is minimized in order to add an egress node to the existing multicast tree with load balancing through a multipath.

In the same way as the static model (MHDB-S), in this dynamic model the main objective consists in minimizing the maximum link utilization (MLU), which is represented by *α*. The hop count (HC) $\sum_{(i,j) \in E} Yd_{ij}^{t'f}$ function, the total bandwidth consumption (BC) $\sum_{(i,j) \in E} bw_f . Xd_{ij}^{t'f}$ function over all the links and the total end-to-end propagation delay (DL) $\sum_{(i,j) \in E} v_{ij} Yd_{ij}^{t'f}$ function over all links are added.

All constraints except constraint (4.3) have the same objective as in the static model (MHDB-S). Constraint (4.3) shows that the flow taken to send it to this new egress node emerges from one node and fulfills the restriction of transmitting 100% of the multicast traffic.

Once the analytical model of the dynamic case (MHDB-D) has been presented, in the next section we present an algorithm (called MMR-D) that gives an approximate solution to the model presented.

4.3 Dynamic Multicast Multitree Routing Algorithm (MMR-D)

In this section we propose a heuristic algorithm to solve the multi-objective for the dynamic multicast routing model proposed in (4.1). This algorithm, shown in Algorithm 4.1, has as its parameters *G(N,E)*, *s*, T_f, bw_f, *r* and *t'*, which are defined in the section 4.2. Let *s'*= $T_f \cup \{s\}$ be the set of initial egress nodes plus the ingress node. Let $Xd'(s')_{ij}^{t'f}$ be the fraction of flow *f* to the new egress node *t'* assigned to link *(i,j)* from node *s'*. Let $d(s')_i^{t'f}$ denote the length of a SPT (shortest path tree) from node *s'* to the new egress nodes *t'*. Let λ_f be a path from *s'* to *t'*, associated with flow *f*, which consists of several paths $P(s')_k^{t'f}$ with $1 \leq k \leq h$, where *h* is the number of paths. This algorithm consists of one outer loop for each node s' and two internal steps: 1) obtaining graph *G'* with the distance based on hop count, bandwidth consumption and delay, and 2) finding the multicast tree.

Step 1) Obtaining modified graph *G'*. In this step, all possible paths between any ingress node *s'* and the new egress node *t'* are looked for. This step consists of two nested loops which calculate (for each path $P(s')_k^{t'f}$ with a destination at node *t'* and for each node *i*, $i \in P(s')_k^{t'f}$) the distance value $d(s')_i^{t'f}$ based on the maximum link utilization, hop count, bandwidth consumption and delay.

Step 2) Finding the multicast tree. The second step consists of finding for the new egress node *t'* the paths required to transmit the flow of information, according to graph *G'* with its distance variables, $d(s')_i^{t'f}$, and the available capacity of each path. To find out the cost of the path, the r_i values obtained with the SNOPT solver in the solution of the MHDB-D model are taken into account. Out of the paths that have a capacity greater than zero, the path with the minimum cost is selected and the maximum possible flow is applied in accordance with the maximum link utilization constraint (4.6). Finally, the best metric (Zs) is selected with its paths. The $Xd_{ij}^{t'f}$ values resulting from the algorithm allow us to calculate the values of HC, DL and BC according to the multi-objective (4.1) of the MHDB-D model.

algorithm MMR-D (G(N,E),s',Tf,bwf,r,t'):

begin
for each s'← ($T_f \cup \{s\}$ do
$Xd'(s')_{ij}^{t'f}$ ← 0
Step1: (Obtain modified graph G')
while (exists a path $P(s')_k^{t'f}$ from node s' to node t') do
    for each ($i \in P(s')_k^{t'f}$) do
        $d(s')_i^{t'f} \leftarrow \{\alpha_p, HC_p, DL_p, BC_P\}$;
    endfor
endwhile
α ← min αp
Step2: (Finding multicast tree)
while (bwf > 0 &&
($Xd'(s')_{ij}^{t'f}$ ← identify a path $P(s')_k^{t'f}$ in $d(s')_i^{t'f}$ from node s' to node t)
/, min(r1*α+r2*HC_P+r3*DL_P+r4*BC_P) ≠ 0))
    do
Zs← min(r1*α+r2*HC_P+r3*DL_P+r4*BC_P)

    bwf ← bwf – δ;
endwhile
endfor
$Xd_{ij}^{t'f}$, $P_k^{t'f}$ ← $Xd'(s')_{ij}^{t'f}$, min (Zs)
δ ← min{max_flowij, (i,j)∈ $P_k^{t'f}$ } ;
increase δ units of flow along $P_k^{t'f}$ and update G';
end algorithm;

Algorithm 4.1. Algorithm MMR-D (Dynamic Multicast Multi-Tree Routing)

4.4 Conclusions and General Limitations

In this section, we have presented a dynamic multi-objective optimization model (MHDB-D) and a heuristic algorithm (MMR-D) to develop a multicast transmission with load balancing using multiple paths. In this case, as in the static case, a label mapping solution to give an implementation in MPLS through LSPs is applied. We have employed a multi-objective load-balancing scheme to minimize the following parameters: the maximum link utilization (α), the hop count (HC), the total bandwidth consumption (BC), and the total end-to-end delay (DL). Using mathematical programming, we obtained the optimal set of $Xd_{ij}^{t'f}$ (the fraction of flow f to destination node t' assigned to link *(i,j)*) for the problem (MHDB-D) for the new egress node.

The MHDB-D model proposed in this chapter for the dynamic case and the models proposed in previous chapters for the static case have the following limitations:

1. The solutions found through the weighted sum method present problems when the space of feasible solutions is not convex. As this method works in relation to linear combinations, it is possible that many solutions cannot be found.
2. In the MHDB-S and MHDB-D models proposed in Chapters 3 and 4, it is not possible to know exactly when the load balancing is been carried out and how many subflows are being created from a multicast flow. Moreover, it is also not possible to know what is the percentage transmitted by each subflow.
3. Due to the previous limitation, the dynamic model proposed just selects the path to the new egress node from the ingress node and from the previous egress nodes. Until now, it has been impossible to consider any node that is a part of the transmission flow or subflow because it was not considered a sub-index indicates the proportion of traffic of a particular flow considering each subflow.
4. Only some objectives of those presented in other related works have been considered.

To solve limitation 1, a solution of the general model (GMM-model) by using MOEA is presented in Chapter 6.

Limitations 2, 3 and 4 are solved by adding to the GMM- model a new sub-index to represent each subflow and through which it is possible to find out what is the percentage of information transmitted through each one. With regards to limitation 4 in the GMM-model, 11 objective functions for optimization have been included.

4.5 Papers Published with this Chapter

International Conference

- [DON04a] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Model and Heuristic Algorithm for Dynamic Multicast Routing". IEEE & VDE Networks 2004. Vienna, Austria. June 2004.

- [DON04b] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Scheme for Dynamic Multicast Groups". IEEE ISCC (The 9th IEEE Symposium on Computers and Communications). Alexandria, Egypt. June 2004.

- [DON04e] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Model and Heuristic Algorithm for Multipath Routing of Static and Dynamic Multicast Group". III Workshop on MPLS Networks. Girona, Spain. March 2004.

Chapter 5 Mapping subflows to P2MP LSPs

5.1 Overview

In this chapter we focus on the specific problem of mapping subflows to point-to-multipoint (P2MP) LSPs for MPLS network implementations. The aim is to obtain an efficient solution to formulate P2MP LSPs given a set of optimum subflow values. In [SOL04] the author presents a subflow mapping solution based on a linear equation system that requires a large number of equations and variables. To solve this problem, a subflow mapping heuristic for creating multiple P2MP LSPs based on the optimum subflow values obtained with the MHDB-S model is proposed in this sub-section.

5.2 The problem of lack of labels

A general problem of supporting multicasting in MPLS networks is the lack of labels. The MPLS architecture allows aggregation in point-to-point (P2P) LSPs. Aggregation reduces the number of labels that are needed to handle a particular set of flows, and may also reduce the amount of label distribution control traffic needed [ROS01]. Adding new LSPs increases the label space and hence the lookup delay. Therefore, reducing the number of labels used is a desirable characteristic for any algorithm that maps flows to LSPs.

As pointed out in [ROS01], the label based forwarding mechanism of MPLS can also be used to route along multi-point to point (MP2P) LSPs. In [SAI00] and [BHA02], aggregation algorithms that merge P2P LSPs into a minimal number of MP2P LSPs are considered. In this case, labels assigned to different incoming links are merged into one label assigned to an outgoing link. If two P2P LSPs follow the same path from an intermediate node to the egress node, these aggregation algorithms allocate the same label to the two P2P LSPs and thus reduce the number of labels used. In [APP03], an algorithm that reduces the number of MPLS labels for $|N|$ (number of nodes) + $|E|$ (number of links) without increasing the link load is presented. For differentiated services with K traffic classes with different load constraints, their limit increases to $K(|N|+|E|)$. Their stack-depth is only one, justifying MPLS implementations with limited stack depths. To reduce the number of labels used for multicast traffic, another label aggregation algorithm is presented in [OH03]. In this case, if two P2MP

LSPs follow the same tree from an ingress node to the egress node set, the aggregation algorithm allocates the same labels to the two P2MP LSPs. Ingress nodes have a new table (called the Tree Node Table), which saves node information from the P2MP LSP. Label allocation is carried out using this table.

The label stack was introduced into the MPLS framework to allow multiple LSPs to be added to a single LSP tunnel [ROS01]. In [GUP03], a comprehensive study of label size versus stack depth trade-off for MPLS routing protocols in lines and trees is undertaken. This study shows that, in addition to LSP tunneling, label stacks can also be used to dramatically reduce the number of labels required to set up LSPs in a network. Their protocols have numerous practical applications that include implementing multicast trees, and virtual private networks using MPLS as the underlying signaling mechanism.

5.3 Subflow to LSPs mapping problem in P2MP

In this section, we detail the problem of mapping multiple P2MP LSPs based on the optimum subflow values X_{ij}^{tf} obtained with the MHDB-S model (3.1). However, this mapping is difficult using the MHDB-S model because there is no index for identifying subflows [SOL04]. Remember that X_{ij}^{tf} is the fraction of flow *f* with a destination at node *t* assigned to link *(i,j)*. As the presented algorithm applies only to one flow *f*, the index *f* will be omitted when it does not cause confusion.

To explain the problem, the MHDB-S models have been applied to the topology of Fig 5.1, with a single flow *f*, where s=N1 and T={N5, N6}. In this case, a possible subflow solution (X_{ij}^{t}) obtained is shown in Fig. 5.2. The simplest solution (Fig. 5.3.) to create LSPs based on the optimum subflow values is to send each subflow (0.4 and 0.6 fractions) to the group separately, and in this case each subflow is mapped to one P2MP LPS. In Fig. 5.3, each packet represents a 0.2 fraction of the flow. With this mapping, subflows X_{12}^{5} and X_{12}^{6} are different and the maximum link utilization constraint (5) could be violated. Moreover, the network is used inefficiently because multicast node capabilities are not considered. In section 3.2 the difference between the multicast and unicast equations in the bandwidth consumption constraint was explained.

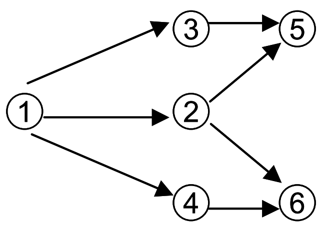


Fig 5.1. Physical network topology

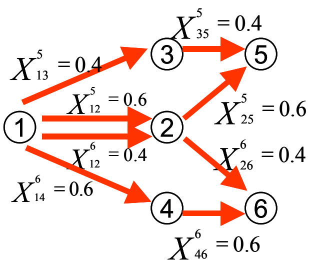


Fig 5.2. MDDB-S solution

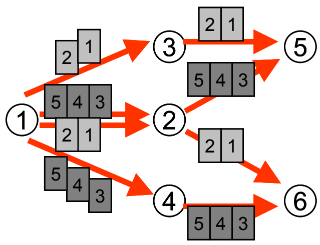


Fig 5.3. Simplest P2MP mapping

A second approach considers that one subflow is included in the other, i.e. min $(X^5_{12}, X^6_{12}) \subseteq$ max (X^5_{12}, X^6_{12}), in the example $X^5_{12} \subseteq X^6_{12}$. If both subflows X^5_{12} and X^6_{12} are sent over the link (1,2) to each member of the group separately (Fig 5.4), a part of the same flow is transmitted over the same link and the network is also used inefficiently.

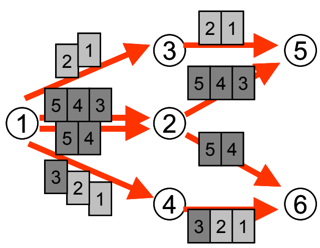


Fig 5.4. P2MP assignment: unicast transmission

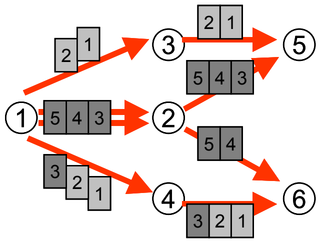


Fig 5.5. P2MP assignment: multicast transmission

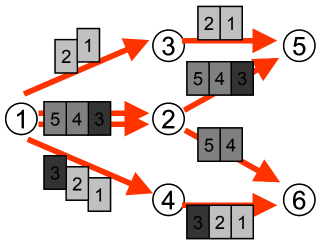


Fig 5.6. P2MP assignment: subflow mapping

Moreover, if a node has multicast capabilities, it is not necessary to transmit all the subflows over the link. In particular, if N2 has multicast capabilities, only the max (X_{12}^{5}, X_{12}^{6}) must be transmitted over link (1,2) (Fig 5.5). However, this solution presents a problem in the forwarding mechanism. Some incoming packets at node 2 must be forwarded only once (packet 3), but other packets of the same subflow must be forwarded by different output links (packets 4 and 5). To solve this, the ingress node must split this subflow into several LSPs (Fig 5.6).

5.4 Subflow assignment based on a linear equation system

In this sub-section, a linear system equation to split a subflow into several subflows is proposed. The solution is the set of desired multicast trees for the set of X_{ij}^{tf} values. For a unique flow *f* and given values X_{ij}^{t} , the δ solution of the presented system is a set of encoded *P2MP* LSPs, in which each, *P2MP*$_k$, sends a fixed fraction, c_{min}, of the whole bandwidth to all destinations in T_f. Let $\delta_{ij}^{p2mp_k}$ be a natural number, possibly 0, indicating the number of destinations that link (i, j) at *P2MP*$_k$ broadcasts.

To compute the c_{min} value, it should be taken into consideration that a low value can result in many equations with many P2MP LSPs that are the same, and in contrast, a high value can result in an unsolvable problem because some fractions of X_{ij}^{t} could not be assigned to the

P2MP LSP. On the other hand, there is a set of P2MP LSPs found, and thus δ can be regarded as a linear combination of X_{ij}^{t} values. Therefore, an optimal c_{min} value must divide all X_{ij}^{t} and the difference between them, hence $c_{\min} = \gcd^{*}\left(X_{ij}^{t}\right)$, where $\gcd^{*}(x, y)$ is the greatest common divisor operator used with real numbers between 0 and 1. In the example in Fig 4.7, the c_{min} value is 20%.

The greatest common divisor of two real numbers $\gcd^{*}(x, y)$ can be found following a modified Euclid's algorithm:

1	Let a/b = irreducible fraction (x) and c/d = irreducible fraction(y), where $x < y$
2	Let $k = lcd(a/b, c/d)$, where lcd is the least common divisor of two fractions.
3	Find $m = \mathrm{mod}(k \cdot a/b, k \cdot c/d)$
4	If $m = 0$, the result is a/b
5	$a/b = c/d$
6	$c/d = m/k$
7	Return to 2

Algorithm 5.1. Modified Euclid's Algorithm

The Table 5.1 presents the variable definitions.

Table 5.1.
Variable Definitions

Terms	Definition
δ	A solution of the presented system is a set of encoded *P2MP* LSPs
c_{min}	The whole bandwidth to all destinations in T_f
$\delta_{ij}^{p2mp_k}$	A natural number, possibly 0, indicating the number of destinations that link (i, j) at $P2MP_k$ broadcasts
$\gcd^{*}(x, y)$	It is the greatest common divisor operator used with real numbers between 0 and 1

The following three equation sets model P2MP LSPs in general.

$$\forall k, \forall j \quad \sum_{j \in N} \delta_{sj}^{p2mp_k} = |T_f| \tag{5.1}$$

$$\forall k, \forall (i,j) \in E, \forall m \quad \delta_{ij}^{p2mp_k} = \begin{Bmatrix} \sum_{m \in N} \delta_{jm}^{p2mp_k} + 1, & if \ j \in T_f \\ \sum_{m \in N} \delta_{jm}^{p2mp_k}, & otherwise \end{Bmatrix} \tag{5.2}$$

And for this problem in particular:

$$\forall (i,j) \in E \quad c_{min} \sum_k \delta_{ij}^{p2mp_k} = \sum_{t \in T_f} X_{ij}^t \tag{5.3}$$

Note that, the set $\{(i,j) | \delta_{ij}^{p2mp_k} \geqslant 1\}$ for a given k is the set of links that make up the tree $p2mp_k$.

Equation set (5.1) suggests that the number of destinations reached from a source node s is equal to $|T_f|$. This is clear because all P2MP LSPs reach all destinations. Another obvious consequence is that $|p2mp|=1/c_{min}$, in other words, the number of constructed P2MP LSPs is the inverse of the fraction sent by each P2MP LSP. For the example analyzed in Fig. 5.7, the number of P2MP LSPs to be constructed is 5.

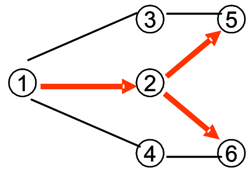
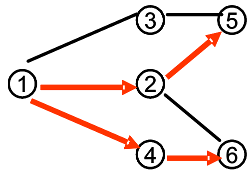
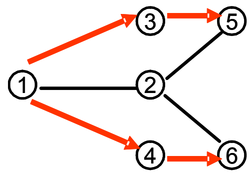
The conservation flow law seen in (3.3), (3.4) and (3.5) can be translated as the set regarded in (5.2). This means that the number of destinations that node j must forward packets to is the same amount after (i.e. i node) and before (i.e. m nodes), or one less if j is a destination.

By looking at a single link (i,j), the total bandwidth consumed in a link for a single destination should be equal to the amount consumed by all P2MP LSPs going to that destination. In general, this holds for all destinations at the same time according to (5.3).

For a given set of X_{ij}^t values, the δ solution set of the presented system is a set of encoded P2MP LSPs, which are made up of those δ values greater or equal to 1. Note that the equation system can be resolved into equal P2MP LSPs. If two P2MP LSPs, A and B, are the same (i.e. contain the same links, no more, no less), these P2MP LSPs can be merged by adding their fractions. Solving the example described in Fig 5.6, the solution (see Table 5.2) shows that 2 pairs of P2MP LSPs (1 $\wedge$ 4 and 3 $\wedge$ 5) can be merged. Therefore, we have a set of 3 P2MP LSPs in our solution; one transmitting 20% of the total bandwidth and two transmitting 40%.

Table 5.2

δ solution and the P2MP LSPs.

a. First tree. (40%) $\{\delta_{12}^{1\wedge 4}=2,\delta_{25}^{1\wedge 4}=1,\delta_{26}^{1\wedge 4}=1\}$	
b. Second tree. (20%) $\{\delta_{12}^{2}=1,\delta_{25}^{2}=1,\delta_{14}^{2}=1,\delta_{46}^{2}=1\}$	
c. Third tree. (40%) $\{\delta_{13}^{3\wedge 5}=1,\delta_{35}^{3\wedge 5}=1,\delta_{14}^{3\wedge 5}=1,,\delta_{46}^{3\wedge 5}=1\}$	
Super indexes $a \wedge b$ indicate that the values were assigned initially to a and b P2MP LSPs	

5.5 Subflow mapping heuristic

In more complex networks the linear system equation previously presented needs a large number of equations and variables. Therefore, in this sub-section, a subflow mapping heuristic to map subflow values into LSPs is proposed.

The method presented in this section returns a set of P2MP that transmits 100% of a given flow f from source node s to the destination node set T_f. In order to explain the procedure, we will start by presenting the complementary notation used. Let $\le_{s,d}$ be a total ordered relationship defined according to a pair of links, i.e. $E \times E$, in which: $(i,j) \le (k,m)$ if, and only if, $j = k$; this means that there is a node, j or k, that can forward packets coming in from node i to node j directly; $\forall (i,j) \in E, (s,x) \le (i,j)$; and $\forall (i,j) \in E, (i,j) \le (x,d)$. Let $(P \subseteq E, \le_{s,d})$ be the set that represents a single unicast path in graph G starting at node s and ending at node d, briefly summarized as $P_{s,d}$. By adding a bandwidth restriction to the set of links, we can restrict the set $P_{s,d}$ to $P_{s,d}^{t,c}$ in the following way: $P_{s,d}^{t,c} = \{(i,j) \mid (i,j) \in P_{s,d}, X_{ij}^{t} \geq c\}$ and $P_{s,d}^{t,c}$ fits in $\leq_{s,d}$. This means that $P_{s,d}^{t,c}$ is a path (found in X_{ij}^{t}) that could transmit *c* percent of the total flow from *s* to *d*. In a similar way to $P_{s,d}^{t,c}$, a tree is modeled as a set of links following a partial order relationship over the links analogous to $\leq_{s,d}$. Let *P2MP* be a set that is initially empty, in which each element of the set is a tuple of a tree, usually denoted as *M*, and a real number *c* between 0 and 1 indicating the percentage of the flow transmitted using that tree.

1 Subflow mapping (P2MP, X_{ij}^{t}, s, T_f)

2 Parameter: X_{ij}^{t} s, T_f

3 Return: p2mp

4 Begin

5 $p2mp = \phi$

6 While $\sum X_{ij}^{t} > 0$

7 $L = \left\{ (i,j) \mid \sum_{t \in T_f} Y_{ij}^{t} = \sup_{(i,j) \in E} \left(\sum_{t \in T_f} Y_{ij}^{t} \right) \right\}$ //** Computes the set of network links (L) that transmit a maximum number of subflows.

8 $(\alpha, \beta) \in \left\{ (i,j) \mid \inf_{(i,j) \in L} \left(X_{ij}^{t} \right) \right\}$ //** Choose link (α,β) in L that transmits the less fraction.

9 $c = \inf_{(\alpha,\beta)} \left(X_{\alpha\beta}^{t} \right) \in \Re, 0 < c \leq 1$

10 For each $t \in T_f$ do

11 If $P_{\beta,t}^{t,c}$ exists and $X_{\alpha,\beta}^{t} \geq c$ then

12 $M_t = P_{s,\alpha}^{t,c} \cup \{(\alpha,\beta)\} \cup P_{\beta,t}^{t,c}$

13 Else

14 $M_t = P_{s,t}^{t,c}$

15 End if

16 $\forall (i,j) \in M_t, X_{ij}^{t} = X_{ij}^{t} - c$

17 End for

18 $M = \bigcup_{t \in T_f} M_t$

19 $p2mp = p2mp \cup \{(M, c)\}$

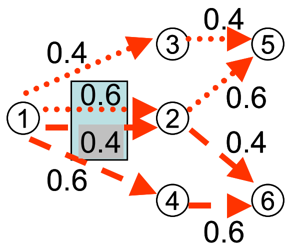
20 End While

21 End

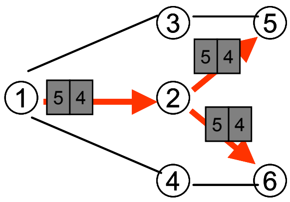
Algorithm 5.2. Subflow mapping algorithm

The path $P_{\beta,t}^{t,c}$, in line 11 of Algorithm 5.2, will still exist when t is the destination index of the selected subflow fraction c in line 9, due to the flow conservation law in (3.3), (3.4) and (3.5). Note that at least one X_{ij}^{t} is eliminated, i.e. takes the value of zero, in each iteration (6 to 20), because line 16 in Algorithm 5.2 reduces the bandwidth of link (α, β) by exactly c units when the destination is t in 9. Therefore, the complexity of the procedure is $O\left(\left|X_{ij}^{t}\right| \cdot \left|T_{f}\right|\right)$ for a single flow.

Fig 5.8a. shows the subflow solution (X_{ij}^{t}) considered to explain the subflow mapping algorithm. Initially, in lines 7 to 9 of Algorithm 5.2, set L only contains link (1,2) for destinations 5 and 6. Therefore, (α, β) will be (1,2) and $c = 0.4$ (see also Fig. 5.7a.) because it is the minimum fraction for all destinations computed in (1,2). Assuming that the cycle in line 10 begins with $t = 5$, then path $P_{2,5}^{5,0.4}$ exists. In the same way, path $P_{2,6}^{6,0.4}$ also exists. Hence, the multicast tree built in the first iteration is $M = \{(1,2),(2,5),(2,6)\}$ (see Fig 5.7b) and the values $X_{1,2}^{5}, X_{1,2}^{6}, X_{2,5}^{5}$ and $X_{2,6}^{6}$ are reduced to 0.4.



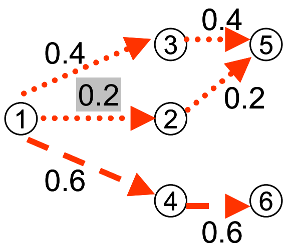
a. Initial subflow fraction values



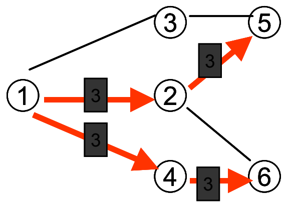
b. First multicast tree: (40%)

Fig 5.7. First iteration of the algorithm

In this case, the set of remaining fraction values X_{ij}^{t} is presented in Fig. 5.8a. The link (α, β) selected could be either (1,2) or (2,5) because both have the smallest capacity (c= 0.2). We will select (1,2) for the example. In this case, path $P_{2,5}^{5,0.2}$ exists, but $P_{2,6}^{6,0.2}$ does not. So, $M_5 = \{(1,2),(2,5)\}$ is computed with line 12 of Algorithm 5.2 and $M_6 = \{(1,4),(4,6)\}$ with line 14. The resulting tree $M = \{(1,2),(1,4),(2,5),(4,6)\}$ with subflow fraction 0.2 is shown in Fig 5.8b.



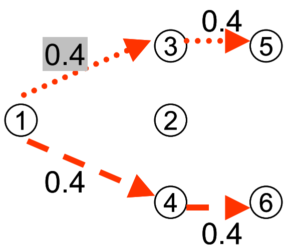
a. Subflow fraction values before second iteration.



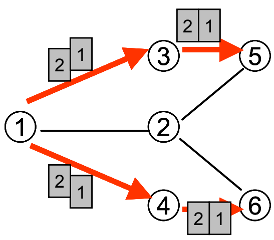
b. Second multicast tree: (20%)

Fig 5.8. Second iteration of the algorithm

Finally, the set of remaining fraction values X_{ij}^{t} is presented in Fig 5.9a. Note that all the fraction values in the last iteration are the same, in this case $c = 0.4$. The algorithm applied here results in the remaining tree, $M = \{(1,3),(1,4),(3,5),(4,5)\}$ (see Fig 5.9b).



a. Subflow fraction values before third iteration.



b. Third multicast tree: (40%)

Fig. 5.9 Final iteration of the algorithm

5.6 Conclusions and Motivations

In this chapter, we have presented a proposal for finding a solution for label assignation and the creation of LSPs in MPLS technology. With this proposal, we have given a solution for limitation 2 presented at the end of chapter 4.

However, we still have the other limitations: solving the multi-objective problem is given through lineal combination and this model does not cover some objective functions analyzed by other authors.

In the next chapter we propose a generalized model with 11 different objective functions and the solution is given in a real multi-objective context using MOEAs (Multi-Objective Evolutionary Algorithms).

5.7 Papers Published with this Chapter

International Conference

- [SOL05] F. Solano, R. Fabregat, Y. Donoso, JL. Marzo. "Asymmetric Tunnels in P2MP LSPs as a Label Space Reduction Method". IEEE ICC 2005. Seoul, Korea. May 2005.

- [SOL04a] F. Solano, R. Fabregat, Y. Donoso, JL. Marzo. "Mapping subflows to P2MP LSPs". IEEE International Workshop on IP Operations & Management (IPOM 2004). Beijing, China. October 2004.

- [SOL04] F. Solano, R. Fabregat, Y. Donoso. "Subflow assignment model of multicast flows using multiple P2MP LSPs". CLEI (Congreso Latinoamericano de Estudios en Informática) 2004. Arequipa, Perú. October 2004.

Chapter 6 Generalized Multi-Objective Multitree model

6.1 Overview

In the previous chapters we presented the proposal for solving the multi-objective problem using load balancing in multicast transmissions. In the last chapters the problem was solved using the weighted sum method. In addition, the limitations and problems of the proposals were presented:

- The solutions found using the weighted sum method present problems when the space of feasible solutions is not convex. Since this method works with linear combinations, it is possible that many solutions cannot be found.
- In the dynamic case the solution found is sub-optimal.
- Only some objectives of those presented in other related works have been considered.

To overcome these drawbacks, in this chapter we propose a *Generalized Multiobjective Multitree model* (GMM-model) in a pure multiobjective context that considers simultaneously for the first time multicast flow, multitree, and splitting. A Multiobjective Evolutionary Algorithm (MOEA) approach is proposed to solve the GMM-model.

6.2 Basic Concepts of Multi-Objective Optimization and MOEAs

6.2.1 Multi-Objective Optimization scheme

A general *MOP* includes a set of n parameters (decision variables), a set of k objective functions and a set of m restrictions. The objective and restriction functions are functions of the decision variables [COL03] [COH78]. Then the MOP can be expressed as:

$$
\begin{array}{ll}
\textit{Optimize} & \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})) \\
\textit{Subject to} & \mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), e_2(\mathbf{x}), \ldots, e_m(\mathbf{x})) \geq \mathbf{0} \\
\textit{Where } \mathbf{x} = (x_1, x_2, \ldots, x_n) \in \mathbf{X} & \\
& \mathbf{y} = (y_1, y_2, \ldots, y_k) \in \mathbf{Y}
\end{array}
\qquad (6.1)
$$

Where $\mathbf{x}$ is the decision factor and $\mathbf{y}$ the objective factor. The decision space is denoted by $\mathbf{X}$, and the objective space by $\mathbf{Y}$. Optimizing, depending on the problem, can also mean minimizing or maximizing.

The restriction set $\mathbf{e}(\mathbf{x}) \geq \mathbf{0}$ determines the feasible solution set $\mathbf{X}_f$ and its corresponding set of feasible objective vectors $\mathbf{Y}_f$.

Definition 6.1: Set of feasible solutions. The set of feasible solutions $\mathbf{X}_f$ is defined as the set of decision vectors $\mathbf{x}$ satisfying the requirements $\mathbf{e}(\mathbf{x})$:

$$\mathbf{X}_f = \{\mathbf{x} \in \mathbf{X} \mid \mathbf{e}(\mathbf{x}) \geq \mathbf{0}\} \qquad (6.2)$$

The image $\mathbf{X}_f$, that is the feasible region of the objective space is denoted by:

$$\mathbf{Y}_f = \mathbf{f}(\mathbf{X}_f) = \bigcup_{x \in X_f} \{\mathbf{y} = \mathbf{f}(\mathbf{x})\} \qquad (6.3)$$

Therefore, from these definitions we can see that each solution of the MOP consists of an n-tuple $\mathbf{x} = (x_1, x_2, \ldots, x_n)$, which leads to an objective vector $\mathbf{y} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x}))$, where each $\mathbf{x}$ must fulfill the restriction set $\mathbf{e}(\mathbf{x}) \geq \mathbf{0}$. The optimization problem consists of finding $\mathbf{x}$ with the "best value" of $\mathbf{f}(\mathbf{x})$. In general, and as we have stated before, there is no one "best value", but rather a set of solutions. In this set no one solution can be considered better than the others if all the objectives are taken into consideration at the same time. This derives from the fact that there may exist – and in general there exist - a conflict between the different objectives that make up the problem. For this reason, when dealing with MOPs a new concept of "optimal" is necessary.

In one objective optimization the set of feasible decision variables is completely ordered through an objective function f. That is, given two solutions $\mathbf{a}, \mathbf{b} \in \mathbf{X}_f$, only one of the following propositions is valid: $f(\mathbf{a}) > f(\mathbf{b})$, $f(\mathbf{a}) = f(\mathbf{b})$ o $f(\mathbf{b}) > f(\mathbf{a})$. The objective consists in finding the solution (or solutions) with the optimal values (maximum or minimum) of f. When we are dealing with various objectives, the situation changes. $\mathbf{X}_f$, in general is not totally ordered by objectives; the order given is usually partial, (that is, with the decision factors ***a*** and ***b***, $\boldsymbol{f}(\boldsymbol{a})$ cannot be considered to be better than $\boldsymbol{f}(\boldsymbol{b})$, and neither can $\boldsymbol{f}(\boldsymbol{b})$ be considered better than $\boldsymbol{f}(\boldsymbol{a})$). The following graph illustrates this concept. It shows the relationship between two functions f_1 and f_2. Function f_1 represents the cost of the safety mechanisms of a system, while f_2 represents the accident occurrence index.

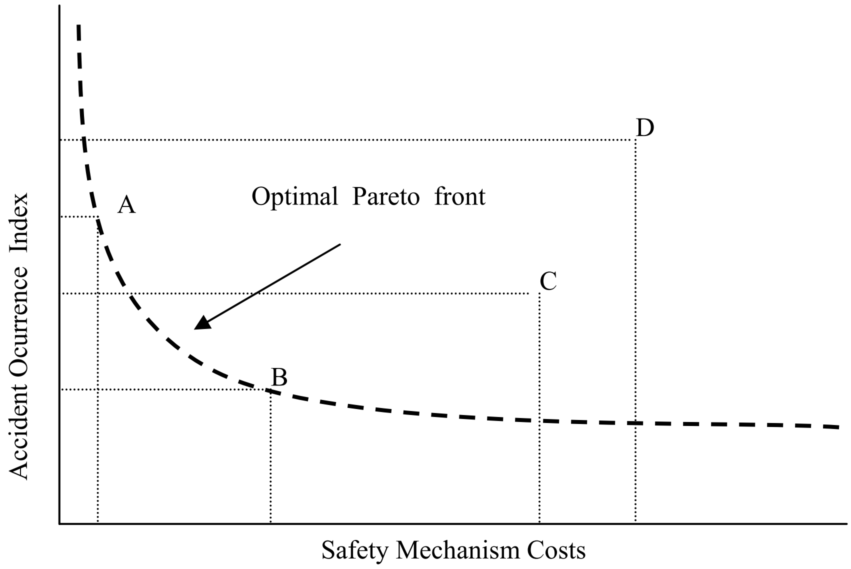


Fig 6.1 Example of Optimal Pareto in the Objective Space

The solution represented by point B is better than that of point C due to the fact that it provides better safety at a lower cost. Solution B would also be chosen in the case of performing an objective optimization. Moreover, all solutions located in the rectangle limited by the origin coordinates and solution C and over the curve are better than C. Comparing A and C we find that by decreasing costs a lot the safety level provided by A is only a bit worse, although C and A cannot really be compared because it is impossible to say that one is better than the other when considering all the objectives. If we compare A with B, we would not be able to establish that one of the two is better either, if we consider that both objectives are equally important and no subjective consideration is introduced. Nevertheless, B is clearly superior to C in both objectives. To express this situation mathematically the =, ≤ y ≥ relations must be extended. This can be performed as follows:

Definition 6.2: Given two decision vectors $\boldsymbol{u} \in \boldsymbol{X}$ and $\boldsymbol{v} \in \boldsymbol{X}$,

$$\begin{aligned}
&\boldsymbol{f}(\boldsymbol{u}) = \boldsymbol{f}(\boldsymbol{v}) \text{ if, and only if } \forall i \in \{1, 2, \dots, k\}\colon f_i(\boldsymbol{u}) = f_i(\boldsymbol{v}) \\
&\boldsymbol{f}(\boldsymbol{u}) \geq \boldsymbol{f}(\boldsymbol{v}) \text{ if and only if } \forall i \in \{1, 2, \dots, k\}\colon f_i(\boldsymbol{u}) \geq f_i(\boldsymbol{v}) \\
&\boldsymbol{f}(\boldsymbol{u}) > \boldsymbol{f}(\boldsymbol{v}) \text{ if and only if } \boldsymbol{f}(\boldsymbol{u}) \geq \boldsymbol{f}(\boldsymbol{v}) \wedge \boldsymbol{f}(\boldsymbol{u}) \neq \boldsymbol{f}(\boldsymbol{v})
\end{aligned} \tag{6.4}$$

The $\leq$ y $<$ are similarly defined.

From this notion it follows that $f(B) < f(C)$, $f(C) < f(D)$, and as a consequence $f(B) < f(D)$. Nevertheless when comparing A and C or A and B, we cannot say that one is superior to the other. For example, even though the solution represented by B is more expensive, it provides a lower accident index than that represented by A.

So it follows that two decision vectors x_1 y x_2 of a *MOP* can only fulfill one of three possible conditions: $f(x_1) > f(x_2)$, $f(x_2) > f(x_1)$ or $f(x_1) \ngtr f(x_2) \wedge f(x_2) \ngtr f(x_1)$. This situation is expressed with the following symbols and terms:

Definition 6.3: Pareto Dominance in a Maximization context. For two objective vectors $\boldsymbol{a}$ and $\boldsymbol{b}$,

$\boldsymbol{a} \succ \boldsymbol{b}$ ($\boldsymbol{a}$ dominates a $\boldsymbol{b}$)	if and only if	$\boldsymbol{a} > \boldsymbol{b}$	(6.5)
$\boldsymbol{b} \succ \boldsymbol{a}$ ($\boldsymbol{b}$ dominates an $\boldsymbol{a}$)	if and only if	$\boldsymbol{b} > \boldsymbol{a}$	
$\boldsymbol{a} \sim \boldsymbol{b}$ ($\boldsymbol{a}$ y $\boldsymbol{b}$ are not comparable)	if and only if	$\boldsymbol{a} \ngtr \boldsymbol{b} \wedge \boldsymbol{b} \ngtr \boldsymbol{a}$	

Definition 6.4: Pareto Dominance in a Minimization context. For two objective vectors a and b,

$\boldsymbol{a} \succ \boldsymbol{b}$ ($\boldsymbol{a}$ dominates a $\boldsymbol{b}$)	if and only if	$\boldsymbol{a} < \boldsymbol{b}$	(6.6)
$\boldsymbol{b} \succ \boldsymbol{a}$ ($\boldsymbol{b}$ dominates an $\boldsymbol{a}$)	if and only if	$\boldsymbol{b} < \boldsymbol{a}$	
$\boldsymbol{a} \sim \boldsymbol{b}$ ($\boldsymbol{a}$ y $\boldsymbol{b}$ are not comparable)	if and only if	$\boldsymbol{a} \nless \boldsymbol{b} \wedge \boldsymbol{b} \nless \boldsymbol{a}$	

So, from now on it will not be necessary to differentiate the type of optimization to be performed (minimization or maximization) to the point that an objective can be maximized, while another can be minimized.

It is possible to introduce the Pareto optimization criterion starting with the concept of Pareto dominance. Going back to the graph, point A is different from the others: its decision vector *a* is not dominated by another decision vector. This implies that *a* is optimal in the sense that it cannot be improved in an objective without degrading another one. These solutions are known as Pareto optimality (the term non-inferior is also used).

Definition 6.5: Pareto Optimality. Given a decision vector $x \in X_f$ and its corresponding objective vector $y = f(x) \in Y_f$, it is said that x is not dominated with respect to a set $A \subseteq X_f$ if and only if

$$\forall \ \boldsymbol{a} \in \boldsymbol{A} : (\boldsymbol{x} \succ \boldsymbol{a} \vee \boldsymbol{x} \sim \boldsymbol{a}) \tag{6.7}$$

In case x is not dominated with respect to the entire set X_f, and only in that case, it is said that x is a Pareto optimal solution ($x \in$ Xtrue, the real Pareto's optimal set). While the corresponding y is part of the real Pareto optimal front Y_{true}. This is defined as follows.

Definition 6.6: Pareto optimal set and Pareto optimal front. Given the set of feasible decision vectors X_f. X_{true} will be the set of non-dominated decision factors belonging to X_f. That is:

$$X_{true} = \{ x \in X_f \mid x \text{ is not dominated with respect to } X_f \} \tag{6.8}$$

The set X_{true} is also known as the Pareto optimal set. While the corresponding set of objective vectors $Y_{true} = f(X_{true})$ constitutes the Pareto optimal front.

Now, the MOP involves two conceptually distinct tasks: search and decision making [HOR97]. Search is related to the processes where the feasible solutions are visited in order to find the Pareto optimal solutions. Decision making refers to the ranking process of alternative solutions. A rational human decision maker determines preferences among the conflicting objectives. The methodologies in MOP can be classified into three primary groups, depending on how the search and decision making tasks are handled. Horn [HOR97] discusses the state-of-art approaches for each of these methodologies in detail [CER04]:

- Decision making before search: In this approach, the objectives are aggregated into a single objective function where the preference information of the decision maker is represented. The aggregation can be carried out in two ways: scalar combination or lexicographical ordering (ranking according to the importance) of the objectives. The aggregation of the objectives into a single objective function requires domain-specific knowledge about the ranges and the behavior of the functions. However, this kind of deep knowledge about the functions is usually not available, since the functions and/or feasible set may be too complex. However, this methodology has the advantage that the SOP strategies can be applied to the problem with the aggregated objective.

- Search before decision making: The feasible set is searched to find a set of best alternatives, without giving any information about preferences. Decision making considers only the reduced set of alternatives. For most of the real-life problems, gaining fundamental knowledge about the problem and alternative solutions can be very helpful in realizing the conflicts that are inherent in the problem. Performing the search before decision making makes this favorable circumstances possible, however the search process becomes more difficult with the exclusion of the preferences of the decision maker.

- Integrating search and decision making: This approach includes the interactive methods where the preferences of the decision maker are used during the search process. At each iteration, the result of the search is evaluated by the decision maker in order to update the preferences. The search space is then reduced and the direction of the search is restricted to some particular regions according to the preferences of the decision maker. This last methodology integrates the theory of decision making into optimization theory.

The interest of this thesis is in the first and in the second categories and comparisons between these kinds of methodologies are shown in the chapter 7. The first categories are given by the proposals presented in the chapters 3 and 4. The second category is given by the proposal presented in this chapter. In the second category when the method obtain an optimal solution set a rational human decision maker determines preferences among the conflicting objectives.

6.2.2 MOEA concepts

The term evolutionary algorithm (EA) refers to searching and optimization techniques inspired by the evolution model proposed by Charles Darwin after his exploratory trips [BAC00].

In nature individuals are characterized by chains of genetic material that are denominated by chromosomes. All information related to the individual and his or her tendencies is coded in the chromosomes. Each element making up the chromosome is called an allele. Each individual has an adaptation level to the environment, which gives him or her a greater survival capacity and possibility to generate descendants. This level of adaptation is linked to the characteristics encoded in the chromosomes. As the genetic material can be passed from parents to children as pairing occurs, the resulting children have chromosome chains resembling those of their parents and they combine characteristics of both. So, if two parents with good characteristics cross, they will probably generate equally good children or even better ones [DEB01].

In order to solve a searching or optimization problem using evolutionary algorithms and the suggested concepts first a given number of possible solutions to the problem are presented as individuals of a finite population. This process is called coding. When coding an individual, all the relevant information concerning the individual and considered to influence optimization or searching must be present. Generally, the coding of an individual or its chromosome is a chain of bits or round numbers, depending on the problem to be solved. In this chain the element located in a given position is called an allele.

Next the ability or adaptation level of each individual is determined (fitness), depending on the quality of the solution it represents. Later, the existing individuals generate new individuals through genetic operators such as selection, crossing or mutation. The selection operator chooses the parents that will be crossed. The probability that an individual is chosen as a parent and/or that it survives up to the next generation is linked to its fitness or ability; the greater the fitness, the greater the probability of surviving and having descendants, in the same way that it occurs in natural processes.

After choosing the parents, their recombination or crossing occurs in order to obtain a new generation. In this way, in every new generation there is a high probability that the new population is composed of better individuals, since the offspring inherit the good characteristics of their parents, which when combined will be better.

On the other hand, during recombination alterations (mutations) may occur in an individual's genetic information. If the alterations that occur are good, they will generate a good individual with a high fitness and the alteration will be transmitted to the next generation; if, on the contrary, the alteration is not beneficial, the altered individual will have a low fitness and very few descendants or none at all. If this is so, this alteration will disappear with it. In this way, after several generations, the population will have evolved genetically to be individuals with a high ability level. That is to say, they represent good solutions to the proposed problem.

The operators described are called searching or genetic operators. Reproduction focuses its attention on individuals with a high degree of fitness, and in this way, it takes advantage of the information at hand concerning the adaptation of the individual to the environment. Recombination and mutation somehow disturb individuals and they provide heuristics for the exploration of searching spaces. This is why the Eas are said to use the concepts of exploitation and exploration. Even though they are simplistic from the Biology point of view, these algorithms are complex enough to provide strong searching mechanisms which can adapt to a great variety of problems.

Since the mid seventies several algorithms in the area of the Eas have been introduced. They can be classified mainly as Genetic Algorithms, (GA), Evolutionary Programming, (EP), and Evolution Strategies, (ES), Classifying Systems, (CS) and Genetic Programming (GP). Even though the principle sustaining them is apparently very simple these algorithms have proven to be generally strong and powerful tools. In the next sub-section there is a short introduction to genetic algorithms.

In this thesis the use of MOEA is oriented to look for a better solution to the multi-objective case presented in this thesis, but the idea is not to look for and to analyze the different kinds and the best MOEA to be applied in this thesis.

Genetic Algorithms

Genetic algorithms are used in several areas especially for searching and optimizations. In the praxis the algorithm is implemented by choosing a coding for the possible solutions to the problem. The coding is done through chains of bits, numbers or characters that represent the chromosomes. The crossing and mutation operations are applied in a very simple way through functions of vector value manipulation. Even though many research projects on coding have been developed using chains of varying lengths and even other structures, most of the work with genetic algorithms is focused on chains of bits with a fixed length. As a matter of fact using fixed length chains and the need to codify the possible solutions are the two crucial characteristics which differentiate genetic algorithms from genetic programming, which does not have a fixed length representation and which normally does not use a coding of the problem and its solutions.

The working cycle of a GA (also known as generation cycle) is generally the following: calculate the fitness of the individuals in a population, create a new population through selection, crossing or mutation, and finally discard the old population and keep on creating using the recently created population. Each iteration in this cycle is known as a generation. It is necessary to mention that there is no theoretical reason for this to be the implementation method. As a matter of fact this punctual behavior is not seen as such in nature. Nevertheless the model remains valid and convenient.

The first generation (generation 0) of this process operates in an individual population generated according to chance. From there the genetic operators are applied to improve the population. Algorithm 6.1 presented next shows the main algorithm.

```
 GA( ) Procedure
Starting
        t = 0 /* start with an initial time */
        Starting Population P (t) /* initiate a population of individuals generated by chance
in */
        Evaluate P (t) /* evaluate the fitness of all the individuals in the initial population
*/
        Until the stopping criterion is not reached
        t = t + 1 /*  increase the time counter */
        P' = Select Parents P (t) /* select a population to generate descendants */
                Recombine P' (t) /* recombine the "genes" of the group of parents selected */
                Mutate P' (t) /* perturb the generated population in a stochastic form */
        Evaluate P' (t) /* calculate fitness of the newly created population */
                P = P, P' Survivors (t)/* Select survivors for the next generation */
        End While
End
```

Algorithm 6.1. Basic Genetic Algorithm

As it has been mentioned before the selection operator simulates the natural selection process where the strongest has the most survival ability. In the GA the survival ability of an individual is linked to the numerical value of the objective function or fitness. This operation is applied to each iteration in a population of individuals of constant size, with the aim of selecting promising individuals to generate the new population. Among the selected individuals there can be two or more identical individuals. The reason for this is that individuals with low fitness have little possibility of being elected, while the ones with good fitness are more frequently selected.

The selection operator can be applied in several ways. Sometimes the selection is developed through tournaments where a group of individuals is chosen by chance and the winner of the tournament is the individual with the best fitness. The number of individuals chosen for the competence is previously fixed and remains constant in the traditional implementation. This form of selection is called tournament selection and it most adequately reflects the natural process of selection. Another form is called roulette selection. It is easily understood by imagining a roulette wheel where the number of parts into which the roulette wheel is divided is the same as the number of individuals in the population. The size of each part is proportional to the fitness of each individual. It is expected that when the roulette wheel spins several times a larger number of individuals with high fitness will be obtained. Although there may be other ways to make the selection the two methods mentioned above are the most common in the published implementations. Next we present algorithm 6.2, which only contains the description of the roulette selection operator because the tournament one is very simple and implementing it is very direct.

```
Selection Procedure( )
Start
        Random Value = random number between 0 and 1
        Partial Addition = 0
        Rand = Random Value * ∑_{i=1}^{N} f_i
        i = 1
        While (i ≤ N y Rand ≥ SumaParcial )
        Partial Addition = Partial Addition + fi
        i = i + 1
        End While
        Return i
End
```

Algorithm 6.2. Genetic Algorithm. Selection Operator

Once individuals have been selected the crossing operator is applied to each pair. Crossing can be done in different ways and we will only discuss one of them here, one point crossing. The one cutting point is selected randomly and it is applied to the pair of chromosomes or coding of the selected individuals. The most significant characters at the point of cutting will be maintained in their relative positions in the new pair of chromosomes and the rest are exchanged from the progeny chromosomes to the two new chromosomes obtained. For example consider the two chromosomes A1 and A2 from a pair of individuals in which the crossing operator will be applied to obtain two new chromosomes. Say the individuals are:

A1 = 0 1 1 | 0 1

A2 = 1 1 0 | 0 0

Where the "|" symbol indicates the cut point. The characters on the left of the cutting point are the most significant, so they remain in their corresponding positions in the new chromosomes, while the characters to the right of the cutting point are crossed with the parental chromosomes as shown in the example:

A'1 = 0 1 1 | 0 0

A'2 = 1 1 0 | 0 1

In this way A'1 and A'2 are the chromosomes resulting from applying the crossing operator to the pair of parental chromosomes

In Algorithm 6.3 the crossing operator is described. In the already mentioned pseudo-code l represents the length (in bits) of the chromosome; Jcross is the cutting point; the vectors Descendant 1 and Descendant 2 are equivalent to the binary representation of a pair of descendants; and Parent 1 and Parent 2 are the parents. The given crossing probability indicates the probability that parents will cross and the probability of generating clones. In this way not all selected individuals are destroyed by crossing.

```
CrossOver Procedure(Parent1, Parent2)
Start
        Rand = random number between 0 and 1
        If Rand is Lower than the given crossing probability
                JCross = random number between 1 and l
        If not
                JCross = l
        End Si
        From i = 1 to i = JCross, i = i + 1
        Offspring1[i] = Parent1[i]
        Offspring2[i] = Parent2[i]
        End From
        From i = Jcross + 1 to i = l, i = i + 1
        Offspring 1[i] = Parent2[i]
        Offspring2[i] = Parent1[i]
        End From
End
```

Algorithm 6.3. Genetic Algorithm. Crossover Operator

The crossing operator represents a form of local search in the regions next to the searching space surrounding the parents. On the other hand the mutation process is basically an random search. A specific position in the chromosome of the mutating individual is selected in an random way, then, the content value in that position is changed.
In nature the probability of a mutation occurring is small in normal living conditions. In the GA we try to represent the same probability with a very low occurrence value for the mutation operator.

Evolutionary algorithms and multi-objective optimization

The Eas are interesting given the fact that at first glance they seem especially apt to deal with the difficulties presented by MOPs. The reason for this is that they can return an entire set of solutions after a simple run and they do not have any other of the limitations of traditional techniques. In addition, some researchers have suggested that the Eas would behave better than other blind searching techniques. This statement still needs a clear demonstration and it should be considered in the light of the "no free lunch" theorems related with optimization, which states that if a method behaves "better" than other methods in one set of problems, it will perform "worse" in another set. The reality is that nowadays there are very few valid alternatives concerning other possible solution methods. In fact, the most recent publications on MOP resolutions using Eas (which are compiled in [DEB01]), seem to consider this fact and they have opened the way to a whole new field of research: evolutionary algorithms applied to multi-objective optimization (MOEA): Multiobjective Optimization Evolutionary Algorithm).

6.3 GMM-model

The proposed GMM-model considers a network represented as a graph $G(N, E)$, with N denoting the set of nodes and E the set of links. The cardinality of a set is denoted as $|.|$, thus $|N|$ represents the cardinality of N.
The set of flows is denoted as F. Each flow $f \in F$ can be split into K_f subflows that after normalization can be denoted as f_k; $k = 1, \dots |K_f|$. In this case, f_k indicates the fraction of $f \in F$ it transports, i.e.

$$\sum_{k=1}^{|K_f|} f_k = 1 \qquad (6.9)$$

For each flow $f \in F$ we have a source $s_f \in N$ and a set of destination or egress nodes $T_f \subset N$. Let t be an egress node, i.e. $t \in T_f$.

Let $X_{ij}^{f_k t}$ denote the fraction of subflow f_k to egress node t assigned to link $(i,j) \in E$, i.e. $0 \leq X_{ij}^{f_k t} \leq 1$. In this way, the n components of decision vector X are given by all $X_{ij}^{f_k t}$. Note that $X_{ij}^{f_k t}$ uses five indexes: i, j, f, k and t for the first time, unlike previous publications that only used a smaller subset of indexes because they did not deal with the same general problem [KIM02] [LEE02] [ROY02] [SEO02] [WAN01]. In particular, the novel introduction of a subflow index k gives an easy way to identify subflows and define LSPs in a MPLS implementation.

Let c_{ij} be the capacity (in bps) of each link $(i,j) \in E$. Let b_f be the traffic request (measured in bps) of flow $f \in F$, traveling from source s_f to T_f. Let d_{ij} be the delay (in ms) of each link $(i,j) \in E$. The binary variables $Y_{ij}^{f_k t}$ represent whether a link (i,j) is being used (value 1) or not (value 0) for transporting subflow f_k to destination node t, i.e.

$$Y_{ij}^{f_k t} = \left\lceil X_{ij}^{f_k t} \right\rceil = \begin{cases} 0, & \mathit{if}\ X_{ij}^{f_k t} = 0 \\ 1, & \mathit{otherwise} \end{cases} \tag{6.10}$$

where $\lceil . \rceil$ denotes the ceiling function and consequently, $\lfloor . \rfloor$ denotes the floor function. Finally, let $connection_{ij}$ be an indicator of whether there is a link between nodes i and j.

Given the above notation and the multiobjective context already presented in equation (6.1), the proposed GMM-model considers the following objective functions:

Maximal link utilization

$$\phi_1 = \max\{\alpha_{ij}\} \quad \text{where} \quad \alpha_{ij} = \frac{1}{c_{ij}} \sum_{f=1}^{|F|} \sum_{k=1}^{|K_f|} b_f \max_{t \in T_f} \{X_{ij}^{f_k t}\} \tag{6.11}$$

Hop count, in several different flavors such as:

Total hop count

$$\phi_2 = \sum_{(i,j) \in E} \sum_{f \in F} \sum_{k \in K_f} \sum_{t \in T_f} Y_{ij}^{f_k t} \tag{6.12}$$

Hop count average

$$\phi_3 = \frac{\sum_{(i,j)\in E} \sum_{f\in F} \sum_{k\in K_f} \sum_{t\in T_f} Y_{ij}^{f_k t}}{\sum_{f\in F} \sum_{k=1}^{|K_f|} |T_f|} \tag{6.13}$$

Maximal hop count, which is useful for QoS assurance

$$\phi_4 = \max_{\substack{\forall f\in F,\\ \forall k\in K_f,\\ \forall t\in T_f}} \left\{ \sum_{(i,j)\in E} Y_{ij}^{f_k t} \right\} \tag{6.14}$$

Maximal hop count variation for a flow, which is useful for jitter and queue size calculations

$$\phi_5 = \max_{\substack{\forall f\in F\\ \forall t\in T_f}} \left\{ \mathrm{H}_{ft} \right\} \tag{6.15}$$

where $\mathrm{H}_{ft} = \max_{k\in K_f} \left\{ \sum_{(i,j)\in E} Y_{ij}^{f_k t} \right\} - \min_{k\in K_f} \left\{ \sum_{(i,j)\in E} Y_{ij}^{f_k t} \right\}$

Delay

Total delay

$$\phi_6 = \sum_{(i,j)\in E} \sum_{f\in F} \sum_{k\in K_f} \sum_{t\in T_f} d_{ij} \cdot Y_{ij}^{f_k t} \tag{6.16}$$

Average delay

$$\phi_7 = \frac{\sum_{(i,j)\in E} \sum_{f\in F} \sum_{k\in K_f} \sum_{t\in T_f} d_{ij} \cdot X_{ij}^{f_k t}}{\sum_{f\in F} \sum_{k=1}^{|K_f|} |T_f|} \tag{6.17}$$

Maximal delay, which is useful for QoS assurance

$$\phi_8 = \max_{\substack{\forall f\in F\\ \forall k\in K_f\\ \forall t\in T_f}} \left\{ \sum_{(i,j)\in E} d_{ij} \cdot Y_{ij}^{f_k t} \right\} \tag{6.18}$$

Maximal delay variation for a flow, which is useful for jitter and queue size calculations

$$\phi_9 = \max_{\substack{\forall f\in F\\ \forall t\in T_f}} \left\{ \Delta_{ft} \right\} \tag{6.19}$$

where $\Delta_{ft} = \max_{k \in K_f} \left\{ \sum_{(i,j) \in E} d_{ij} \cdot Y_{ij}^{f_k t} \right\} - \min_{k \in K_f} \left\{ \sum_{(i,j) \in E} d_{ij} \cdot Y_{ij}^{f_k t} \right\}$

Total bandwidth consumption

$$\phi_{10} = \sum_{(i,j) \in E} \sum_{f \in F} \sum_{k \in K_f} b_f \cdot \max_{t \in T_f} \left\{ X_{ij}^{f_k t} \right\} \tag{6.20}$$

Number of subflows that can give an idea of the maximum number of LSPs for an MPLS implementation

$$\phi_{11} = \sum_{f=1}^{|F|} |K_f| \tag{6.21}$$

As stated in equation (6.1), a MOP formulation usually considers *m* constraints (C), such as the following.

Flow conservation constraints:

for every source node $\forall s_f \in N$ and $\forall f \in F, \forall t \in T_f$,

$$\sum_{j=1}^{|N|} \sum_{k=1}^{|K_f|} X_{s_f j}^{f_k t} = 1 \tag{6.22}$$

for every destination $\forall t \in T_f$ and $\forall f \in F$

$$\sum_{i=1}^{|N|} \sum_{k=1}^{|K_f|} X_{it}^{f_k t} = 1 \tag{6.23}$$

for every other node i_f, $\forall f \in F$, $\forall k \in K_f$, $\forall t \in T_f, \forall i_f \in N$, $i_f \neq s_f, i_f \neq t \in T_f$

$$\sum_{j=1}^{|N|} X_{i_f j}^{f_k t} - \sum_{i=1}^{|N|} X_{i i_f}^{f_k t} = 0, \tag{6.24}$$

A subflow uniformity constraint, to ensure that a subflow f_k always transports the same information:

$$\forall (i,j) \in E, \forall t \in T_f,$$

$$X_{ij}^{f_k t} = \begin{cases} X^{f_k} = \max\limits_{\substack{t \in T_f, \\ (i,j) \in E}} \left\{ X_{ij}^{f_k t} \right\}, & \text{if } Y_{ij}^{f_k t} = 1 \\ 0, & \text{if } Y_{ij}^{f_k t} = 0 \end{cases} \tag{6.25}$$

without this restriction,

$X_{ij}^{f_k t}>0$ may differ from $X_{i'j'}^{f_k t'}>0$ and therefore, the same subflow f_k may not transport the same data to different destinations t and t'. As a consequence of this new constraint, mapping subflows to LSPs is easy.

Link capacity constraint, $\forall i \in N, \forall j \in N,$:

$$\sum_{f=1}^{|F|}\sum_{k=1}^{|K_f|} b_f \cdot \max_{t\in T_f}\left\{X_{ij}^{f_k t}\right\} \le c_{ij} \tag{6.26}$$

Constraint on the maximum number of subflows:

$\forall f \in F, \forall t \in T_f, \forall i \in N,$

a. constant maximum number:

$$\sum_{k=1}^{|K_f|}\sum_{j\in N} Y_{ij}^{f_k t} \le N_{\max} \tag{6.27}$$

b. or alternatively, depending on required bandwidth b_f:

$$\sum_{k=1}^{|K_f|}\sum_{j\in N} Y_{ij}^{f_k t} \le \left\lceil \frac{b_f}{\left\lfloor \sum_{j\in N} c_{ij} \Big/ \sum_{j\in N} connection_{ij} \right\rfloor} \right\rceil \tag{6.28}$$

In summary, the proposed GMM-model follows the general mathematical framework of any MOP, given by equation (6.1). In this context, this model considers 11 objective functions given by equations (6.11) to (6.21), and 7 constraints given by (6.22) to (6.28). Clearly, it is not difficult to increment the number of objectives or constraints of the proposed model if new ones appear in the literature or they are useful for a given situation. In fact, *Packet Loss* was not considered in this first proposal, but including it would be very easy.

At this point, it is important to point out that the mathematical solution of the proposed GMM-model is a complete set **X*** of Pareto optimal solutions $x^*\in$**X***, i.e. any solution x' outside the Pareto set ($x'\notin$**X***) is outperformed by at least one solution x^* of the Pareto set ($\exists x^* \succ x'$); therefore, x' cannot outperform x^* even if not all the objective functions are considered. Consequently, under the same set of constraints, any previous model or algorithm, that only considers a subset of the proposed objective functions, either as a SOP or MOP, can find one or more solutions calculated with the GMM-model or dominated by solutions $x^* \in \mathbf{X}^*$ of this model.

In conclusion, by using the GMM-model it is possible to calculate the whole set of optimal Pareto solutions. This includes any solution that has been found previously using most of the already published alternatives that consider any subset of the proposed objective functions. Now it is clear why we call this model *generalized.*

6.3.1 Computational Solution applying MOEA to the Static case

To solve the GMM-model, a Multiobjective Evolutionary Algorithm (MOEA) approach has been selected because of its well recognized advantages when solving MOPs in general and TE load balancing in particular [CRI04], [CRI04a], [CRI04b], [FAB04], [KOY04], [ROY02], [CUI03] and [ROY04]. A MOEA, as a genetic algorithm, has been inspired by the mechanics of natural evolution (based on the survival of the fittest) [VAN99].

At the beginning, an initial population of P_{max} feasible solutions (known as individuals) is created as a starting point for the search. In the next stages (or generations), a performance metric, known as fitness, is calculated for each individual. In general, a modern MOEA calculates fitness considering the dominance properties of a solution with respect to a population. Based on this fitness, a selection mechanism chooses good solutions (known as parents) for generating a new population of candidate solutions, using genetic operators like crossover and mutation [GOL89]. The process continues iteratively, replacing old populations with new ones, saving the best found solutions (a process known as elitism), until a stop condition is reached.

In this chapter, an algorithm based on the *Strength Pareto Evolutionary Algorithm* (SPEA) [ZIT99] is proposed. It has an evolutionary population P and an external set P_{nd} with the best Pareto solutions found. Starting with a random population P, the individuals of P evolve to optimal solutions that are included in P_{nd}. Old dominated solutions of P_{nd} are pruned each time a new solution from P enters P_{nd} and dominates old ones.

Encoding

Encoding is the process of mapping a decision variable x into a chromosome (the computational representation of a candidate solution). This is one of the most important steps towards solving a TE load balancing problem using evolutionary algorithms. Fortunately, it has been studied a lot in current literature. However, it should be mentioned that to our best knowledge, this paper is the first one that proposes an encoding process, as shown in Figure 6.2, that allows representing several flows (unicast and/or multicast) with as many splitting subflows as necessary to optimize a given set of objective functions to be represented.

In this proposal, each chromosome consists of $|F|$ flows. Each flow f, denoted as *(Flow f)*, contains $|K_f|$ subflows that have resulted from splitting and which flow in several subflows (multitree, for load balancing). Inside a flow f, every subflow *(f,k)*, denoted as *(Subflow f,k)*, uses two fields. The first one represents a tree *(Tree f,k)* which is used to send information about flow f to the set of destinations T_f, while the second field represents the fraction f_k of the total information of flow f being transmitted. Clearly, equation (6.12) should be satisfied to assure that all information in flow f arrives to destinations T_f.

Moreover, every tree *(Tree f,k)* consists of $|T_f|$ different paths *(Path f,k,t)*, one for each destination $t \in T_f$. Finally, each path *(Path f,k,t)* consists of a set of nodes N_l between the source node s_f and destination $t \in T_f$ (including s_f and t). For optimality reasons, it is possible to define *(Path f,k,t)* as not valid if it repeats any of the nodes, because in this case it contains a loop that may be easily removed from the given path to make it feasible. Moreover, in the above representation, a node may receive the same (redundant) information by different paths of the same subflow; therefore, a correction algorithm has been implemented to choose only one of these redundant path segments, making sure that all subflows satisfy the optimality criteria.

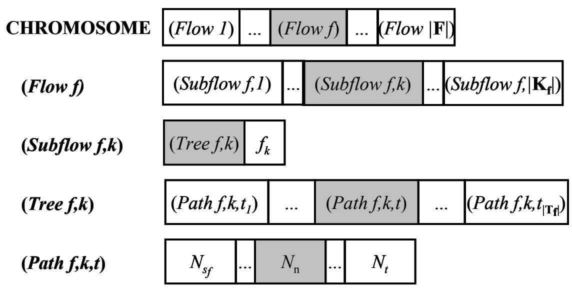
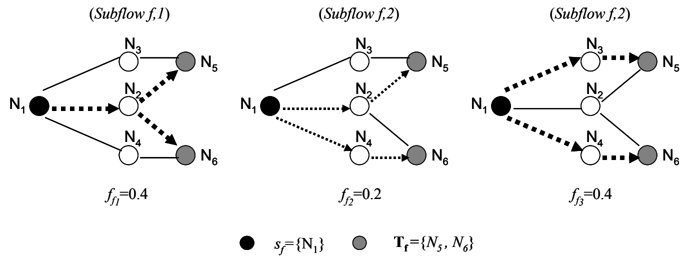


Fig 6.2. Chromosome representation

One interesting advantage of the proposed encoding is that a valid chromosome satisfies restrictions (6.22) to (6.28) completely, making verifying them unnecessary.
An example of the proposed encoding process is presented in Figure 6.3 which shows a chromosome representation for a very simple network topology of six nodes ($|N|=6$) with a flow f between a source node $s_f=\{N_1\}$ and two destination nodes: $T_f=\{N_5, N_6\}$.

In this example, flow f is split into three subflows:

- A first subflow (k=1) transmits 40% of the total flow information through *(Tree f,1)* which consists of two different paths, (*Path f,1,5*)={N_1, N_2, N_5} and (*Path f,1,6*)={N_1, N_2, N_6}.
- A second subflow (k=2) transmits 20% of the total flow through (*Tree f,2)* using two different paths, (*Path f,2,5*)={N_1, N_2, N_5} and (*Path f,2,6*)={N_1, N_4, N_6}.
- Finally, a third subflow (k=3) transmits the final 40% of the total flow through (*Tree f,3)*, which consist of two different paths, (*Path f,3,5*)={N_1, N_3, N_5} and (*Path f,3,6*)={N_1, N_4, N_6}.




```
[
 (Flow f =                    /* Generic flow f split into 3 subflows
  (Subflow f,1 = (                              /* first subflow
    Tree f,1=(
      Path f,1,5 = (N1,N2,N5),              /* path to destination N5
      Path f,1,6 = (N1,N2,N6) ) ),         /* path to destination N6
    fraction f1 = 0.4 )       /* subflow 1 transmits 40% of flow
  (Subflow f,2 = (                            /* second subflow
    Tree f,2=(
      Path f,2,5 = (N1,N2,N5),              /* path to destination N5
      Path f,2,6 = (N1,N4,N6) ) ),         /* path to destination N6
    fraction f2 = 0.2 )        /* subflow 2 transmits 20% of flow
  (Subflow f,3 = (                              /* third subflow
    Tree f,3)=(
      Path f,3,5 = (N1,N3,N5),              /* path to destination N5
      Path f,3,6 = (N1,N4,N6) ) ),         /* path to destination N6
    fraction f3 = 0.4 )        /* subflow 3 transmits 40% of flow
 )
]
```

Fig 6.3. Network topology and chromosome representation for a *flow f* with three subflows

Initial Population

To generate an initial population P of valid chromosomes we have considered each chromosome one at a time, building each (*Flow f*) of that chromosome separately. For each (*Flow f*) we first generated a large enough set of different valid paths from source s_f to each destination $t \in T_f$ (see line 1 of Algorithm 6.4). Then, an initial $|K_f|$ is chosen as a reasonable random number that satisfies constraints on the maximum number of subflows given by equations (6.27) and (6.28). To build each of the $|K_f|$ subflows, we randomly generated a tree with its roots in s_f and its leaves in T_f by randomly selecting a path one at a time for each destination, from the previously generated set of paths.

Trees are made up of a path-set, which can contain redundant segments; i.e. two paths belonging to a tree with different destinations can meet in more than one node causing redundant subflow information to be transmitted between the pair of nodes. To correct this anomaly, a *repair redundant segments process* has been defined: the shortest path of this tree can be taken as a *pattern* and then for each of the remaining paths in the tree, its shortest segment starting at the latest node (branching node) in the *pattern* must be found. The resulting segment will be a *pattern* segment starting at its source at the branching node joint with the old segment starting at the branching node at the destination. Later, an information fraction of $f_k=1/|K_f|$ is set.

In this initialization procedure (see lines 2-3 of Algorithm 6.4), chromosomes are randomly generated one at a time. A built chromosome is valid (and accepted as part of the initial population) if it also satisfies link capacity constraint (6.29); otherwise, it is rejected and another chromosome is generated until the initial population P has the desired size P_{max}.

Selection

Good chromosomes of an evolutionary population are selected for reproduction with probabilities that are proportional to their fitness. Therefore, a *fitness function* describes the "quality" of a solution (or individual). An individual with good performance (like the ones in P_{nd}) has a high fitness level, while an individual with bad performance has a low fitness. In this proposal, fitness is computed for each individual, using the well-known SPEA procedure. In this case, the fitness for every member of P_{nd} is a function of the number of chromosomes it dominates inside the evolutionary set P, while a lower fitness for every member of P is calculated according to the chromosomes in P_{nd} that dominate the individual considered. A *roulette selection operator* is applied to the union set of P_{nd} and P each time a chromosome needs to be selected.

Crossover

We propose two different crossover operators: *flow crossover* and *tree crossover*. With the *flow crossover* operator (line 11 in Algorithm 6.4), $|F|$ different chromosomes are randomly selected to generate one offspring chromosome that is built using one different flow from each father chromosome, as shown in Figure 6.4. A father may be chosen more than once, contributing with several flows to an offspring chromosome.

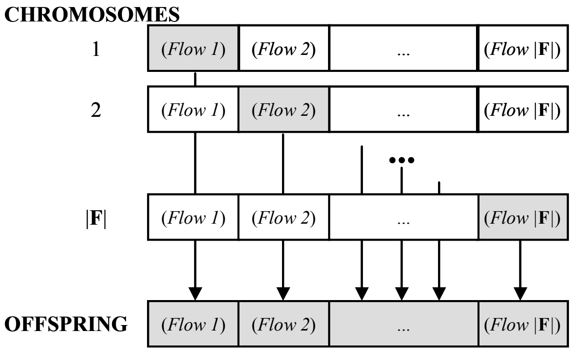


Fig 6.4. Flow crossover operator

The *tree crossover operator* is based on a two-point crossover operator, which is applied to each selected pair of parent chromosomes (line 11 in Algorithm 6.4). In this case, the crossover is carried out by making tree exchanges between two equivalent flows of a pair of randomly selected parent-chromosomes, as shown in Figure 6.5.

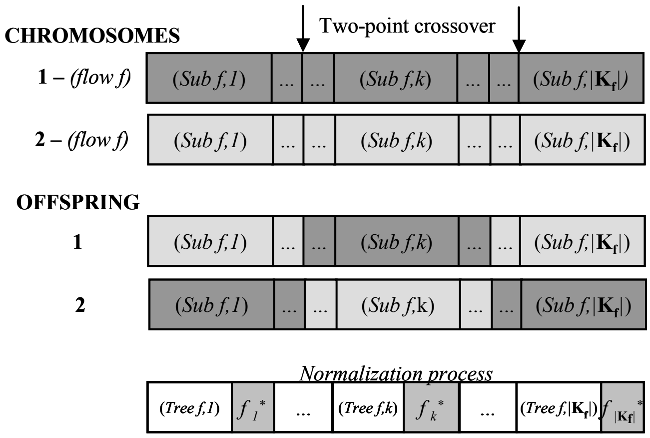


Fig 6.5. Tree crossover operator

Tree crossover without normalization of the information fraction f_k usually generates infeasible chromosomes because equation (6.9) is not satisfied. Therefore, *a normalization process*:

$$f_k{}^* = f_k \Big/ \sum_{k=1}^{|K_f|} f_k \ , \qquad (6.29)$$

is used as the last step of a *tree crossover operator*.

Mutation

A mutation operator is usually used to ensure that an optimal solution can be found with a probability greater than zero. This operator can improve an evolutionary algorithm's performance, given its ability to continue the search for global optimal (or near optimal) solutions even after local optimal solutions have been found, which stops the algorithm from being easily trapped in local sub-optimal solutions. Each time that an offspring chromosome is generated, a (generally low) mutation probability p_m is used to decide if the mutation operator should be applied to this chromosome (line 13 in Algorithm 6.4).

To apply a mutation operator, we first randomly choose a *(Flow f)* for the new offspring, in order to later select (also randomly) a *(Subflow f,k)* in which the mutation will actually apply; therefore, what we implement is a subflow mutation operator. For this work, we propose a subflow mutation operator with two phases: *segment mutation* and *subflow fraction mutation*.

For the *segment mutation phase*, a (*Path f,k,t*) of (*Tree f,k*) is chosen randomly. At this point, a node N_j of (*Path f,k,t*) is selected as a *Mutation Point.* The s*egment mutation phase* consists in finding a *new segment* to connect the selected *Mutation Point* to destination t (see Figure 6.6), followed by the *repair redundant segment process* that has already been explained (see *B - Initial Population*), to achieve better chromosome quality.

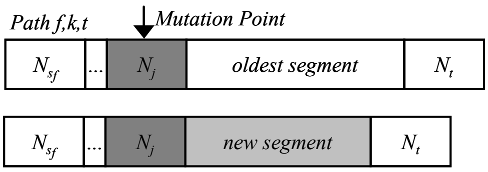


Fig 6.6. Segment mutation

Finally, the *subflow fraction mutation phase* is applied to *(Subflow f,k)* by incrementing (or decrementing) flow fraction f_k in δ (see Figure 6.7), followed by the *normalization process* that has already been explained, to assure that equation (6.12) is satisfied.

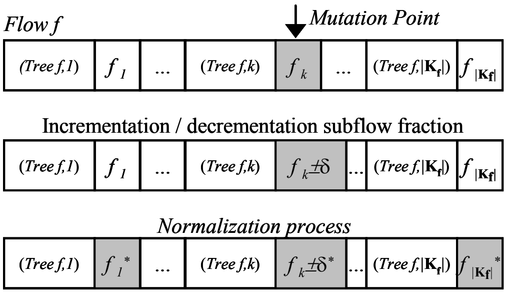


Fig 6.7. Subflow fraction mutation followed by the normalization process

This MOEA is summarized in Algorithm 6.4.

```
Obtain initial set of valid paths
Generate the initial population P of size Pmax
Normalize fractions and remove redundant segments for every chromosome in P
Initialize set Pnd as empty
DO WHILE A FINISHING CRITERIOM IS NOT SATISFIED {
  Add non-dominated solutions of P into Pnd
  Remove dominated solutions in Pnd
  Calculate fitness of individuals in P and Pnd
  REPEAT Pmax times {
     Generate new chromosome set C using
       Tree crossover with Selection (in P ∪ Pnd) and normalization process
       Flow crossover with Selection (in P ∪ Pnd)
     With probability pm mutate set C and the normalization process and remove
     redundant segments
     Add to P the valid chromosomes in C not yet included in P
  END REPEAT
END WHILE
```

Algorithm 6.4. Proposed Algorithm

6.3.2 Limitations

In the previous section the GMM model and a computational solution using MOEAs in the static case were presented. However, now it is necessary to give a solution to the dynamic case with the same aims as the static case, because in multicast transmission it is possible that the egress nodes enter and leave the tree in transmission time.

6.4 Dynamic GMM-model

Inspired by the GMM-model, the present work proposes a *Dynamic Multiobjective Multitree model* (Dynamic-GMM-model) in a pure multiobjective context that considers simultaneously for the first time, multicast flow, multitree, splitting and dynamic nodes. The Dynamic-GMM-model is an extension of the GMM-model.

The proposed Dynamic-GMM-model considers a network represented as a graph $G(\mathbf{N}, \mathbf{E})$, with $\mathbf{N}$ denoting the set of nodes and $\mathbf{E}$ the set of links. Remember that the cardinality of a set is denoted as $|.|$, thus $|\mathbf{N}|$ represents the cardinality of $\mathbf{N}$. The set of flows is denoted as $\mathbf{F}$. Each flow $f \in \mathbf{F}$ can be split into $\mathbf{K_f}$ subflows that after normalization can be denoted as f_k; $k = 1, \ldots |\mathbf{K_f}|$. In this case, f_k indicates the fraction of $f \in \mathbf{F}$ it transports, i.e.

$$\sum_{k=1}^{|K_f|} f_k = 1 \tag{6.30}$$

For each flow $f \in \mathbf{F}$ we have a source $s_f \in \mathbf{N}$ and a set of destination or egress nodes $\mathbf{T_f} \subset \mathbf{N}$. Let t be an egress node, i.e. $t \in \mathbf{T_f}$.

Let $X_{ij}^{f_k t}$ denote the fraction of subflow f_k to egress node t assigned to link $(i,j) \in \mathbf{E}$, i.e. $0 \le X_{ij}^{f_k t} \le 1$. In this way, the n components of decision vector $\mathbf{x}$ are given by all $X_{ij}^{f_k t}$. Note that $X_{ij}^{f_k t}$ uses five indexes: i, j, f, k and t.

We also need to include the following notation. Let c_{ij} be the capacity (in bps) of each link $(i,j) \in \mathbf{E}$. Let b_f be the traffic request (measured in bps) of flow $f \in \mathbf{F}$, traveling from source s_f to $\mathbf{T_f}$. Let d_{ij} be the delay (in ms) of each link $(i,j) \in \mathbf{E}$. The binary variables $Y_{ij}^{f_k t}$ represent whether a link (i,j) is being used (value 1) or not (value 0) for transporting subflow f_k to destination node t, i.e.

$$Y_{ij}^{f_k t} = \left\lceil X_{ij}^{f_k t} \right\rceil = \begin{cases} 0, & \mathit{if}\ X_{ij}^{f_k t} = 0 \\ 1, & \mathit{otherwise} \end{cases} \tag{6.31}$$

where $\lceil . \rceil$ denotes the ceiling function and consequently, $\lfloor . \rfloor$ denotes the floor function. Finally, let $connection_{ij}$ be an indicator of whether there is a link between nodes i and j.

Given the above notation and the multiobjective context already presented by equation (6.1), the proposed Dynamic-GMM-model considers the following objective functions:

Maximal link utilization

$$\phi_1 = \max\{\alpha_{ij}\} \tag{6.32}$$

where $\alpha_{ij} = \dfrac{1}{c_{ij}} \sum_{k=1}^{|K_f|} b_f X_{ij}^{f_k t}$

Hop count, in several different flavors such as:

Total hop count

$$\phi_2 = \sum_{(i,j)\in E} \sum_{k\in K_f} Y_{ij}^{f_k t} \tag{6.33}$$

Hop count average

$$\phi_3 = \frac{\sum_{(i,j)\in E} \sum_{k\in K_f} Y_{ij}^{f_k t}}{\sum_{k=1}^{|K_f|} |T_f|} \tag{6.34}$$

Maximal hop count, which is useful for QoS assurance

$$\phi_4 = \max_{\substack{\forall f\in F,\\ \forall k\in K_f,\\ \forall t\in T_f}} \left\{ \sum_{(i,j)\in E} Y_{ij}^{f_k t} \right\} \tag{6.35}$$

Maximal hop count variation for a flow, which is useful for jitter and queue size calculations

$$\phi_5 = \max_{\substack{\forall f\in F\\ \forall t\in T_f}} \{ \mathrm{H}_{ft} \} \tag{6.36}$$

where $\mathrm{H}_{ft} = \max_{k\in K_f} \left\{ \sum_{(i,j)\in E} Y_{ij}^{f_k t} \right\} - \min_{k\in K_f} \left\{ \sum_{(i,j)\in E} Y_{ij}^{f_k t} \right\}$

Delay

Total delay

$$\phi_6 = \sum_{(i,j)\in E} \sum_{k\in K_f} d_{ij} \cdot Y_{ij}^{f_k t} \tag{6.37}$$

Average delay

$$\phi_7 = \frac{\sum_{(i,j)\in E} \sum_{k\in K_f} d_{ij} \cdot Y_{ij}^{f_k t}}{\sum_{k=1}^{|K_f|} |T_f|} \tag{6.38}$$

Maximal delay, which is useful for QoS assurance

$$\phi_8 = \max_{\substack{\forall f\in F \\ \forall k\in K_f \\ \forall t\in T_f}} \left\{ \sum_{(i,j)\in E} d_{ij} \cdot Y_{ij}^{f_k t} \right\} \tag{6.39}$$

Maximal delay variation for a flow, which is useful for jitter and queue size calculations

$$\phi_9 = \max_{\substack{\forall f\in F \\ \forall t\in T_f}} \{\Delta_{ft}\} \tag{6.40}$$

where $\Delta_{ft} = \max_{k\in K_f} \left\{ \sum_{(i,j)\in E} d_{ij} \cdot Y_{ij}^{f_k t} \right\} - \min_{k\in K_f} \left\{ \sum_{(i,j)\in E} d_{ij} \cdot Y_{ij}^{f_k t} \right\}$

Total bandwidth consumption

$$\phi_{10} = \sum_{(i,j)\in E} \sum_{k\in K_f} b_f \cdot X_{ij}^{f_k t} \tag{6.41}$$

Number of subflows that can give an idea of the maximum number of LSPs for a MPLS implementation

$$\phi_{11} = \sum_{f=1}^{|F|} |K_f| \tag{6.42}$$

As stated in equation (7.1), a MOP formulation usually considers m constraints (C), such as the following:

Flow conservation constraints:

for every source node $\forall s_f \in N$ and $\forall f \in F, \forall t \in T_f$,

$$\sum_{j=1}^{|N|} \sum_{k=1}^{|K_f|} X_{s_f j}^{f_k t} = 1 \tag{6.43}$$

for every destination $\forall t \in T_f$ and $\forall f \in F$

$$\sum_{i=1}^{|N|}\sum_{k=1}^{|K_f|} X_{it}^{f_k t} = 1 \tag{6.44}$$

for every other node i_f, $\forall f \in F$, $\forall k \in K_f$, $\forall t \in T_f$,$\forall i_f \in N$, $i_f \neq s_f , i_f \neq t \in T_f$

$$\sum_{j=1}^{|N|} X_{i_f j}^{f_k t} - \sum_{i=1}^{|N|} X_{i i_f}^{f_k t} = 0, \tag{6.45}$$

A subflow uniformity constraint, to ensure that a subflow f_k always transports the same information:

$\forall (i, j) \in E, \forall t \in T_f$,

$$X_{ij}^{f_k t} = \begin{cases} X^{f_k} = \max\limits_{\substack{t \in T_f,\\ (i,j) \in E}} \left\{ X_{ij}^{f_k t} \right\}, & if\, Y_{ij}^{f_k t} = 1 \\ 0, & if\, Y_{ij}^{f_k t} = 0 \end{cases} \tag{6.46}$$

without this restriction,

$X_{ij}^{f_k t}$ >0 may differ from $X_{i'j'}^{f_k t'}$ >0 and therefore, the same subflow f_k may not transport the same data to different destinations t and t'. As a consequence of this new constraint, mapping of subflows to LSPs is easy.

Link capacity constraint, $\forall i \in N, \forall j \in N,$:

$$\sum_{f=1}^{|F|}\sum_{k=1}^{|K_f|} b_f \cdot \max_{t \in T_f} \left\{ X_{ij}^{f_k t} \right\} \leq c_{ij} \tag{6.47}$$

Constraint on the maximum number of subflows:

$\forall f \in F, \forall t \in T_f, \forall i \in N,$

a. constant maximum number:

$$\sum_{k=1}^{|K_f|} \sum_{j \in N} Y_{ij}^{f_k t} \leq N_{\max} \tag{6.48}$$

b. or alternatively, depending on required bandwidth b_f:

$$\sum_{k=1}^{|K_f|} \sum_{j \in N} Y_{ij}^{f_k t} \leq \left\lceil \frac{b_f}{\left\lceil \sum_{j \in N} c_{ij} \Big/ \sum_{j \in N} connection_{ij} \right\rceil} \right\rceil \qquad (6.49)$$

In summary, the proposed Dynamic-GMM-model follows the general mathematical framework of any MOP. This work considers 11 objective functions given by equations (6.32) to (6.42), and 7 constraints given by (6.43) to (6.49). Clearly, in a similar way to the GMM-model, it is not difficult to increment the number of objectives or constraints of the proposed model if new ones appear in the literature or they are useful for a given situation.

6.5 Dynamic GMM Model Resolution Using a Probabilistic Breadth First Search Algorithm

In this section we propose the D-GMM algorithm to solve the Dynamic-GMM-model using a probabilistic Breadth First search algorithm. The D-GMM algorithm, shown in Algorithm 6.5, has as its parameters *G(N,E), S_f, bw_f and t',* which have been defined previously. Let S_f be the subflows set for flow *f*. Let *t'* be the new node for flow *f*.

This algorithm consists in three steps:

- The first is to find paths. This step consists of obtaining different paths from each node of the subflow with a destination at the new egress node that will enter the multicast flow. This step has three internal cycles because all subflows must be considered, all the paths along which this subflow is transmitted and finally each one of the nodes of this path. For each one of these nodes a probabilistic Breadth_First search algorithm is applied to the new node. This search algorithm is probabilistic because every path has a probability greater than zero of being found. In this case, the search is not exhaustive.

- The second is to combine Paths. In this step, each subflow is selected from the paths found in the previous step. With all the paths selected for each of the functions, a solution is constructed that extends the P2MP LSPs and therefore 100% of the flow is transmitted to the new egress node. In this same procedure the different objective functions for each one are calculated from the solutions found.

- The next step is to make a Non-dominated selection. This step consists of determining which solutions, out of all the solutions obtained in step 2, are not dominated. These non-dominated solutions are the set of optimal solutions.

```
D-GMM algorithm (G(N,E),Sf,bwf,t')
begin
P ← Φ
 for each subflow s ← Sf do
   while (exists a path p ∈ Sf) do
     for each node i in path p  i ∈ p do
        p' ← Breadth_First_Search(i,t)
       P ← p'U P
    endfor
   endwhile
 endfor
 combine_paths();
 no_dominated_selection();
end algorithm

function combine_paths()
begin
  while (exists a path p ∈ Sf) do
    for each subflow s ← Sf do
        T ← one path p by each subflow s
        Solutions(T) ← by each p calculate objective functions values
    endfor
  endwhile
end function

function no_dominated_selection()
begin
Optimal_Solutions ← Φ
  for each combined solution t ∈ T do
     while (t' ∈ T-{t} and t is non-dominated) do
         for each objective function do
              calculate dominance between t and t'
         endfor
     endwhile
     if (t is non-dominated for every t') then
         Optimal_Solutions ← Optimal_Solutions U t
   endif
   endfor
end function
```

Algorithm 6.5. D-GMM algorithm

The complexity of the algorithm is given by the first step. The Breadth First Search probabilistic function has the next complexity O(nlogn). Every loop has an *n* as the maximum number of iterations. Finally, the complexity of the algorithm is $O(n^3 logn)$.

6.6 Conclusions and Motivations

Given that one of the main contributions of this chapter is to formalize the GMM-model as a general model, it is interesting to compare it to another recently published model, like the MHDB-model. When comparing them, we can mention the following advantages of the GMM-model over the previous MHDB-model:

- The MHDB-model recognizes the multiobjective nature of the load balancing problem considering four objective functions (ϕ_1, ϕ_2, ϕ_6, ϕ_{10}), but it only solves a SOP using a weighted sum cost function, therefore only finding one solution for the whole Pareto set.

- The MHDB-model simultaneously considers a weighted sum of objectives that are highly correlated, like ϕ_2 and ϕ_6, which seems inefficient in a SOP context. On the other hand, the GMM-model can also consider correlated objective functions, but only to discriminate similar solutions in a multi-objective context.

- The weighted sum method proposed in the MHDB-model and several other papers (see Table II), is not good enough for finding all the solutions of a Pareto set in multiobjective non-convex problems, as stated in [VAN99].

- Given that the GMM-model clearly identifies each subflow (and even each subpath), it is very easy mapping subflows to LSPs for MPLS implementations. However, this mapping is difficult using the MHDB-model because there is no index for identifying subflows [SOL04].
- Given that each subflow is identified clearly, it is easier and more efficient to extend this generalized model to the dynamical case given that a node that wants to be included in a flow only needs to find the "closer" nodes from which subflows can be obtained.

In this chapter we have proposed a *Generalized Multiobjective Multitree model* (GMM-model) that is able to consider any type of flow (unicast and multicast traffic), any number of flows (unipath / multipath - unitree / multitree), considering (or not) splitting (or subflows) in a more general multiobjective context.

If eventually a single-objective context is preferred, techniques like weighted sum may be used to combine objective functions in a unique cost function that can be combined with restrictions on some objectives, such as an upper bound on the delay, hop count, etc. However, any optimal single objective solution, like the ones proposed in several previous papers, could be a solution of the GMM-model or be dominated by one of its Pareto solutions when all analyzed objective functions are simultaneously considered.

To solve the proposed model, a Multi-Objective Evolutionary Algorithm (MOEA) inspired by the Strength Pareto Evolutionary Algorithm (SPEA) has been implemented, proposing a new encoding process to represent multitree-multicast solutions using splitting.

Finally, we have proposed a dynamic-GMM-model to give a solution to multicast transmission when the egress nodes can enter and leave in transmission time. A computational solution has been presented using a Probabilistic Breadth First Search Algorithm. In this case, this proposal is better than the proposal presented in chapter 4, because in this new proposal we resolve the dynamic case using every node of the transmission trees, and the computational solution is given in a real multi-objective context.

6.7 Paper Published with this Chapter

International Conference

- [DON05a] Y. Donoso, R. Fabregat, F. Solano, JL. Marzo, B. Baran. “Generalized Multiobjective Multitree model for Dynamic Multicast Groups”. IEEE ICC. Seoul. Korea. May 2005.

- [FAB04] R. Fabregat, Y. Donoso, F. Solano, JL. Marzo. “Multitree Routing for Multicast Flows: A Genetic Algorithm Approach”. Setè Congrés Català d’Intel·ligència Artificial (CCIA’2004), Universitat Autònoma de Barcelona, Barcelona (Spain). October 2004.

Research Reports

- [BAR04] B. Barán, R. Fabregat, Y. Donoso, F. Solano, JL. Marzo. "Generalized Multiobjective Multitree model". Proceedings of IIiA Research Report, ref. IIiA 04-08-RR, Institut d'Informàtica i Aplicacions, University of Girona. September 2004.

Chapter 7 Analysis and Simulation Results

7.1 Overview

This chapter shows the experiments that have been carried out in order to evaluate the different models and algorithms proposed (MHDB-S, MMR-S, MHDB-D, MMR-D, GMM-model and Dynamic GMM-model). These experiments are focussed on testing the different models and the algorithms in them, as well as taking into account other proposals. First the different models (mono-objective models and the MHDB-S model) in the static case are tested, and then the MHDB-S model is compared with the MMR-S algorithms. The results obtained with the MMR-S algorithm and the MMR-D algorithm are compared. Finally, the results obtained of the GMM-model through the MOEA solution and the Dynamic GMM-model through BFS Probabilistic are compared.

The experiments were developed following the methodologies proposed by [MON04] and [HER03]. These proposals define 7 steps to develop the research evaluation process: identifying and stating the problem; selecting the factors, levels and ranks; selecting response variables; choosing the experimental design; developing the experiment; a data statistical analysis, and finally conclusions and recommendations.

7.2 Introduction

To analyse the performance in extended networks four network scenarios were considered: NSF (National Science Foundation), SPRINT Network, UUNET Network and several random networks. These topologies were mainly selected because they have been used in many previous works.

The NSF network considered has 14 nodes (Fig. 7.1). The costs on the links represent the delay and all links have 1.5 Mbps of bandwidth capacity.

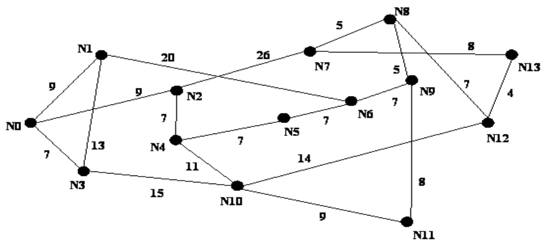


Fig. 7.1. NSF Network

The SPRINT network has 15 nodes (Fig. 7.2). The costs on the links represent the delay but these values don't appear in the figure. Links have 45Mbps, 155 Mbps, 622Mbps and 2.5Gbps of bandwidth capacity.

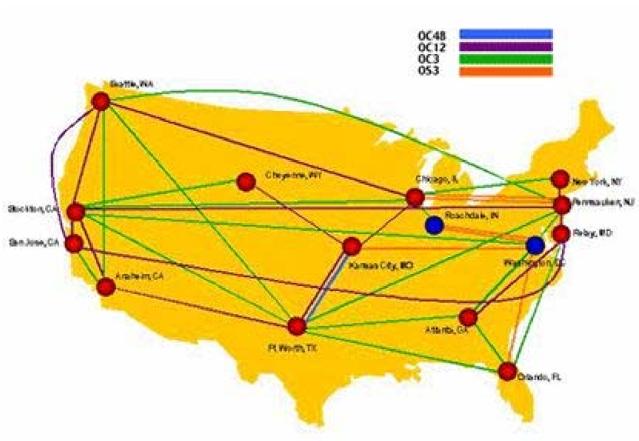


Fig. 7.2. SPRINT Network

The UUNET network has 15 main nodes and 30 total nodes (Fig. 7.3). The costs on the links represent the delay but these values don't appear in the figure. In this case the links also have 45Mbps, 155 Mbps, 622Mbps and 2.5Gbps of bandwidth capacity.

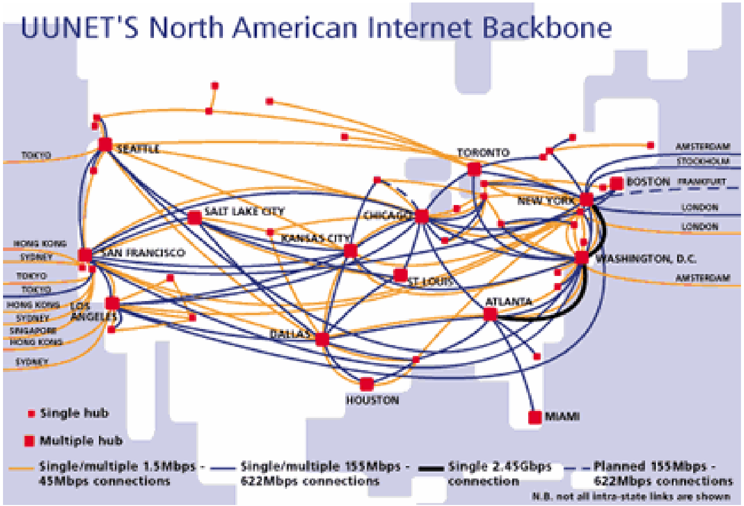


Fig. 7.3. UUNET Network

Random topologies have been used following the scheme proposed by [MED01]. These topologies are considered to prove the validity of the MMR-S and MMR-D algorithms proposed in this thesis in larger topologies than those presented in the three preceding scenarios. Using this generator several types of net topologies can be obtained by following the traditional scheme of Internet interconnectivity. It is also possible to obtain other types of topologies using this generator.

In short, the parameter values used in the tests were the following:

- Transmission rates ranging from 8% of the link capacity to 200%. Exactly 8%, 16%, 33%, 66%, 100%, 133%, 166% and 200%. In this case the flow demand increases but the number of flows doesn't.

- The selection of the ingress node was randomly generated.

- The selection of the egress nodes was randomly generated.

- The amount of egress nodes compared with the number of nodes in the network topology for each flow changed from 20% to 95%. We did not work with 100% because the ingress node or multicast flow transmitter was not the ingress node or egress node at the same time.

- Each simulation was developed with one ingress node, which was randomly selected. The number of egress nodes started at three nodes and in the case of

NSF reached up to 13 egress nodes, with values corresponding to 95% of the total number of topology nodes. For SPRINT and UUNET the same procedure was developed.

- The order in which the egress nodes entered the process was determined randomly.

When developing the tests, an important factor was choosing the sample size. As the NSF topology has 14 nodes and UUNET and SPRINT each have 15 nodes we then chose the sample size that would indicate the number of tests to be developed. Given the test variables (egress nodes randomly selected and transmitted flow percentage selected in an increasing way) the number of tests developed was the following:

- For the tests developed with the GAMS tool: For each kind of flow (eight different types in total) and for the maximum number of egress nodes (13 for NSF and 14 for UUNET and SPRINT) 104 tests were developed with NSF and 112 for UUNET and SPRINT. The total number of tests associated with each scenario type was 328.

- For tests used to compare the solutions given by GAMS and the heuristic MMR-S and MMR-D the same number of tests were developed: 104 tests for NSF and 112 for UUNET and SPRINT. The total number of tests developed with the MMR-S heuristic and with the MMR-D was the same: 328 each. This gives a total of 656 tests with the heuristics.

- Tests were also developed with larger topologies in order to test the heuristic and meta-heuristic solutions proposed. Topologies of 50, 100 and 200 nodes were randomly generated. In this case tests were developed for each flow, with egress nodes corresponding to 25%, 50%, 75% and 95% of the total number of flows. Therefore 32 experiments were carried out for each topology. That is, a total of 96 experiments.

- For the tests developed with the MOEA solution, the same number of tests was carried out in order to compare them with the former results. That is, 328 tests were developed with MOEA and 328 tests with the probabilistic BFS algorithm to solve the dynamic case. This means that in order to solve the GMM model and the Dynamic-GGM model 656 tests were developed.

- This indicates that adding up GAMS, heuristics and MOEA and probabilistic BFS 1736 tests were developed in order to establish the corresponding comparisons.

7.3 MHDB-S model versus simplified models: static case

In this section, the MHDB-S model proposed in chapter 3, which considers four objective functions, is compared with "traditional" multicast routing algorithms such as DVMRP, PIM-DM and BGMP, which are mono-objective. In particular, the MHDB-S model is compared with simple models which only consider one objective function: maximum link utilization in the MLU-model (section 7.3.1), delay in the DL-model (section 7.3.2), bandwidth consumption in the BC-model (section 7.3.3) and hop count in the HC-model (section 7.3.4). These models are proposed in several works: MLU-model in [FOR02], [LEE02] and [ABR02]; DL-model in [VUT00] and [CHE01]; BC-model in [CHO03] and one cost function which encompasses many objectives in [LEU98], [INA99], [LI99] and [SUN99].

In section 7.3.5. model MHDB-S is compared with other models, which have more than one objective and have been presented in other research works. Firstly model MHDB-S is compared with model DL-BC, considered in [RAO98] and [BAN01], in which only delay and bandwidth consumption are minimized. Later, model MHDB-S is compared with model HC-BC considered in [SON03], in which only hop count and bandwidth consumption are minimized. Finally model MHDB-S is compared with model MLU-HC-DL considered in [ABO98], in which maximum link utilization, hop count and bandwidth are minimized.

To analyse the analytical model GAMS was used as an optimisation problem solving tool. The testing consisted in using GAMS to numerically solve the MHDB-S model with the four objective functions as well as simplified models with only one objective function, which have been considered in other works ([FOR02], [LEE02], [ABR02] [VUT00], [CHE01], [CHO03], [LEU98], [INA99], [LI99] and [SUN99]), and also with the models that consider combinations of two or three objective functions ([RAO98], [BAN01], [SON03] and [ABO98]).

In the following sections the results for NSF are presented. For the SPRINT and UUNET topologies the same behaviour was observed.

7.3.1 Comparative Analysis of MHDB-S versus MLU

In this section, the MHDB-S and MLU models are compared. Remember that in the MLU model only the maximum link utilization objective is minimized.

Figures 7.4a, 7.4b, 7.4c and 7.4d respectively show the behaviour of MLU, HC, DL and BC when the information flow is 10% of the total link capacity. It can be seen that Model MHDB-S behaves similarly to Model MLU in all cases.

In this section the relevant value is the analysis of the MLU function due to the fact that model MHDB-S is being compared with model MLU, in which only this function is minimized. The maximum difference observed is 10% of the optimal value found by function MLU. The figures with the values corresponding to variables HC, DL, and BC are shown in order to be able to analyse the differences found between the MHBD-S model and the model in which these three functions are not minimized.

In the next figure every point represents the average value obtained to run with 8 different flow values, 3 different topologies and a varying number of egress nodes.

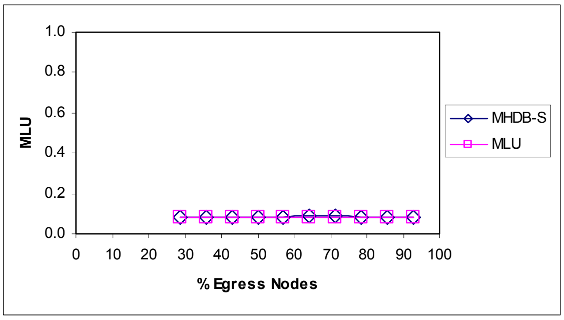


Fig. 7.4a. MLU of MHDB-S and MLU by 15%

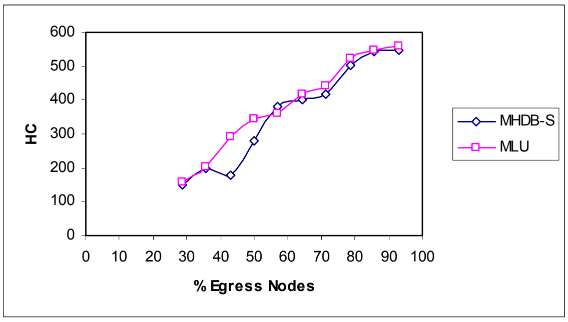


Fig. 7.4b. HC of MHDB-S and MLU by 15%

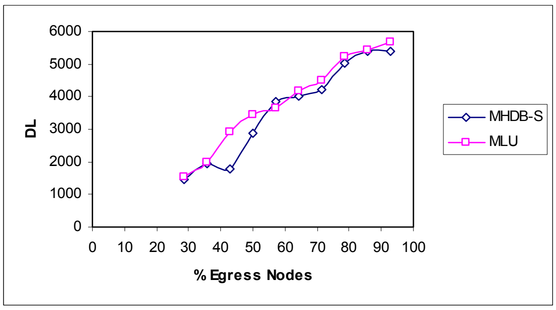


Fig. 7.4c. DL of MHDB-S and MLU by 15%

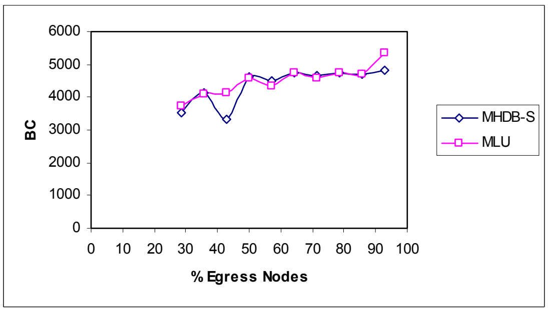


Fig. 7.4d. BC of MHDB-S and MLU by 15%

Figures 7.5 show the behaviour when the information flow is 33% of the total link capacity.

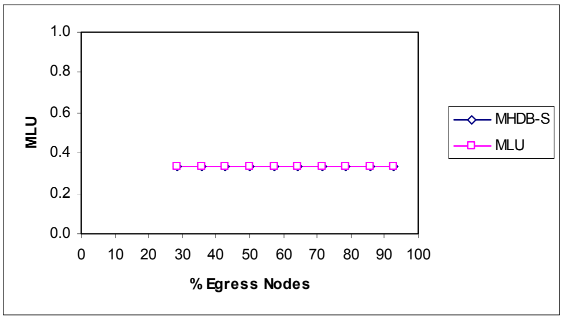


Fig. 7.5a. MLU of MHDB-S and MLU by 30%

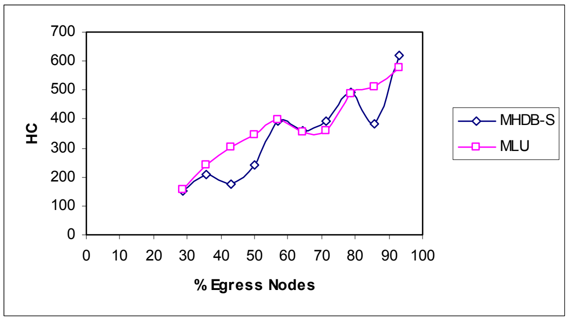


Fig. 7.5b. HC of MHDB-S and MLU by 30%

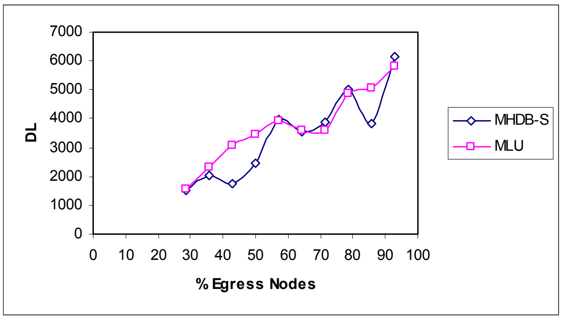


Fig. 7.5c. DL of MHDB-S and MLU by 30%

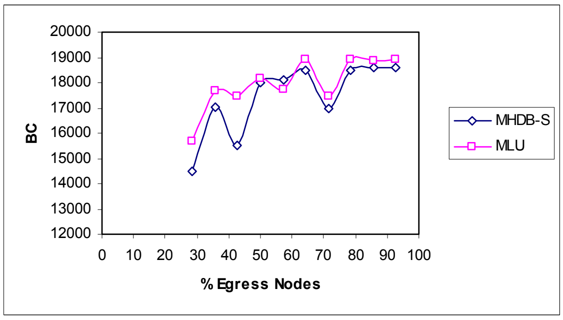


Fig. 7.5d. BC of MHDB-S and MLU by 30%

Figures 7.6 show the behaviour when the information flow is 100% of the total link capacity.

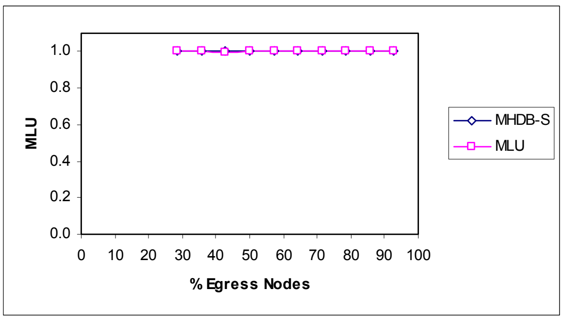


Fig. 7.6a. MLU of MHDB-S and MLU by 100%

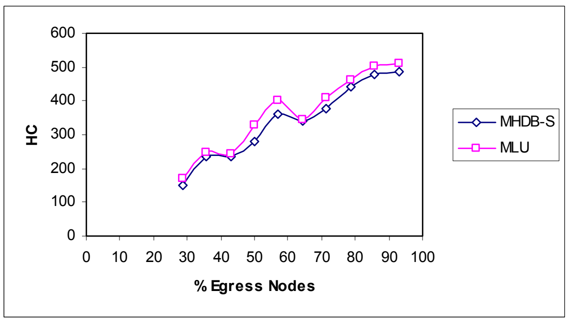


Fig. 7.6b. HC of MHDB-S and MLU by 100%

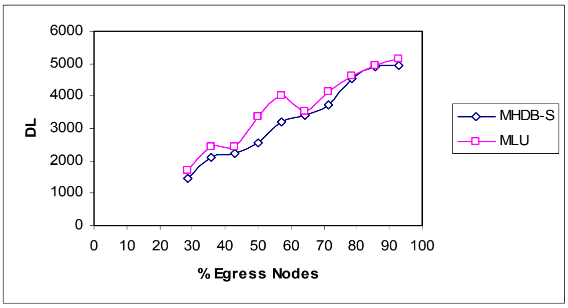


Fig. 7.6c. DL of MHDB-S and MLU by 100%

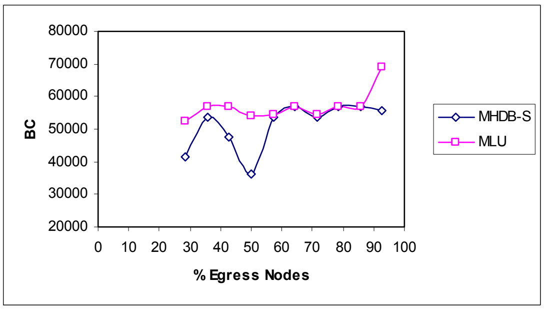


Fig. 7.6d. BC of MHDB-S and MLU by 100%

Now if model MHDB-S is compared with the MLU minimization and the behavior of the other functions is analysed (HC, DL and BC), see figures b, c, and d of 7.4, 7.5 and 7.6, we can observe that in most cases model MHDB-S behaves in a better way and the tendendy is very similar between the MHDB-S model and the others kind of functions (HC, DL and BC).

To compare the incidence of increases or decreases in a proposal with respect to the other proposal, different normalized variables were calculated. For example, to compare the MHDB-S model with the MLU model, the normalized variable is given by $MLU_{MHDB\text{-}S\ vs\ MLU} = \dfrac{MLU_{MHDB\text{-}S} - MLU_{MLU}}{MLU_{MLU}}$.

When the "normalized value" < 0, the MHDB-S model is better than the MLU model; when the "normalized value" > 0 the MHDB-S is worse; and finally, when the "normalized value"= 0 the same behaviour is shown by both models.

Figure 7.7 shows the normalized MLU values for different flow fractions and the number of egress nodes. In this case it can be observed that on average the MHDB-S model behaves in the same way as the MLU model where only the function maximum link utilization (MLU) is minimized. In the worst case a 12% difference was found. This kind of behavior is due to the main function in the MHDB-S model is the MLU and this model try to optimize mainly this function.

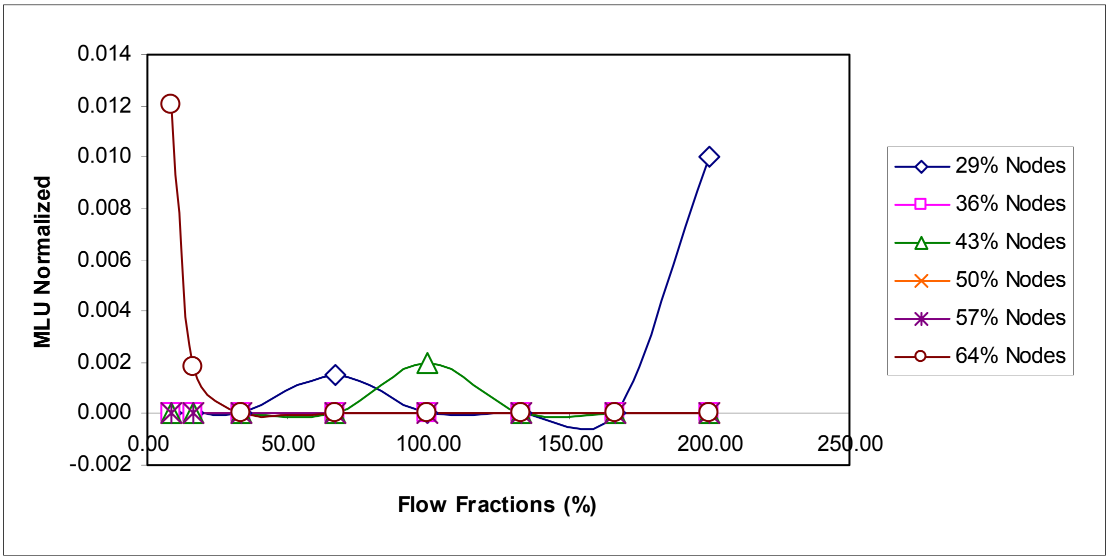


Fig 7.7. Normalized MLU versus Flow Fractions (%) with different percentages of egress nodes

When analysing all the normalized values for model MHDB-S and model MLU, it was observed that model MHDB-S compared to:

- Function HC, model MHDB-S behaves better in most cases (up to 40% better) and in the worst case 17% worse.
- Function DL, model MHDB-S behaves better in most cases (up to 45% better) and in the worst case 20% worse.
- Function BC, model MHDB-S behaves better in most cases (up to 35% better) and in the worst case 10% worse.

In all cases, the MHDB-S model show a better behavior in comparison with the function HC, DL, BC is due to the MHDB-S model optimize these function too. Only some few points are betther than the MHDB-S model due to the multi-objective optimization process.

7.3.2 Comparative Analysis of MHDB-S versus HC

In this section, the MHDB-S model and the HC model are compared. Remember that in the HC model only hop count objective is minimized.

In Figure 7.8 model MHDB-S is compared with model HC. Figures 7.8a, 7.8b, 7.8c and 7.8d show the behaviour of HC when the information flow is 15%, 33%, 66% and 100% respectively, of the total link capacity.

It can be observed that model MHDB-S behaves with values near to those shown by function HC. The maximum difference observed is 35% of the optimal value found by the HC function.

It can be observed in the figures that the HC values obtained for the MHDB-S model are greater than those obtained for the HC model, which implies that we cannot minimize the variable too much and therefore the MHDB-S model has worse behaviour with respect to hop count (HC).

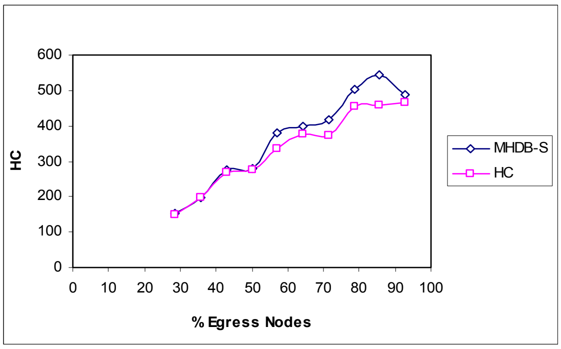


Fig. 7.8a. HC of MHDB-S and HC by 15%

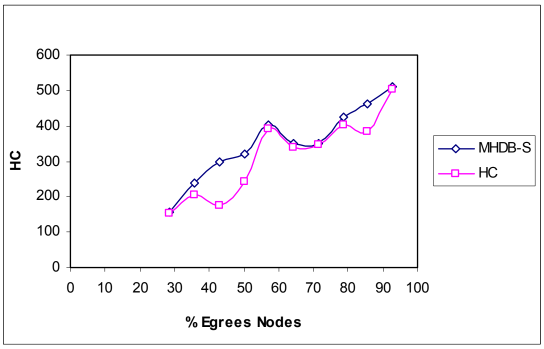


Fig. 7.8b. HC of MHDB-S and HC by 33%

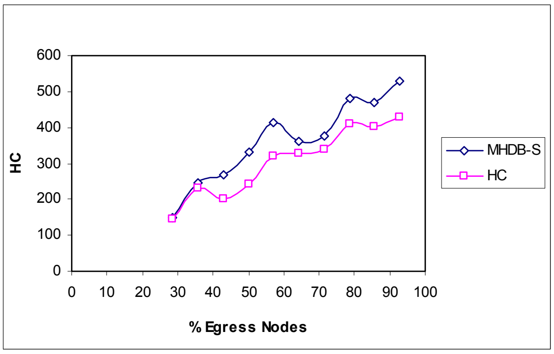


Fig. 7.8c. HC of MHDB-S and HC by 66%

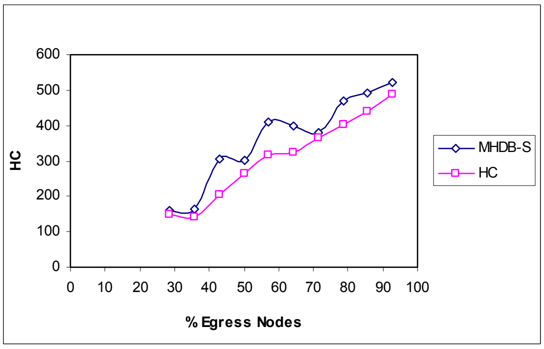


Fig. 7.8d. HC of MHDB-S and HC by 100%

For the other objective functions (MLU, DL, and BC) the behaviour is similar to that presented in the previous section, i.e. in the graphs with non-minimized functions. In this case, in the HC model MLU, DL and BC are not minimized. In this case it was also observed that in most cases model MHDB-S behaved better than the values observed for the non-minimized functions.

Figure 7.9 shows the normalized HC values for different flow fractions and the number of egress nodes. In this case it can be observed that model MHDB-S behaves at a maximum 27%

compared to model HC. This mean that the MHDB-S model optimizes the HC function too and the values are very close in comparison when the HC function is optimized only. In this figure we are testing up 200% tot he flow fraction because int this cas ewe are congestioning the network.

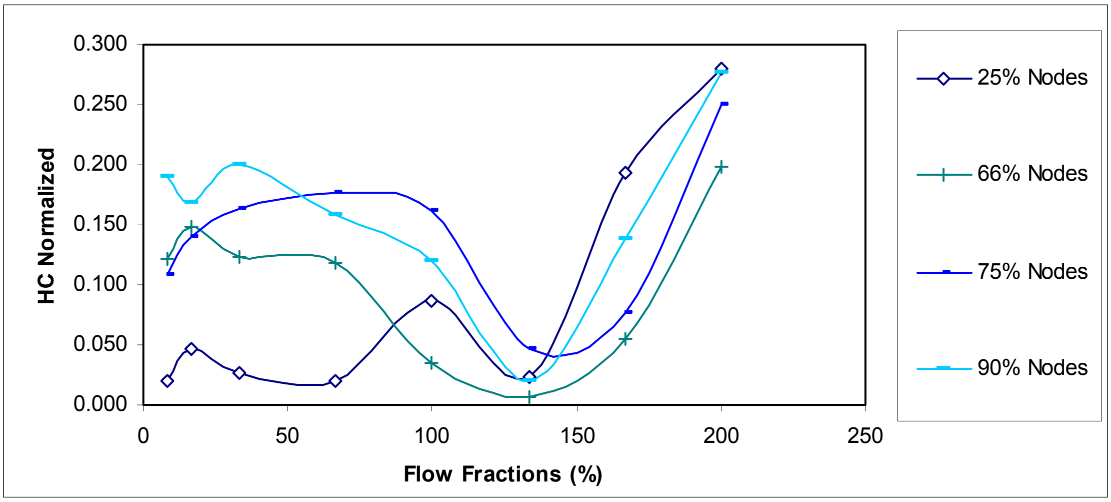


Fig 7.9. Normalized HC versus Flow Fractions (%) with different percentages of egress nodes in the NSF Topology minimizing HC

Analysing all normalized values for MHDB-S model and the HC model it is observed that model MHBD-S compared to:

- Function MLU, model MHBD-S behaves on average 10% better and in the best cases 60% better. The reasons area the same than the MLU analysis.
- Function DL, model MHBD-S behaves on average 20% better and in the best cases 80% better.
- Function BC, model MHBD-S behaves on average 10% better and 60% in the best cases better.

7.3.3 Comparative analysis of MHDB-S versus DL

In figure 7.10 the comparative graph between model MHDB-S and function DL is shown. It can be observed that model MHDB-S behaves with values closer to those shown by function DL. The maximum difference observed is 35% of the maximum value found for function DL. For the other functions (MLU, HC and BC) the behaviour is similar to that presented in the previous section where the graphs whose functions were not minimized were shown. In this case it was also observed that in most cases model MHDB-S behaved better than the values presented when only function DL was minimized.

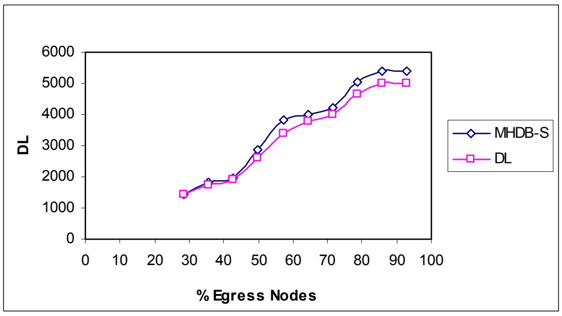


Fig. 7.10a. DL of MHDB-S and DL by 15%

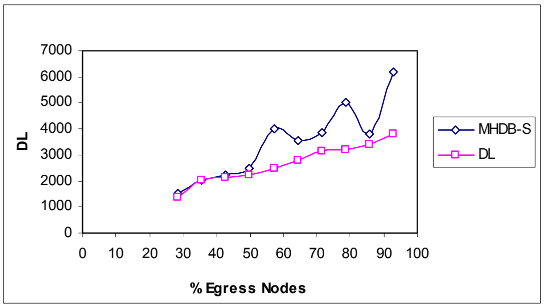


Fig. 7.10b. DL of MHDB-S and DL by 33%

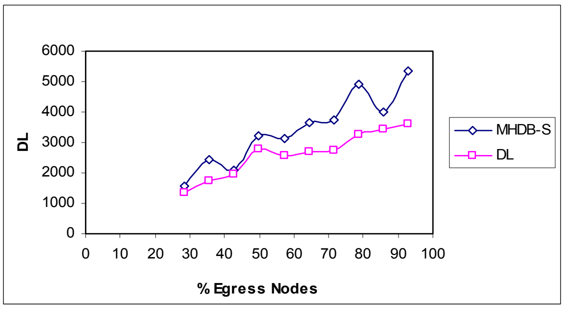


Fig. 7.10c. DL of MHDB-S and DL by 66%

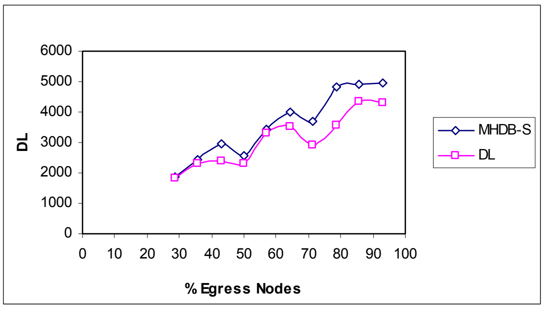


Fig. 7.10d. DL of MHDB-S and DL by 100%

Figure 7.11 shows the normalized DL values for different flow fractions and number of egress nodes. In this case it is observed that model MHDB-S behaves at a maximum 27% compared to model DL. This mean that the MHDB-S model optimizes the DL function too and the values are very close in comparison when the DL function is optimized only.

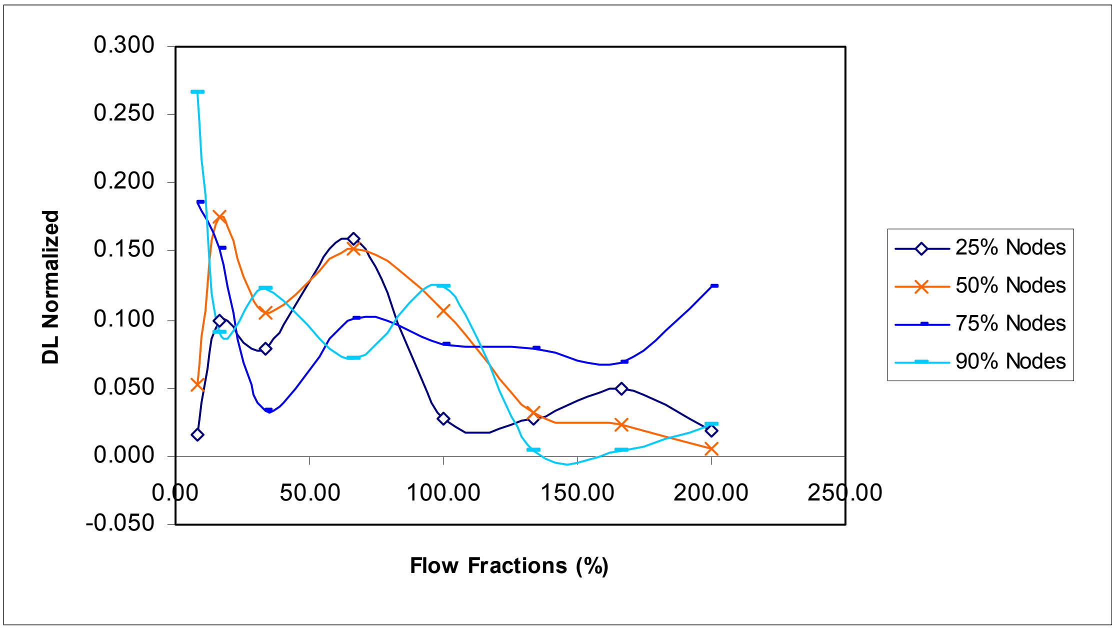


Fig 7.11. Normalized DL versus Flow Fractions (%) with different percentages of egress nodes in the NSF Topology minimizing DL

When analysing all the normalized values for model MHDB-S and model DL, it is observed that model MHDB-S compared to:

- Function MLU, model MHBD-S behaves on average 5% better and in the best cases 40% better.
- Function HC, model MHBD-S behaves on average 20% better and in the best cases 80% better.
- Function BC, model MHBD-S behaves on average 10% better and in the best cases 60% better.

7.3.4 Comparative analysis of MHDB-S versus BC

In figure 7.12 the comparative figures between model MHDB-S and function BC are shown. It can be observed that model MHDB-S behaves with values closer to those shown by function BC. The maximum difference observed is 30% the optimal value found for function BC. For the other functions (MLU, HC and DL) the behaviour is similar to that presented in the previous section where the graphs whose functions were not minimized were presented. In this case it was also observed that in most cases model MHDB-S behaved better than the values presented when only function BC was minimized.

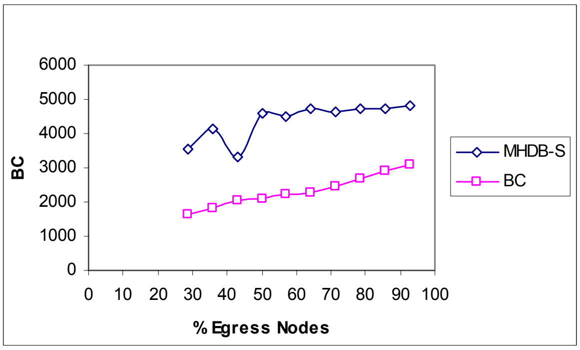


Fig. 7.12a. BC of MHDB-S and BC by 15%

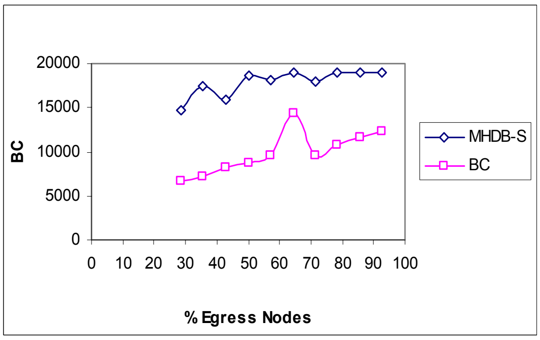


Fig. 7.12b. BC of MHDB-S and BC by 33%

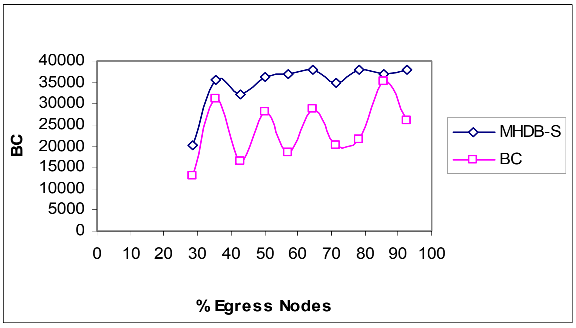


Fig. 7.12c. BC of MHDB-S and BC by 66%

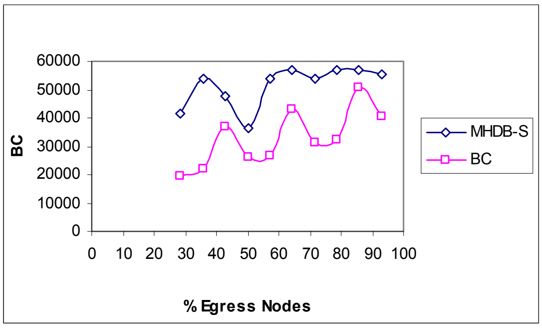


Fig. 7.12d. BC of MHDB-S and BC by 100%

Figure 7.13 shows the normalized BC values for different flow fractions and number of egress nodes. In this case it is observed that model MHDB-S behaves at a maximum 32% compared to model BC. This mean that the MHDB-S model optimizes the BC function too and the values are very close in comparison when the BC function is optimized only. In summary, the analysis of the four different functions show the same behavior to the MHDB-S model is compared with the models when just one function is optimized.

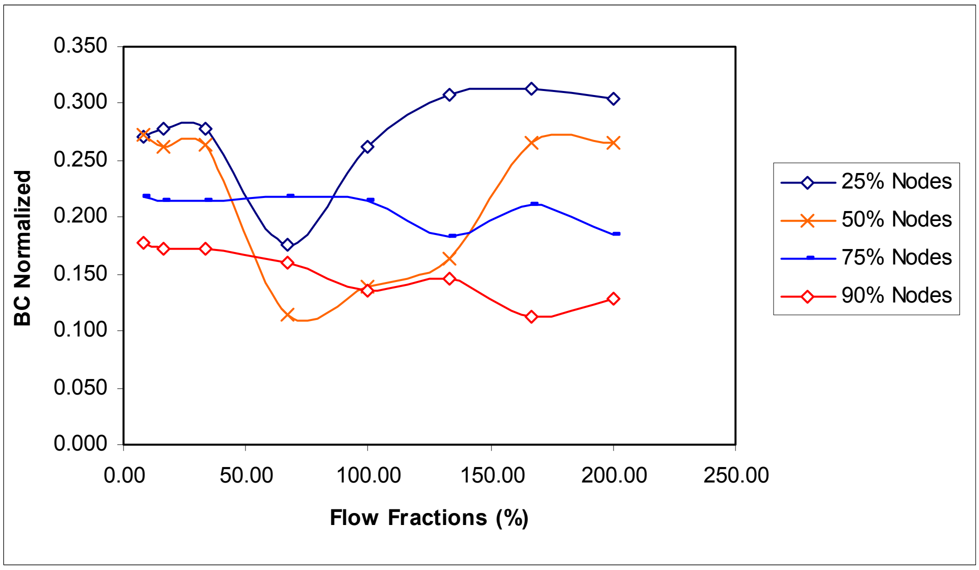


Fig 7.13. Normalized BC versus Flow Fractions (%) with different percentages of egress nodes in the NSF Topology minimizing BC

When analysing all the normalized values for model MHDB-S and model BC it is observed that model MHDB-S compared to:

- Function MLU, model MHBD-S behaves on average 50% better and in the best cases 100% better.
- Function HC, model MHBD-S behaves on average 5% better and in the best cases 80% better.
- Function DL, model MHBD-S behaves on average 5% better and in the best cases 80% better.

7.3.5 Comparative analysis of MHDB-S versus other functions with more than one objective

In this section model MHDB-S is compared with other models with more than one objective, which have been presented in other research works.

In this case the behaviour typically observed is that model MHDB.S behaves in a similar way to model DL-BC in DL and BC functions. But model MHDB-S shows better behaviour for functions MLU and HC. These trends are given for all flow percentages and for all three topologies tested: NSF, SPRINT and UUNET.

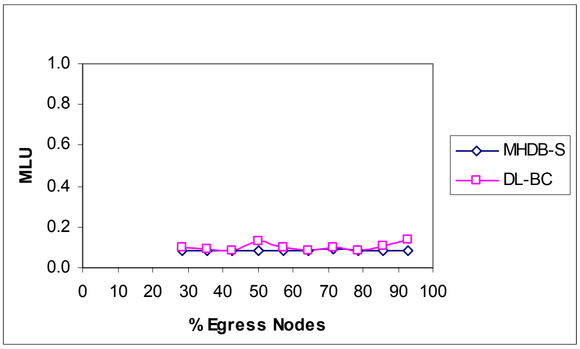


Fig. 7.14a. MHDB-S vs DL-BC comparing MLU

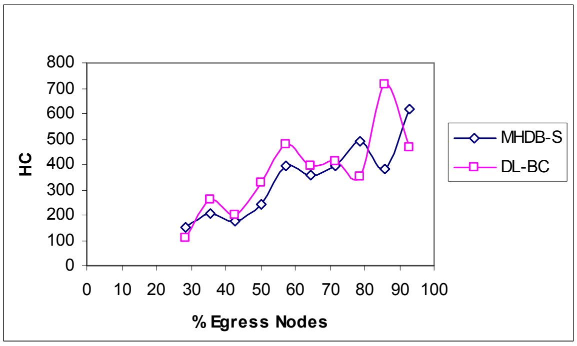


Fig. 7.14b. MHDB-S vs DL-BC comparing HC

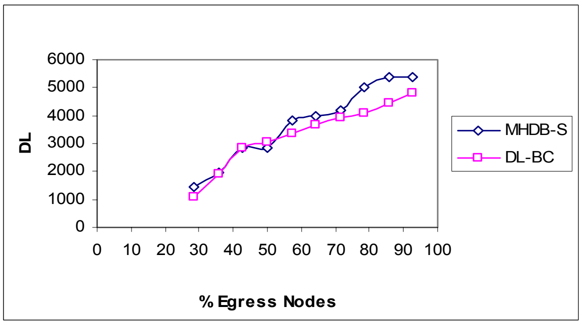


Fig. 7.14c. MHDB-S vs DL-BC comparing DL

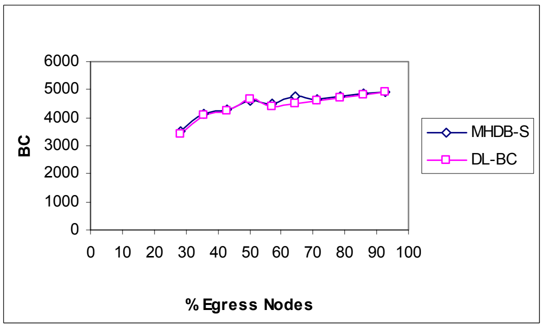


Fig. 7.14d. MHDB-S vs DL-BC comparing BC

Secondly, model MHDB was compared with model HC-BC [SON03] in which only hop count (HC) and Bandwidth consumption (BC) are minimized.

In this case the behaviour typically observed is that model MHDB-S behaves in a similar way to model HC-BC in functions HC and BC. But model MHDB-S shows better behaviour for functions MLU and DL. These trends are given for all flow percentages and for the three typologies tested: NSF, SPRINT and UUNET.

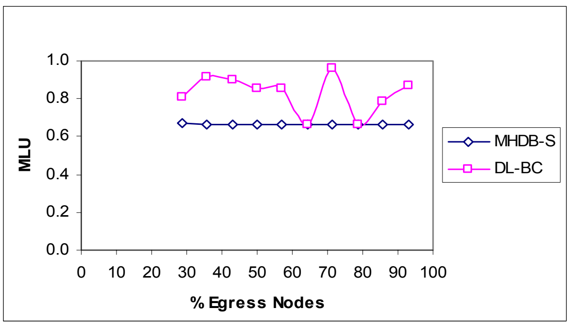


Fig. 7.15a. MHDB-S vs HC-BC comparing MLU

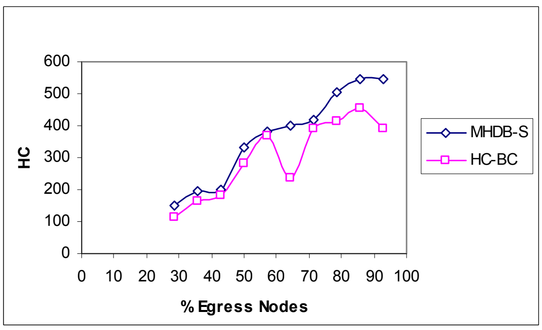


Fig. 7.15b. MHDB-S vs HC-BC comparing HC

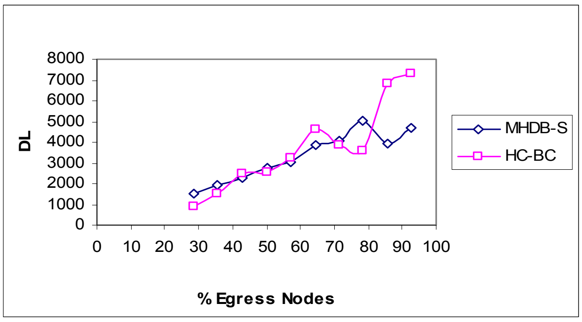


Fig. 7.15c. MHDB-S vs HC-BC comparing DL

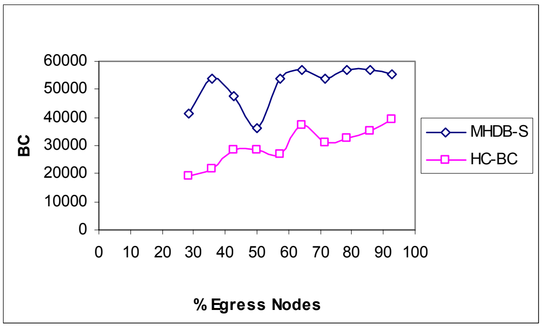


Fig. 7.15d. MHDB-S vs HC-BC comparing BC

Thirdly, model MHDB-S was compared with model MLU-HC-DL [ABO98] where maximum link utilization (MLU), hop count (HC) and bandwidth consumption (BC) are minimized.

In this case the behaviour typically observed was that model MHDB-S behaved in a very similar way to model MLU-HC-DL in functions MLU, HC and DL. But model MHDB-S had better behaviour for functions MLU and DL. These trends remained the same for all the flow percentages and for the three tested typologies: NSF, SPRINT and UUNET.

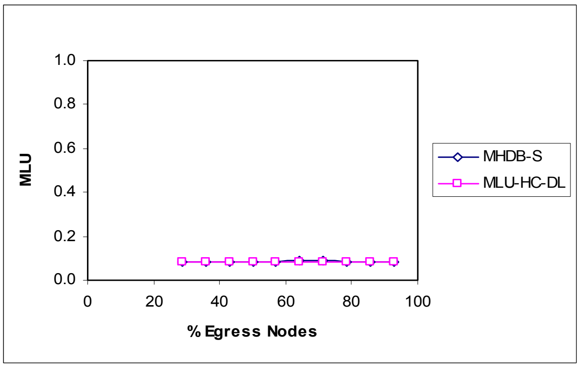


Fig. 7.16a. MHDB-S vs MLU-HC-DL comparing MLU

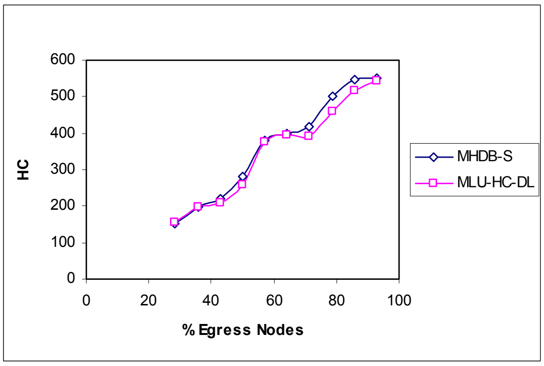


Fig. 7.16b. MHDB-S vs MLU-HC-DL comparing HC

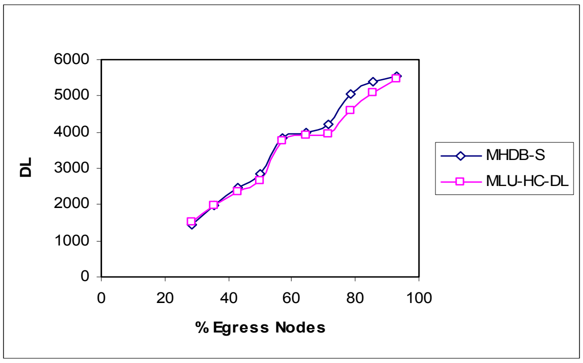


Fig. 7.16c. MHDB-S vs MLU-HC-DL comparing DL

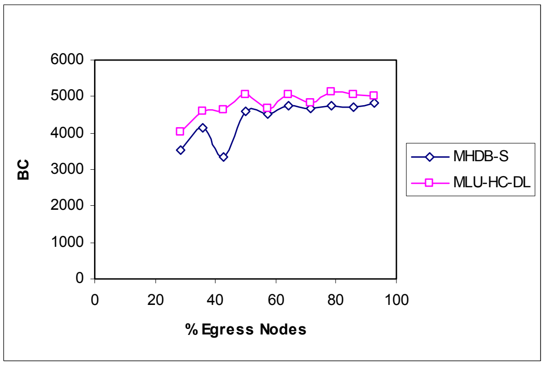


Fig. 7.16d. MHDB-S vs MLU-HC-DL comparing BC

7.4 Comparing static model (MHDB-S) with Heuristic (MMR-S)

In this section the results of the static model versus those of the (MMR-S) algorithm are shown. In figures 17 the observed behaviour for topology NSF is shown.

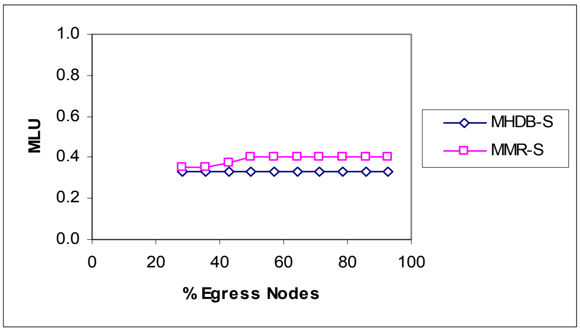


Fig. 7.17a. MHDB-S vs MMR-S comparing MLU

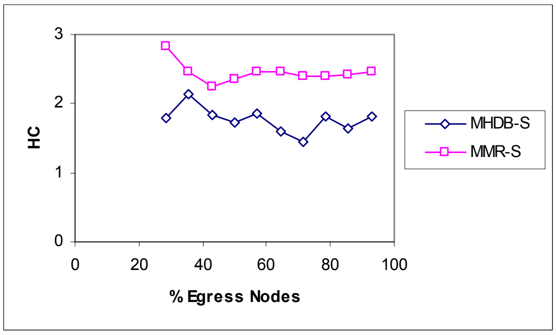


Fig. 7.17b. MHDB-S vs MMR-S comparing HC

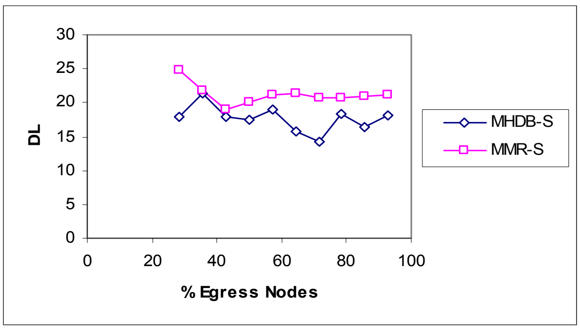


Fig. 7.17c. MHDB-S vs MMR-S comparing DL

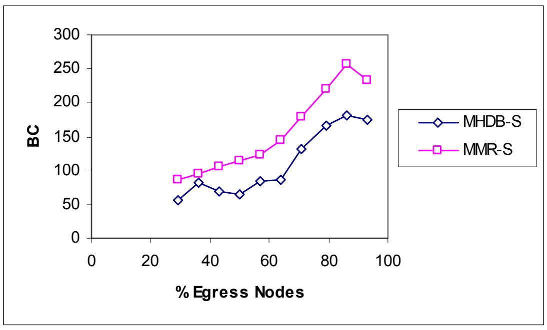


Fig. 7.17d. MHDB-S vs MMR-S comparing BC

When analyzing all the normalized values for model MHDB-S and MMR-S algorithm it is observed that model MHDB-S compared to:

- Function MLU, model MHDB-S behaves on average 5% better and in the best cases 80% better. This behavior is due to the main objective in our research is the MLU function in both cases (MHDBS-S model and MMR-S algorithm).
- Function HC, model MHDB-S behaves on average 30% better and in the best cases 70% better. This behavior is normal because the MHDB-S model optimize these functions while MMR-S algorithm is an approximation to the ideal behavior.
- Function DL, model MHDB-S behaves on average 15% better and in the best cases 80% better. It is the same behaviour as the HC analysis..
- Function BC, model MHDB-S behaves on average 30% better and in the best cases 60% better. It is the same behaviour as the HC analysis.

The SPRINT and UUNET topologies showed the same trends.

7.5 Comparison of Proposal of Dynamic Case (MMR-D) versus the Proposal of Static case (MMR-S)

In this section, we present the results of comparing the algorithm to solve the static model (MMR-S) and the algorithm to solve the dynamic model MMR-D. In this case, the tests were divided into two types. In the first kind of test, the three topologies used in the previous tests (NSF, SPRINT and UUNET) were used as references. In the second type of test, three kinds of randomly generated topologies with base sizes of 50, 100 and 200 nodes, were used as references. In this case, the entering or discharged nodes from the multicast flow transmission were chosen randomly. The comparison between the static case and the dynamic one was carried out as follows:

- The first execution performed was in the static case.
- Later, whether a node randomly entered or discharged from the transmission flow was randomly selected.
- For each randomly selected node, the dynamic model was executed again in order to add that node only.

- For each randomly selected node, the static model was executed completely, that is, all results were calculated again for both the previous egress nodes and the new entering node.

In both cases, we obtained results indicating the possible nature of flow transmission trees.

With respect to the results observed in the first type of test for topologies NSF, SPRINT and UUNET, the behaviour is shown in Figures 7.18.

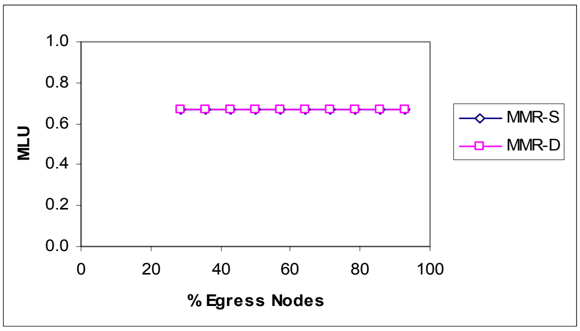


Fig. 7.18a. MMR-S vs MMR-D comparing MLU

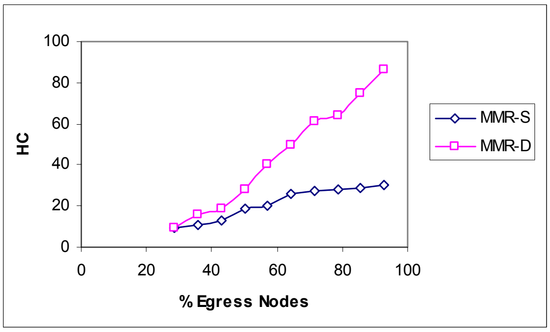


Fig. 7.18b. MMR-S vs MMR-D comparing HC

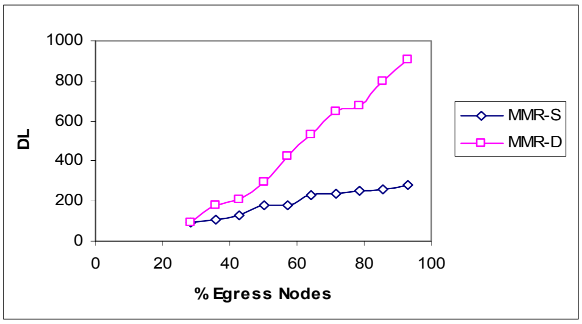


Fig. 7.18c. MMR-S vs MMR-D comparing DL

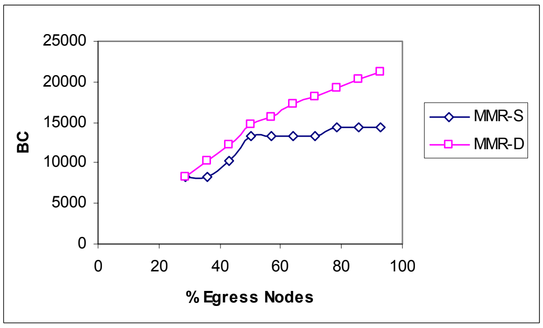


Fig. 7.18d. MMR-S vs MMR-D comparing BC

Now, with respect to the tests of random networks with 50 nodes, the behaviour is shown in Figures 7.19.

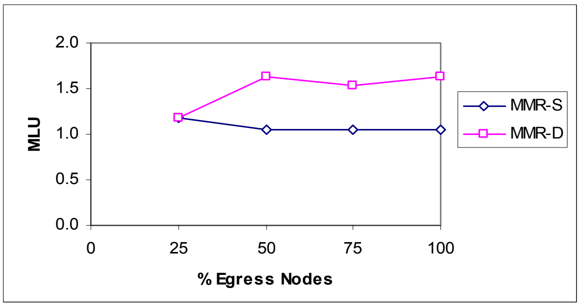


Fig. 7.19a. MMR-S vs MMR-D comparing MLU

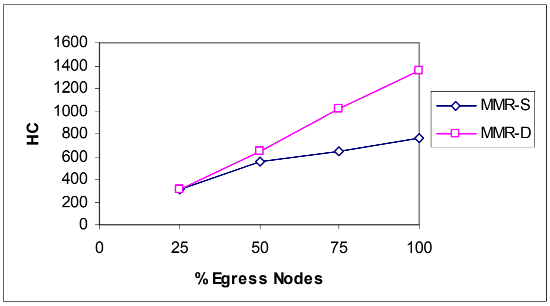


Fig. 7.19b. MMR-S vs MMR-D comparing HC

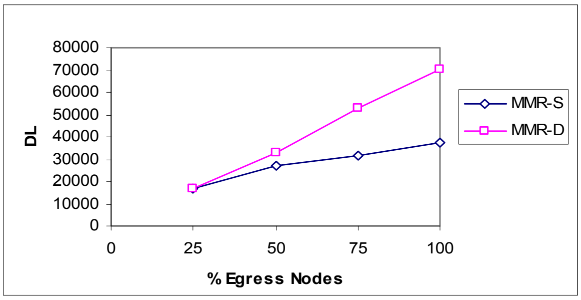


Fig. 7.19c. MMR-S vs MMR-D comparing DL

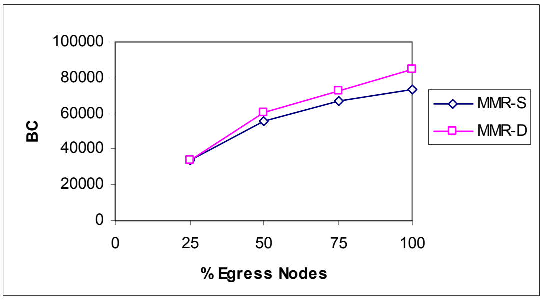


Fig. 7.19d. MMR-S vs MMR-D comparing BC

With respect to the tests of random networks with 100 nodes, the behaviour is shown in Figures 7.20.

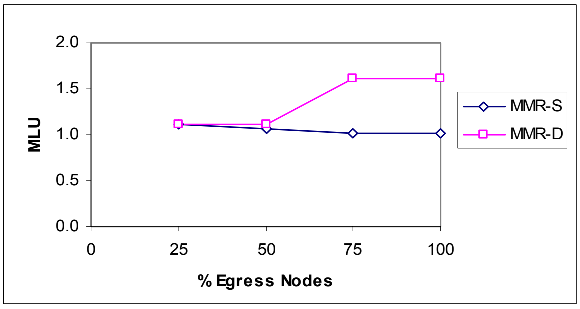


Fig. 7.20a. MMR-S vs MMR-D comparing MLU

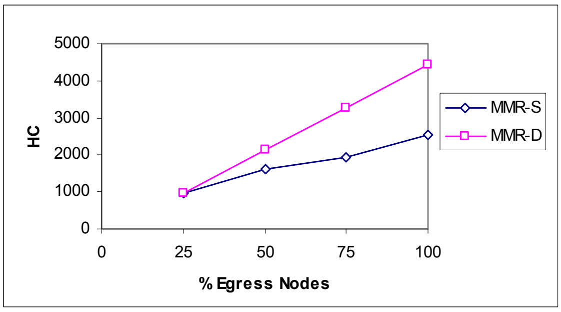


Fig. 7.20b. MMR-S vs MMR-D comparing HC

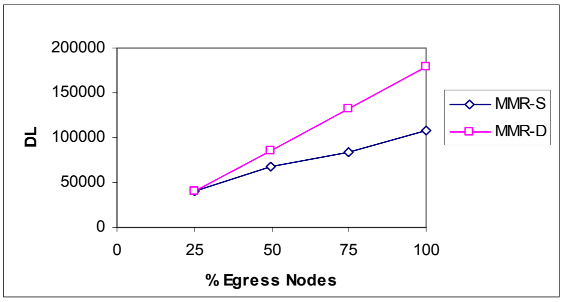


Fig. 7.20c. MMR-S vs MMR-D comparing DL

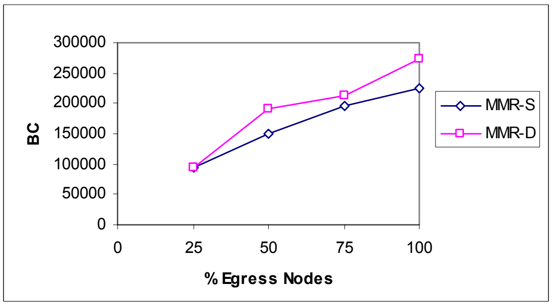


Fig. 7.20d. MMR-S vs MMR-D comparing BC

By analysing all the normalized values for model MMR-S and the algorithm MMR-D, it can be observed that model MMR-S compared to:

- Function MLU, MMR-S behaves on average in a similar way to both algorithms and in the best cases the MMR-S algorithm 10% better. This behaviour is because the main objective of our research is the MLU function in both algorithms.
- Function HC, the MMR-S behaves in the best cases 80% better. This behaviour is normal because the MMR-S algorithm optimises these functions while MMR-D is an approximation of the value given by the MMR-S algorithm.
- Function DL, model MMR-S behaves in the best cases 80% better. This behaviour is the same as that observed in the HC function.
- Function BC, model MMR-S behaves in the best cases 100% better. This behaviour is the same as that observed in HC function.

7.6 Comparison of the multi-objective solution for Static Case

In this section we present the results obtained for GMM-model. The GMM model considers the static case and was solved through a MOEA.

In these tests, the same topologies (NSF, SPRINT and UUNET) were used, and in this way the results obtained can be compared with those of the models and heuristics presented in previous chapters. In addition, the results were compared with larger randomly generated topologies.

As these two new algorithms are multi-objective, the result is not just a solution but a set of non-dominated solutions. In this section, the minimum values found by these algorithms are presented in order to compare them with the previously obtained values.

To perform these tests, each case was carried out 10 times on average and the solutions found were kept in a repository. Once a case was performed several times, a dominance analysis was carried out on the solutions found.

We followed the same test design as that used in the previous sections. For each type of test, the flow percentage was increased, and for each flow percentage, the number of egress nodes was increased. The ingress of a new egress node was randomly selected, but it follows the same pattern as the one analysed in previous sections so that we can compare the results with those previously obtained.

For the NSF, SPRINT and UUNET topologies, on average, around 300 non-dominated solutions were obtained.

Table 7.1 shows 20 solutions out of the 320 obtained using the algorithm GMM for a flow of 8% network capacity and 4 destination nodes. Among these solutions we have highlighted those that present the minimum values for each target function. The Table 7.1 shows the values when all nodes were added through the solution by using MOEA.

Table 7.1
GMM model using MOEA

ID	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}	ϕ_{11}
1	8,33%	137	5,7	9	6	1335	18,2	84	40	2592	6
2	8,33%	243	5,5	11	7	2535	10	124	93	2640	11
3	8,33%	39	4,8	8	0	374	46,7	73	0	2048	2
4	8,33%	35	4,3	7	0	365	45,6	79	0	2176	2
5	16,6%	31	3,8	7	0	310	38,7	59	0	2176	2
6	16,6%	31	3,8	6	0	358	44,7	70	0	2560	2
7	16,6%	37	4,6	7	0	338	42,2	62	0	2048	2
8	16,6%	47	3,9	7	4	468	26,6	56	15	2144	3
9	16,6%	49	4,1	6	2	432	24,3	56	11	2176	3
10	16,6%	54	4,5	7	1	517	28,5	65	15	2176	3
11	16,6%	58	4,8	8	1	507	26,5	70	17	2512	3
12	16,6%	51	4,2	7	2	484	27,4	61	15	2096	3
13	13,5%	87	4,4	7	2	806	17,1	71	26	2352	5
14	11,5%	71	4,4	7	2	645	21,7	71	26	2352	4
15	13,5%	50	4,2	7	4	488	28,6	62	15	2144	3
16	16,6%	50	4,2	7	4	490	28,3	62	15	2096	3
17	10,4%	56	4,7	7	2	499	29,3	71	26	2464	3
18	12,5%	87	4,4	7	3	821	16,0	65	30	2032	5
19	11,4%	251	5,2	10	7	2494	8,4	118	80	2512	12
20	13,5%	64	4,0	7	4	648	20,1	55	22	2336	4

In Table 7.1 each column represents a target function of the GMM model. Each shaded square represents the minimum value found, at least by a Pareto front solution, for each target solution.

To use and to implement just one solution, as was explained in the chapter 6 (Search before decision making methodology), a rational human decision maker determines preferences among the conflicting objectives and search the best alternatives. The results shown in the sections between 7.3 and 7.5 were realized using a Decision making before search methodology

The values shown in the Table 7.1 are very close to the minimum values obtained by the MHDB-S model and the MMR-S algorithm. But, with this kind of solutions it is possible to find the optimal pareto set instead of just one solution given by the others methods.

7.7 Comparison of the multi-objective solution for Dynamic Case

In this section we show the results obtained for the Dynamic GMM-model. To solve the Dynamic GMM model, a probabilistic BFS (Breadth First Search) was developed. The parameters in this case were exactly the same as those of the MOEA.

Table 7.2 shows a sample of 20 solutions resulting from solving the dynamic case with probabilistic BFS. In this case, the values are shown for when the last node was added with probabilistic BFS.

Table 7.2
Dynamic GMM model using BFS Probabilistic

ID	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}	ϕ_{11}
1	8,33%	145	7,1	8	4	1245	20,7	75	30	2322	7
2	8,33%	135	4,3	9	6	1567	17,0	88	25	2530	10
3	10,2%	43	5,2	7	0	450	50,4	70	0	2136	3
4	16,6%	45	5,3	7	0	395	52,4	80	0	2435	2
5	16,6%	31	3,8	7	0	310	38,7	59	0	2176	2
6	16,6%	31	3,8	6	0	358	44,7	70	0	2560	2
7	16,6%	37	4,6	7	0	338	42,2	62	0	2048	2
8	16,6%	50	4,0	10	0	570	32,5	70	0	2087	3
9	16,6%	49	4,1	6	2	432	24,3	56	11	2176	3
10	18,0%	66	5,2	6	3	678	29,3	73	11	2456	6
11	18,0%	58	4,6	9	0	550	24,5	72	12	2434	4
12	11,4%	251	5,2	10	7	2494	8,4	118	80	2512	12
13	16,6%	51	4,2	7	2	484	27,4	61	15	2096	3
14	13,5%	64	4,0	7	4	648	20,1	55	22	2336	4
15	11,4%	112	6,2	5	3	1020	33,3	82	11	2467	9
16	12,5%	87	4,4	7	3	821	16,0	65	30	2032	5
17	16,6%	43	4,3	8	4	1245	45,6	72	12	2145	4
18	11,5%	71	4,4	7	2	645	21,7	71	26	2352	4
19	16,6%	54	5,1	7	3	682	31,4	86	12	2420	8
20	16,6%	50	4,2	7	4	490	28,3	62	15	2096	3

In Table 7.2 each column represents a target function of the Dynamic GMM. Each shaded square represents the minimum value found, at least by the Pareto front solution, for each target solution.

The minimum values for each function obtained by the D-GMM algorithm (Table 7.2) are very close to the minimum values obtained by the GMM-MOEA (Table 7.1). The best value of the functions $\phi_1, \phi_2, \phi_3, \phi_5, \phi_9, \phi_{11}$ given by the D-GMM algorithm is not improved by the GMM-MOEA. The other kinds of tests (using UUNET, SPRINT and random topologies) show the same behavior.

7.8 Comparison between static and dynamic using multi-objective metrics

In this section the MOEA algorithm to give a solution to the GMM-model and the BFS probabilistic algorithms to give a solution to the Dynamic GMM-model are compared using multi-objective metrics.

- Overall Nodominated Vector Generation (ONVG). This metric measures the total number of nondominated found and is defined as: $ONVG \overset{\Delta}{=} |Y_{known}|_c$.
- Overall Nondominated Vector Generation Ratio (ONVGR). This metric measures the ratio between the number of nondominated found and the number of nondominated existents. The metric is defined as: $ONVGR \overset{\Delta}{=} \frac{ONVG}{|Y_{true}|_c}$

In our tests, the Y_{know} are the optimal values found by the BFS algoritms and the Y_{true} are the optimal values found by the MOEA algorithms and the idea of this kind of test it is to prove the nearness between the BFS algorithms and the MOEA algoritms to give a solution to the dynamic case. The Table 7.3 shows some values of 10 executions.

Table 7.3
Multi-objective metrics

Execution	ONVG	ONVGR
1	210	0.75
2	250	0.85
3	280	0.64
4	190	0.78
5	310	0.65
6	245	0.82
7	278	0.76
8	251	0.67
9	266	0.73
10	280	0.79

With this kind of metrics it is possible to observe the nearness in terms of number of solution found between the solution obtained by the MOEA algoritms and the BFS algorithm to give a solution to the dynamic case.

7.9 Summarized comparative analysis, correlation analysis, and confidence intervals

Table 7.4 shows the results obtained by minimizing each one of the target functions (MLU, HC, DL and BC); the minimum values obtained with model MHDB-S, and the minimum values obtained with MOEA. In this case, it can be observed that all minimum values of the presented models were found at least by one of the solutions given by MOEA.

Table 7.4
MOEA vs Other solutions

	ϕ_1 (MLU)	ϕ_2 (HC)	ϕ_6 (DL)	ϕ_{10} (BC)
MOEA	8,33%	31	310	2032
MHDB-S	8,33%	35	325	2120
MHDB-D	12,0%	45	370	2205
MMR-S	8,33%	38	330	2150
MMR-D	12,0%	50	390	2300
MLU	8,33%			
HC		31		
DL			310	
BC				2032

Once the tests were carried out, it was observed that there is some type of correlation between some of the functions. In the table 7.5, the correlation values between functions are shown.

Table 7.5
Correlation Analysis

	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}	ϕ_{11}
ϕ_1	-0,23	-0,12	-0,24	-0,26	-0,25	0,19	-0,28	-0,19	0,09	-0,20
ϕ_2		0,70	0,78	0,80	0,95	-0,81	0,78	0,78	0,56	0,98
ϕ_3			0,78	0,48	0,67	-0,35	0,62	0,41	0,52	0,59
ϕ_4				0,82	0,76	-0,58	0,90	0,73	0,47	0,72
ϕ_5					0,84	-0,83	0,85	0,93	0,34	0,80
ϕ_6						-0,83	0,79	0,78	0,53	0,98
ϕ_7							-0,69	-0,78	-0,36	-0,85
ϕ_8								0,83	0,40	0,75
ϕ_9									0,33	0,77
ϕ_{10}										0,53

It can be observed that in Table 7.5, there is a high correlation between the following target functions:

- Total Hop Count (ϕ_2), termed HC in models MHDB-S and MHDB-D, and the Hop Count Average (ϕ_3) with a correlation value of 0.70.

- Total Hop Count (ϕ_2) and Maximal Hop Count (ϕ_4) with a correlation value of 0.78.

- In addition, the Total Hop Count function (ϕ_2) had a high correlation with the other target functions (ϕ_3, ϕ_4, ϕ_5, ϕ_6, ϕ_7, ϕ_8, ϕ_9, ϕ_{11}). This means that if we are minimizing this function, we would be also minimizing the other target functions.

- The same effect, but not so sharp, it is observed for the Total Delay function (ϕ_6), which is also a function of Models MHDB-S and MHDB-D.

Finally, the analysis of confidence intervals was carried out for each of the analysed results. Assuming independence among the algorithm types and normality of data, by applying an F test at 95% for the tests of the three topologies and random topologies, it was proved that for each algorithm performed the variances were different ($\sigma_1 \neq \sigma_2$). By using a t test at 95%, it was proved that there are differences between means ($\mu_1 \neq \mu_2$). The confidence intervals for these differences are shown in the following table.

Table 7.6
Confidence intervals

	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}	ϕ_{11}
μ	0,12	136,2	4,9	8,6	5,1	1362	16,7	88,0	52,4	2443	6,8
σ	0,02	60,2	0,4	1,2	1,8	634,1	7,1	16,8	24,2	197,9	2,7
IC(95%)	0,002	6,5	0,04	0,1	0,2	68,2	0,8	1,8	2,6	21,3	0,3

7.10 Conclusions

In this chapter we have presented the results and can make the following conclusions:

- The stability shown by model MHDB-S with respect to the 4 minimized functions when compared to other models in which 1, 2 or 3 target functions are minimized, models that have been proposed by previous studies.

- The closeness of model MHDB-S values compared to the minimum value of each target function and the improvement with respect to the values of the other target functions, which are not minimized in these cases.

- The closeness of the values of the heuristic MMR-S and MMR-D to the values presented by analytical models MHDB-S and MHDB-D. At this point, it is important to highlight that some algorithmic solutions to analytical models have been proposed whose solutions are by nature NP-Hard. Even so, it is important to mention that the solution presented by model MHDB-S and the algorithm MMR-D is not the best solution because it can not estimate all the nodes of the multicast tree to perform the transmission to the new egress node that is entering. It can be seen in the results that the proposed heuristic (probabilistic BFS) with the Dynamic-GMM model shows better values than the ones observed for the MMR-D algorithm when giving a solution to model MHDB-S.

- It is important to mention the scalability of the solution because tests performed with large topologies using the random generation of topologies show similar behaviour to those results obtained with the NSF, SPRINT and UUNET topologies.

- The importance of both the GMM model and the computer solution using MOEA, whose values can reach the minimum values presented by analytical models and heuristics (MHDB-S and MMR-S), and also all the possible solutions found through each Pareto front.

- Finally, the closeness of the values found by the multi objective algorithm for the generalized dynamic case compared to the values found by the GMM model using the MOEA algorithm.

Chapter 8 Conclusion and Future Work

8.1 Conclusion

How in different networks is possible to have congestion and many multicast applications, such as audio and videoconferencing or collaborative environments and distributed interactive simulation, have multiple quality of service requirements on bandwidth, packet delay, packet loss, cost, etc. This thesis presents a proposal to optimize the network resources through the load balancing technique. In this case, this thesis presents a scheme to transmit the multicast flow by more than one tree instead of by just one tree how is the functioning of the current multicast routing protocols such as: DVMRP, MOSPF, BGMP, PIM-DM, PIM-SM and CBT.

In the research process, at the beginning, just one objective function was chosen. This function was the Maximum Link Utilization (MLU) and through this function was possible to do the load balancing. With this function is possible to create more than one tree. After, three more functions were adding: end-to-end delay (DL), hop count (HC) and bandwidth consumption (BC). With these functions were possible to improve the following aspects: long paths, paths with high delay and it reduce the bandwith consumption in every tree. Finally, it was necessary to improve the analytical model and the computational solution. The new analytical model considers eleven different functions and in this case, it is a generalized optimization model. The new computational solution was realized using MOEA and in this case it is possible to find the optimal Pareto front and it is possible to find solutions in nonconvex solutions set.

In multicast flows, the egress nodes can enter or leave of the tree transmission. In the proposals presented previously were considered the dynamic case. In the first proposal, the ingress node and the current egress nodes just can be the new point of transmission to the new egress node. In the last proposal, that is, in the generalized model is possible to use the ingress node and anyone current node that belong to the transmission tree.

The main aspects considered in this thesis are:

- Load Balancing in Multicast Flows.
- The analytical models
- The computational solutions using heuristic algorithms and MOEAs.
- The static and dynamic functioning of the multicast flows.

Many experiments have been carried out in order to test the proposals presented in this thesis. In these tests were considered different real networks topologies such as: NSF, SPRINT and UUNET and others large random topologies. Different percentages of flows were chosen. Several percentages of egress nodes in comparison with the total nodes of the topology were selected. With these kind of scenarios was possible to probe that our proposal presents a good behaviour in comparison with current multicast routing protocol and with respect to the others proposals.

The results obtained from our thesis show that the MHDB-S and MHDB-D models present a very close beaviour in comparison with the function that is being minimized, but these models present a better beaviour in comparison with the others objective functions. The computational solutions to give solution to the analytical models presented very close results in comparison with the MHDB-S and MHDB-D models. This means that the algorithms proposed give very good solutions. Finally, the MOEA solution gives an optimal Pareto front and in this case the results given by the previous algorithms are similar in comparison with the minimum values found by the MOEA solution. With the MOEA solution is possible to have a nondominated solution set, however, in the algorithms just one solution is obtained and this functioning is a limitation of the first proposal.

8.2 Future Work

There are many issues that have been left as future work throughout this thesis. The most significant ones are the following:

- This thesis can be applied in MPLS technology, but it would be interesting to apply these models in GMPLS technology in the photonic label.

- Out multi-objective optimization models can be extended to the wireless technology and in this case, it would be necessary to analize the different functions that affect the flow transmission in wireless technology.

- In this thesis were proposed three different heuristic to give solutions in the static and dynamic case and one meta-heuristic is proposed using MOEA to give a solution to the generalized model. It could be interesting to probe other kind of meta-heuristic to look for solutions to the same problem. These kinds of meta-heuristic could be: Ant Colony, Memetic Algorithms, Tabu Search, Simulated Annealing, etc.

- Due to the load balancing technique is possible to have some problems such as: packet disordering. It could be interesting to analyse a proposal using effective Hashing functions to avoid this kind of problems.

- In computer networks is possible to have some problems as failed links or paths and different proposal to give solutions have been proposed. It could be interesting to apply this proposal of load balancing in the recovery proceedings due to in our proposal we have more than one transmission tree and these trees can be used like a protection tree.

- Our models are no-lineal models, but it would be interesting to convert our models as lineal model through approximation methods such as relaxation of the models.

- How future work will be possible to analyze an algorithm that gets advantage of the MPLS label stack features in order to reduce the number of labels used in P2MP LSPs. Some P2MP tunnels methods and their MPLS implementation drawbacks can be also discussed.

References

[ABO98] E. Aboelela, C. Douligeris. "Fuzzy Generalized Network Approach for Solving an Optimization Model for Routing in B-ISDN". IEEE CCECE 1998/CCGEI 1998.

[ABR02] H. Abrahamsson, B. Ahlgren, J. Alonso, A. Andersson, P. Kreuger. "A Multi Path Routing Algorithm for IP Networks based on Flow Optimisation". LCNS 2511. QoFIS 2002.

[AHU97] R. Ahuja, M. Kodialam, A. Mishra, J. Orlin. "Theory and Methodology, Computational Investigations of Maximum Flow Algorithms". European Journal of Operational Research. Published by Elsevier Science. 1997.

[AHU93] R. Ahuja, T. Magnanti, J. Orlin. "Network flows: Theory, algorithms and applications". Prentice-Hall. NJ. 1993.

[ALO02] S. Alouf, E. Altman, P. Nain. "Optimal on-line estimation of the size of a dynamic multicast group". IEEE Infocom. 2002.

[APP03] D. Applegate, M. Thorup. "Load optimal MPLS routing with N+M labels". IEEE INFOCOM 2003.

[ASH02] J. Ash, M. Girish, E. Gray, B. Jamoussi, G. Wright. "Applicability statement for CR-LDP". RFC 3213. Enero 2002.

[ASH02a] J. Ash, Y. Lee, P. Ashwood-Smith, B. Jamoussi, D. Fedyk, D. Skalecki, L. Li. "LSP modification using CR-LDP". RFC 3214. Enero 2002.

[AWD01] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow. "RSVP-TE: Extensions to RSVP for LSP tunnels". RFC 3209. Diciembre 2001.

[BAC00] T. Back, D. Fogel, T. Michalewicz. "Evolutionary Computation 1". IOP Publishing. 2000.

[BAL97] A. Ballardie. "Core Based Trees (CBT version 2) Multicast Routing". RFC 2189. Septiembre 1997.

[BAL97a] A. Ballardie. “Core Based Trees (CBT) Multicast Routing Architecture”. RFC 2201. Septiembre 1997.

[BAN01] N. Banerjee, S. Das. “Fast Determination of QoS Multicast Routes in Wireless Networks using Genetic Algorithm”. IEEE ICC 2001.

[BAR04] B. Baran, R. Fabregat, Y. Donoso, F. Solano, JL. Marzo. “Generalized Multiobjective Multitree model”. Research Report Girona University 2004.

[BAZ90] M. Bazaraa, J. Jarvis, H. Sherali. “Linear Programming and Network Flows”. John Wiley & Sons, 2nd ed, USA, 1990.

[BAZ93] M. Bazaraa, H. Sherali, C. Shetty. “Nonlinear Programming, Theory and Algorithms”. John Wiley & Sons, 1993.

[BHA02] S. Bhatnagar, S. Ganguly, B. Nath. “Label space reduction in multipoint-to-point LSPs for traffic engineering”. ECUMN2002.

[CAO00] Z. Cao, Z. Wang, E. Zegura. “Rainbow Fair Queueing: Fair Bandwidth Sharing Without Per-Flow”. IEEE INFOCOM 2000.

[CAO00a] Z. Cao, Z. Wang, E. Zegura. “Performance of hashing-based schemes for Internet load balancing”, IEEE INFOCOM 2000.

[CER04] S. Cerav-Erbas. “Traffic Engineering in MPLS Networks with Multiple Objectives: Modeling and Optimization”. Thesis of Master of Science in Industrial and Systems Engineering. Turkey. 2004.

[CET04] C. Cetinkaya, E. Knightly. “Opportunistic Traffic Scheduling Over Multiple Network Paths”. IEEE INFOCOM 2004.

[CHA03] D. Chakraborty, G. Chakraborty, N. Shiratori. “A dynamic multicast routing satisfying multiple QoS constraints”. International Journal of Network Management, John Wiley & Sons. 2003.

[CHE01] J. Chen, S. Chan. "Multipath Routing for Video Unicast over Bandwidth-Limited Networks". IEEE GLOBECOM 2001.

[CHO03] H. Cho, J. Lee, B. Kim. "Multi-path Constraint-based Routing Algorithms for MPLS Traffic Engineering". IEEE ICC 2003.

[COH78] J. Cohon. "Multiobjective programming and planning". New York: Academic Press. 1978.

[COL03] Y. Collette, P. Siarry. "Multiobjective Optimization, Principles and Case Studies". Springer. 2003.

[CRI04] J. Crichigno, B. Barán. "A Multicast Routing Algorithm using Multiobjective Optimization". IEEE ICT 2004.

[CRI04a] J. Crichigno, B. Barán. "Multiobjective Multicast Routing Algorithm". IEEE ICT 2004.

[CRI04b] J. Crichigno, B. Barán. "Multiobjective Multicast Routing Algorithm for Traffic Engineering". IEEE ICCCN 2004.

[CUI03] X. Cui, C. Lin, Y. Wei. "A Multiobjective Model for QoS Multicast Routing Based on Genetic Algorithm". ICCNMC 2003.

[CUI03a] Y. Cui, K. Xu, J. Wu. "Precomputation for Multi-constrained QoS Routing in High-speed Networks". IEEE INFOCOM 2003.

[DEB01] K. Deb. "Multi-Objective Optimiation using Evolutionary Algorithms". John Wiley & Sons. 2001.

[DEE98] Deering. "Protocol Independent Multicast Version 2 Dense Mode. Specification". draft-ietf-pim-v2-dm-01.txt. Noviembre 1998.

[DON03] Y. Donoso, R. Fabregat, L. Fàbrega. "Multi-Objective Scheme over Multi-Tree Routing in Multicast MPLS Networks". ACM/IFIP LANC 2003.

[DON03a] Y. Donoso, R. Fabregat. "Ingeniería de Tráfico aplicada a LSPs Punto-Multipunto en Redes MPLS". CLEI (Congreso Latinoamericano de Estudios en Informática) 2003. La Paz, Bolivia. October 2003.

[DON04] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Algorithm for Multicast Routing with Traffic Engineering". IEEE ICN 2004.

[DON04a] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Model and Heuristic Algorithm for Dynamic Multicast Routing". IEEE & VDE Networks 2004.

[DON04b] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Scheme for Dynamic Multicast Groups". IEEE ISCC 2004.

[DON04c] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Algorithm for Multicast Routing with Traffic Engineering." Telecommunication Systems Journal. Kluwer Publisher. 2004.

[DON04d] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Model and Heuristic Algorithm for Multipath Routing of Static and Dynamic Multicast Group." III Workshop on MPLS Networks. Girona, Spain. March 2004.

[DON04e] Y. Donoso, R. Fabregat, JL. Marzo. "Multicast Routing with Traffic Engineering: a Multi-Objective Optimization Scheme and a Polynomial Shortest Path Tree Algorithm with Load Balancing". Proceedings of CCIO -2004 Cartagena de Indias, Colombia, March 2004.

[DON05] Y. Donoso, R. Fabregat, JL. Marzo. "Multicast Routing With Traffic Engineering: A Multi-Objective Optimization Scheme And A Polynomial Shortest Path Tree Algorithm With Load Balancing." Annals of Operations Research. Kluwer Publisher. 2005.

[DON05a] Y. Donoso, R. Fabregat, F. Solano, JL. Marzo, B. Baran. "Generalized Multiobjective Multitree model for Dynamic Multicast Groups". IEEE ICC. Seoul. Korea. May 2005.

[EST98] D. Estrin. "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification". RFC 2362. Junio 1998.

[FAB04] R. Fabregat, Y. Donoso, F. Solano, JL. Marzo. "Multitree Routing for Multicast Flows: A Genetic Algorithm Approach". CCIA 2004.

[FAB04a] R. Fabregat, Y. Donoso, JL. Marzo, A. Ariza. “A Multi-Objective Multipath Routing Algorithm for Multicast Flows”. SPECTS 2004.

[FOR02] B. Fortz, M. Thorup. “Optimizing OSPF/IS-IS weights in a changing world”. JSAC, February 2002.

[GAM04] GAMS . Solver for large mathematical programming problems. http://www.gams.com

[GHA03] R. Ghanea-Hercock. “Applied Evolutionary Algorithms in Java”. Springer. 2003.

[GOL89] D. Goldberg. “Genetic Algorithms is Search, Optimization and Machine Learning”. Addison-Wesley Publishing. 1989.

[GUP03] A. Gupta, A. Kumar, R. Rastogi. “Exploring the trade-off between label size and stack depth in MPLS Routing”. IEEE INFOCOM 2003.

[HER03] R. Hernandez, C. Fernandez P. Baptista. “Metodología de la investigación”. Mc Graw Hill. 3rd edition. 2003

[HOR97] J. Horn. “Multicriteria decision making and evolutionary computation”. InT. Baeck, D. B. Fogel, and Z. Michalewicz, editors, Handbook of Evolutionary Computation. Institute of Physics Publishing, Bristol, UK, 1997.

[IM95] Y. Im, Y. Lee, S. Wi, K. Lee, Y. Choi, C. Kim. “Multicast Routing Algorithms in Hihg Speed Networks”. IEEE. 1995.

[INA99] J. Inagaki, M. Haseyama, H. Kitajima. “A Genetic Algorithm for Determining Multiple Routes and its Applications”. Circuits and Systems, ISCAS 1999.

[IZM02] R. Izmailov, D. Niculescu. “Flow splitting approach for path provisioning and path protection problems”. HPSR 2002.

[KIM02] C. Kim, Y. Choi, Y. Seok, Y. Lee. “A Constrained Multipath Traffic Engineering Scheme for MPLS Networks”. IEEE ICC 2002.

[KIM04] J. Kim, C. Kim, S. Seok, C. Kang. “Traffic Engineering using Adaptive Multipath-Forwarding Against Dynamic Traffic in MPLS Networks”. IEEE ICN 2004.

[KOY04] A. Koyama, L. Barolli, K. Matsumoto, B. Apduhan. “A GA-based Multi-purpose Optimization Algorithm for QoS Routing”. IEEE AINA 2004.

[LAY04] L. Layuan, L. Chunlin. “QoS Multicast Routing in Networks with Uncertain Parameter”. IEEE IPDPS 2004.

[LEE02] L. Lee, Y. Seok Y. Choi, C. Kim. “A Constrained Multipath Traffic Engineering Scheme for MPLS Networks”. IEEE ICC 2002.

[LEU98] Y. Leung, G. Li, Z. Xu. “A Genetic Algorithm for the Multiple Destination Routing Problems”. IEEE Transactions on Evolutionary Computation, Vol. 2, No. 4, November 1998.

[LI99] Y. Li, Y. Bouchebaba. “A new genetic algorithm for the optimal communication spanning tree problem”. Proceedings of Artificial Evolution: Fifth European Conference. LNCS 1829. 1999.

[MED01] A. Medina, A. Lakhina, I. Mata, J. Byers. “BRITE: An Approach to Universal Topology Generation”. In proceedings of the international workshop on modeling, analysis and simulation of computer and telecommunications systems. 2001.

[MON04] D. Montgomery. “Diseño y Análisis de Experimentos”. Limusa Wiley. 2nd edition. 2004.

[MOY94] J. Moy. “Multicast Extensions to OSPF”. RFC 1584. Marzo 1994.

[OH03] Y. Oh, D. Kim, H. Yoen, M. Do, J. Lee. “Scalable MPLS multicast using label aggregation in Internet broadcasting systems”. ICT 2003..

[POM04] D. Pompili, L. Lopez, C. Scoglio. “DIMRO: a DiffServ-Integrated Multicast algorithm for Internet Resource Optimization in source specific multicast applications”. ICC 2004.

[PUS00] T. Pusateri. “Distance Vector Multicast Routing Protocol”. draft-ietf-idmr-dvmrp-v3-10. Agosto 2000.

[RAG99] S. Raghavan, G Manimaran, C. Siva Ram Murthy. "A rearrangeagle algorithm for the construction of delay-constrained dynamic multicast trees". IEEE/ACM Transactions on Networking, Vol. 7, Nº.4, pp. 514-529, August 1999

[RAO98] N. Rao, S. Batsell. "QoS Routing via multiple Paths using Bandwidth Reservation". IEEE INFOCOM 1998.

[ROS01] E. Rosen, A. Viswanathan, R. Callon. "Multiprotocol Label Switching Architecture". RFC 3031. 2001.

[ROY02] A. Roy, N. Banerjee, S. Das. "An Efficient Multi-Objective QoS-Routing Algorithm for Wireless Multicasting". IEEE INFOCOM 2002.

[ROY04] A. Roy, S. Das. "QM2RP: A QoS-based Mobile Multicast Routing Protocol using Multi-Objective Genetic Algorithm". The Journal of Mobile Communication, Computation and Information, Wireless Networks. 2004.

[SAI00] H. Saito, Y. Miyao, M. Yoshida. "Traffic engineering using multiple multipoint-to-point LSPs". INFOCOM 2000.

[SEO02] Y. Seok, Y. Lee, Y. Choi, C. Kim. "Explicit Multicast Routing Algorithms for Constrained Traffic Engineering". IEEE ISCC 2002.

[SOL04] F. Solano, R. Fabregat, Y. Donoso. "Sub-flow assignment model of multicast flows using multiple p2mp LSPs". 30th Conferencia Latinoamericana de Informática (CLEI2004). September 2004.

[SOL04a] F. Solano, R. Fabregat, Y. Donoso, JL. Marzo. "Mapping subflows to P2MP LSPs". IEEE International Workshop on IP Operations & Management (IPOM 2004). Beijing, China. October 2004.

[SOL05] F. Solano, R. Fabregat, Y. Donoso, JL. Marzo. "Asymmetric Tunnels in P2MP LSPs as a Label Space Reduction Method". IEEE ICC 2005. Seoul, Korea. May 2005.

[SON03] J. Song, S. Kim, M. Lee, H. Lee, T. Suda. "Adaptive Load Distribution over Multipath in MPLS Networks". IEEE ICC 2003.

[SRI03] A. Sridharan, R. Guerin, C. Diot. “Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks”. IEEE INFOCOM 2003.

[STR02] A. Striegel, G. Manimaran. “A survey of QoS Multicasting Issues”. IEEE Communications Magazine - June 2002.

[SUN99] Q. Sun. “A genetic algorithm for delay-constrained minimum-cost multicasting”. Technical Report, IBR, TU Braunschweig, Butenweg, 74/75, 38106, 1999.

[THA00] D. Thaler. “Border Gateway Multicast Protocol (BGMP): Protocol Specification”. draft-ietf-bgmp-spec-02.txt. Noviembre 2000.

[TRA03] H. Tran, R. Harris. “Genetic algorithm approach to rearrangement of QoS multicast tree”. 3rd ATcrc Telecommunications & Networking Conference and Workshop, 11-12 Dec. 2003.

[VAN99] D. Van Veldhuizen. “Multiobjective Evolutionary Algorithms: Classifications, Analysis and New Innovations”. Ph.D. Thesis. Air Force Institute of Technology. 1999.

[VIL99] C. Villamizar. “MPLS Optimized Multipath (MPLS-OMP)”. Internet Draft. 1999.

[VUT00] S. Vutukury, J. Garcia. “A traffic engineering approach based on minimum-delay routing”. IEEE ICCN 2000.

[XIA99] F. Xiang, L. Junzhou, W. Jieyi, G. Guanqun. “QoS routing based on genetic algorithm”. Computer Communications 22. pp. 1293-1399. May 1999.

[WAI88] D. Waitzman. “Distance Vector Multicast Routing Protocol”. RFC 1075. 1988.

[WAN01] Z. Wang. “Internet QoS. Arquitecture and Mechanism for Quality of Service”. Morgan Kaufmann Publishers. 2001.

[WAN01a] Y. Wang, Z. Wang, L. Zhang. “Internet Traffic Engineering without Full Mesh Overlaying”. INFOCOM 2001.

[ZIT99] E. Zitzler, L. Thiele. "Multiobjective Evolutionary Algorithm: A comparative case study and the Strength Pareto Approach". IEEE Transaction Evolutionary Computation. Vol.3. No.4. 1999.

Appendix A Multi-Objective Optimization in NonConvex Solution Spaces

A. 1. Overview

Instead of using a weighted sum of the objectives, other means of combining multiple objectives into a single objective can also be used. For this purpose, weighted metrics such as L_p and L_∞ distance metrics are ofen used. For non-negative weights, the weighted L_p distance measure of any solution *x* from the ideal solution z^* can be minimized as follows:

$$
\begin{aligned}
&\textit{Minimize} \quad L_p(x) = \left(\sum_{m=1}^{M} w_m \, | f_m(x) - z_m^* |^p \right)^{1/p}, \\
&SA \quad g_j(x) \geq 0, \qquad j = 1,2,\ldots,J \\
&\quad h_k(x) = 0, \qquad k = 1,2,\ldots,K
\end{aligned}
\tag{A.1}
$$

The parameter *p* can take any value between 1 and ∞. When using $p = 1$, the resulting problem is equivalent to the weighted sum approach. When using $p = 2$, a weighted Euclidean distance of any point in the objective space of the ideal point is minimized. When a large *p* is used, the above problem is reduced to a problem of minimizing the largest deviation $| f_m(x) - z_m^* |^p$. This problem has a special name – a *weighted Tchebycheff* problem:

$$
\begin{aligned}
&\textit{Minimize} \quad L_\infty(x) = \max_{m=1}^{M} w_m \, | f_m(x) - z_m^* |, \\
&SA \quad g_j(x) \geq 0, \qquad j = 1,2,\ldots,J \\
&\quad h_k(x) = 0, \qquad k = 1,2,\ldots,K
\end{aligned}
\tag{A.2}
$$

A. 2. Optimization Scheme

In this case using L_∞ the mathematical model MHDB-S can be written as follows:

Minimize

$$MAX\left(r_1\left|\alpha - z_1^*\right|, r_2\left|\sum_{f\in F}\sum_{t\in T_f}\sum_{(i,j)\in E} Y_{ij}^{tf} - z_2^*\right|, r_3\left|\sum_{f\in F}\sum_{(i,j)\in E} bw_f \max_{t\in T_f}\left(X_{ij}^{tf}\right) - z_3^*\right|, r_4\left|\sum_{f\in F}\sum_{t\in T_f}\sum_{(i,j)\in E} v_{ij} Y_{ij}^{tf} - z_4^*\right|\right)$$

(MHDB-S model) (A.3)

Subject to

$$\alpha = \max\{\alpha_{ij}\}, \text{ where } \alpha_{ij} = \frac{\sum_{f\in F} bw_f . \max_{t\in T_f}(X_{ij}^{tf})}{c_{ij}} \qquad \text{(A.4)}$$

$$\sum_{(i,j)\in E} X_{ij}^{tf} - \sum_{(j,i)\in E} X_{ji}^{tf} = 1 \quad , t\in T_f, f\in F, i = s \qquad \text{(A.5)}$$

$$\sum_{(i,j)\in E} X_{ij}^{tf} - \sum_{(j,i)\in E} X_{ji}^{tf} = -1 \quad , i,t\in T_f, f\in F \qquad \text{(A.6)}$$

$$\sum_{(i,j)\in E} X_{ij}^{tf} - \sum_{(j,i)\in E} X_{ji}^{tf} = 0 \quad , t\in T_f, f\in F, i\neq s, i\notin T_f \qquad \text{(A.7)}$$

$$\sum_{f\in F} bw_f . \max_{t\in T_f}(X_{ij}^{tf}) \leq c_{ij} \quad ,(i,j)\in E \qquad \text{(A.8)}$$

$$\sum_{j\in N}\sum_{t\in T_f} Y_{ij}^{tf} \leq \left\lceil \frac{bw_f}{\left\lfloor \sum_{j\in N} c_{ij} \Big/ \sum_{j\in N} connection_{ij} \right\rfloor} \right\rceil , \qquad \text{(A.9)}$$

$i\in N, f\in F$

where

$$X_{ij}^{tf} \in \Re, 0 \leq X_{ij}^{tf} \leq 1 \qquad \text{(A.10)}$$

$$Y_{ij}^{tf} = \left\lceil X_{ij}^{tf} \right\rceil = \begin{cases} 0 & , X_{ij}^{tf} = 0 \\ 1 & , 0 < X_{ij}^{tf} \leq 1 \end{cases} \qquad \text{(A.11)}$$

$$\sum_{i=1}^{m} r_i = 1, \quad r_i \in \Re, \quad r_i \geq 0, m > 0 \qquad \text{(A.12)}$$

In this method it is clear that when p = 1 or 2, not all Pareto-optimal solutions can be obtained. However, when the weighted Tchebycheff metric is used, any Pareto-optimal solution can be found in non-convex objective space.

Since different objectives can take values of different orders of magnitude, it is advisable to normalize the objective functions. This means that we need to know the minimum and maximum function values of each objective. Moreover, this method also requires the ideal solution z^*. Therefore, all M objectives need to be independently optimized before optimizing the L_p metric.

A. 3. Conclusions and Motivations

Although by using this method we can find the solutions of the Optimum Pareto Front in convex and non-convex solution spaces, the problem is presented in the following aspects:

- It is necessary to know the Z* point which is the reference point for finding the Pareto Front solutions. It can be very difficult to find this point.

- Each execution of this method gives a point of the Pareto Front as a result. Due to this, it is necessary to perform this model many times in order to find different points of the Pareto Front.

- This mathematical model continues to have the problem of ignoring the number of sub flows and their transmission percentage.

Due to the aspects mentioned above, in chapter 6, a mathematical model was presented, adding a sub index k in order to be able to control how many and which sub flows of a particular flow in the Multicast transmission were used. Moreover, an algorithm scheme was presented to give solutions to the problems found, using mono-objective solution methods presented in chapters 3 and 4.

Appendix B Framework Considerations

In this appendix, we present the theoretical concepts, such as mathematical notation for the networks representation, distance labels, layered networks, mathematical programming formulation for the SPT primal and dual problem and the different lemmas of the primal-dual relationship theory.

The network is modelled as a directed graph $G=(N,E)$, where N is the set of nodes and E is the set of links. Let $s \in N$ be the ingress node. Let $t \in T$ be any egress node, where T is the set of egress nodes. Let $(i,j) \in E$ be the link from node i to node j, and let w_{ij} be the weight associated with each link (i,j).

Let f be any multicast flow $f \in F$, where F is the flow set and T_f is the egress nodes subset of the multicast flow f. $T=\bigcup_{f \in F} T_f$. Let X_{ij}^{tf} be the fraction of flow f to egress node t assigned to link (i,j).

B. 1. SPT Primal Problem for Multicast Transmission

The Shortest Path Tree Primal problem (P-SPT) can be formulated as

$$\min \sum_{f \in F} \sum_{t \in T_f} \sum_{(i,j) \in E} w_{ij} . X_{ij}^{tf} \tag{B.1}$$

Subject to

$$\sum_{(i,j) \in E} X_{ij}^{tf} - \sum_{(j,i) \in E} X_{ji}^{tf} = 1 \quad , t \in T_f, f \in F, i = s \tag{B.2}$$

$$\sum_{(i,j) \in E} X_{ij}^{tf} - \sum_{(j,i) \in E} X_{ji}^{tf} = -1 \quad , i, t \in T_f, f \in F \tag{B.3}$$

$$\sum_{(i,j) \in E} X_{ij}^{tf} - \sum_{(j,i) \in E} X_{ji}^{tf} = 0 \quad , t \in T_f, f \in F, i \neq s, i \notin T_f \tag{B.4}$$

$$X_{ij}^{tf} \in \Re, 0 \leq X_{ij}^{tf} \leq 1 \tag{B.5}$$

The objective function (B.1) consists of minimizing the total weight of links w_{ij} used. Constraint (B.2) denotes that the total flow emerging from the ingress node to an egress node t must be 1. Constraint (B.3) denotes that the total flow coming to an egress node t must be -1.

Constraint (B.4) denotes that for any intermediate node which is different from the ingress node, (i $\neq$ s), and from the egress nodes, $(i \notin T)$, the sum of their output flows to the egress node t minus the input flows with a destination at the egress node t must be 0. Constraint (B.5) shows that the X_{ij}^{tf} variable may have a value of between 0 and 1. Solving the problem without load balancing implies that this variable is only able to take the values 0 and 1, which show respectively, whether the *(i,j)* link is or is not being used to carry information to the egress node *t*.

The solution using the X_{ij}^{tf} variables provides optimum flow values. These variables form the multiple trees that transmit a multicast flow *f* from the ingress node to the set of egress nodes T_f.

B. 2. Distance labels, layered networks and SPT Optimality Conditions

At this point, we will apply the concept of distance labels to the P-SPT problem.

A distance function $d : N \rightarrow Z^{+} \bigcup \{0\}$ with respect to a graph is a function from the set of nodes *N* to the set of nonnegative integers. A distance function is valid with respect to a flow *f* if it satisfies the following two conditions:

Lemma 1*. d(s) = 0 and* $d(j) \leq d(i) + 1$ for every link *(i,j)* in graph *G(N,E)*. *d(i)* is referred to as the distance label of node *i* [AHU93], [AHU97], [BAZ90] and [BAZ93].

Lemma 2*.* If the conditions of the distance labels are valid, the distance label *d(j)* is a lower bound on the length of the shortest (directed) path from node *j* to node *s* [AHU93], [AHU97], [BAZ90] and [BAZ93].

Now, with respect to a given flow *f*, we can define the layered network *G'(N,E)* by determining the exact distance labels *d* in *G(N,E)*. The layered network consists of the links *(i,j)* in *G(N,E)* satisfying the condition *d(j) = d(i) + 1*.

Let d_j^{tf} for $j \neq s$ denote the length of a SPT from the ingress node to the egress nodes T_f. Thus, in keeping with lemma 1 and lemma 2, $d_s^{tf} = 0, t \in T_f, f \in F$. If the distance labels are SPT distances associated to the weights w_{ij} for each link (i,j), they must satisfy the following optimality conditions:

Lemma 3 (SPT Optimality Conditions). Let λ_f be a tree from s to t, $t \in T_f$, associated to the flow f, which consists of several paths P_k^{tf} with $1 \leq k \leq h$, where h is the number of paths. For every node $j \in N$ in the tree, let d_j^{tf} denote the length of a directed path in the path P_k^{tf} from the ingress node to node j with a destination at the egress node t in the flow f. d_j^{tf} represents the SPT distances for all egress nodes to flow f if and only if they satisfy the following SPT optimality conditions:

$$d_j^{tf} \leq d_i^{tf} + w_{ij}, (i,j) \in E, t \in T_f, f \in F$$

$$d_s^{tf} = 0, t \in T_f, f \in F$$

B. 3. Mathematical programming duality theory

Each mathematical programming problem (P-Primal) is associated with another mathematical programming problem (D-Dual). P and D are mathematically equivalent in the sense that they have the same optimal objective value, and the optimal solution of D can be derived from the optimal solution of P and, vice versa, due to the relationship of complementary slackness.

Lemma 4 (Weak duality). Let $z(x)$ denote the objective function value of a feasible solution, x, of the SPT problem and let $\beta(d)$ denote the objective function value of a feasible solution of its dual. Then $\beta(d) \leq z(x)$ [AHU93].

The weak duality lemma implies that the objective function value of any dual feasible solution is a lower bound of the objective function value of any primal feasible solution. One consequence of this result is immediate: if a dual solution d and a primal solution x have the same objective function value $\beta(d) = z(x)$, d must be an optimal solution of the dual problem and x must be an optimal solution of the primal problem.

Lemma 5 (Strong duality). For any choice of problem, the SPT problem always has a solution $\bar{x}$ and the dual SPT problem has a solution $\bar{d}$ satisfying the property that $\beta(\bar{d}) = z(\bar{x})$ [AHU93].

This lemma shows that any optimal solution, $\bar{x}$, of the SPT problem always has an associated dual solution, $\bar{d}$, satisfying the condition $\beta(\bar{d}) = z(\bar{x})$. The weak and strong duality lemmas imply several fundamental results concerning relationships between the primal and dual problems. The following complementary slackness optimality conditions, which are another way of relating the two problems, makes some of these relationships more explicit.

Lemma 6 (Complementary slackness optimality conditions) [AHU93]. If $\bar{x}$ is an optimal solution of the primal SPT problem and $\bar{d}$ is an optimal solution of the dual SPT problem, the pair $(\bar{x}, \bar{d})$ satisfies the complementary slackness optimality conditions:

If $X_{ij}^{tf} > 0$, *then* $d_j^{tf} - d_i^{tf} = w_{ij}$.

If P_k^{tf} is a path of the tree λ_f determined by $\overline{X}_{ij}^{tf}, (i,j) \in E, t \in T_f, f \in F$, for each link $(i,j) \in P_k^{tf}$, if $X_{ij}^{tf} > 0$, then this implies that $d_j^{tf} - d_i^{tf} = w_{ij}$.

B.4. Shortest Path Tree Dual Problem

Using the definition of the P-SPT problem and the concepts of the duality theory described previously, in this section we demonstrate the relationship between the primal and dual SPT problems.

The Shortest Path Tree Dual Problem (D-SPT) can be formulated as

$$\max \sum_{f \in F} \sum_{t \in T_f} d_t^{tf} \tag{B.6}$$

Subject to

$$d_j^{tf} - d_i^{tf} \le w_{ij}, (i,j) \in E, t \in T_f, f \in F \tag{B.7}$$

By lemma 3 $d_s^{tf} = 0, t \in T_f, f \in F$ (B.8)

$$w_{ij} \ge 0 \tag{B.9}$$

Let $\overline{X}_{ij}^{tf}$ be the optimal solution of (P-SPT). Then $\overline{X}_{ij}^{tf}$ determines the shortest path for each egress node in a flow. Let $\bar{d}_j^{tf}$ be the optimal solution of the dual problem (D-SPT). The values of $\bar{d}_j^{tf}$ can be viewed as the distance from the ingress node *s* to the node *j* based on the SPT to the flow *f* with a destination at egress node *t* determined in (P-SPT). In particular, $\bar{d}_t^{tf}, t \in T_f, f \in F$ is the total length of the SPT from *s* to *t*, $t \in T_f$. If d_j^{tf} is any solution of distance labels satisfying the constraints of this problem and the path P_k^{tf} defined as s, i_1, i_2, ..., i_m, t is any path from node s to node t, then

$d_{i_1}^{tf} - d_s^{tf} \le w_{si_1}$

$d_{i_2}^{tf} - d_{i_1}^{tf} \le w_{i_1 i_2}$

...

$d_t^{tf} - d_{i_m}^{tf} \le w_{i_m t}$

so, by adding these inequalities and using the fact that $d_s^{tf} = 0$, we see that

$d_t^{tf} \le w_{si_1} + w_{i_1 i_2} + ... + w_{i_m t}$

This result shows that if d_t^{tf} is any feasible solution to the optimization problem, then d_t^{tf} is a lower bound of the length of any path from node s to every node t of flow f and therefore, it is a lower bound of the shortest distance between these nodes and, as a consequence, d_t^{tf} equals the SPT distance for every flow f, $f \in F$. Now, the total distance of the tree, λ_f, where λ_f represents the tree given by the values , $\overline{X}_{ij}^{tf}$ for flow f, would be as follows:

$$\sum_{t \in T_f} d_t^{tf} = \sum_{t \in T_f} d_t^{tf} - d_s^{tf} = \sum_{t \in T_f} \sum_{j=1}^{l_{tf}} w_{i_{j-1}, i_j} \qquad \text{(B.10)}$$

where l_{tf} is the number of nodes in a path with a destination at egress node t in flow f, i.e., $l_{tf} = |P_k^{tf}|$, where | | is the number of elements in a set.

By applying the complementary slackness relationship of the linear programming duality theory, presented in lemma 6, to the primal (P-SPT) dual (D-SPT) pair, we have:

Theorem 1: Let λ_f be a tree from s to t, $t \in T_f$. If for every link $(i,j) \in \lambda_f$, $d_j^{tf} - d_i^{tf} = w_{ij}$, then λ_f is a SPT with respect to the link weights $\{w_{ij}\}$.

Proof: Let $\lambda_f = \{P_1^{tf}, P_2^{tf}, .., P_h^{tf}\}$, be the set of paths from the ingress node to each of the egress nodes t, for flow, f of the multicast tree. Each path $P_k^{tf} = p_0^k, p_1^k, ..., p_{l_{tf}-1}^k, p_{l_{tf}}^k$ is made up of all the nodes so that $p_0^k = s$ y $p_{l_{tf}}^k = t$, where $1 \le k \le h$, with h being the number of paths in the tree λ_f.

Thus we have $d^{tf}_{p^k_j} - d^{tf}_{p^k_{j-1}} = w_{p_{j-1},p_j}$

for $0 \le j \le l_{tf}$. If we sum up all these equations using equation (B.10) we get

$$\sum_{t \in T_f} d^{tf}_t = \sum_{t \in T_f} d^{tf}_t - d^{tf}_s = \sum_{t \in T_f} \sum_{j=1}^{l_{tf}} w_{p_{j-1},p_j} \tag{B.11}$$

Note that $\sum_{t \in T_f} \sum_{j=1}^{l_{tf}} w_{p_{j-1},p_j}$ is the length of the tree λ_f. Let $\lambda'_f = \{Q^{tf}_1, Q^{tf}_2, .., Q^{tf}_{r_f}\}$ be any other tree between s and λ'_f and every path $Q^{tf}_k = q^k_0, q^k_1, ..., q^k_{r_{tf}-1,} q^k_{r_{tf}}$ where r_{tf} is the number of nodes, that is to say, $r_{tf} = |Q^{tf}_k|$. Then, by constraint (B.7),

$d'^{tf}_j - d'^{tf}_{j-1} \le w_{q_{j-1},q_j}$

for $0 \le j \le r_{tf}$. Similarly, if we sum up all these equations using equation (B.10), we get

$$\sum_{t \in T_f} d^{tf}_t \le \sum_{t \in T_f} \sum_{j=1}^{r_{tf}} w_{q_{j-1},q_j} \tag{B.12}$$

Hence equations (B.11) and (B.12) imply that $\sum_{t \in T_f} \sum_{j=1}^{l_{tf}} w_{p_{j-1},p_j} \le \sum_{t \in T_f} \sum_{j=1}^{r_{tf}} w_{q_{j-1},q_j}$. That is, the length of the tree λ'_f is at least as long as the length of the tree λ_f. Therefore, λ_f is a shortest path tree. ■

In summary, lemma 6 and theorem 1 taken together, say that every tree determined by $\overline{X}^{tf}_{ij}$ is a shortest path tree.

B. 5. Optimal Multicast Routing

In Section 3 we demonstrated that the primal problem (P-SPT) could be converted via the dual problem (D-SPT) into a SPT problem by setting appropriate link weights. In this section, we propose the multi-objective load-balancing scheme (P-MHDB), which includes the maximum link utilization (α), the hop count (HC), the total bandwidth consumption (BC), and the total end-to-end delay (DL). Later, we saw that, using the corresponding dual problem, D-MHDB the solution obtained for the P-MHDB problem was the SPT.

B. 5.1. Multi-objective primal problem

Given the graph G=(N,E) defined in Section 3, we will now introduce some additional notation. Let c_{ij} be the capacity of each link *(i,j)*. Let bw_f be the traffic demand of a flow *f* from the ingress node *s* to T_f. The binary variable, Y_{ij}^{tf}, represents whether link *(i,j)* is used (1) or not (0) by the multicast tree rooted at the ingress node *s* and reaching egress node subset T_f. Let v_{ij} be the delay of link *(i,j)*. Let *m* be the number of variables in the multi-objective function. Let N_T be the maximum number of bifurcation paths for each node.

The problem of minimizing *n* multicast flows from source node *s* to the egress nodes of each subset T_f is formulated as follows:

min

$$r_1.\alpha + r_2 \sum_{f\in F} \sum_{t\in T_f} \sum_{(i,j)\in E} Y_{ij}^{tf} + r_3 \sum_{f\in F} \sum_{(i,j)\in E} bw_f \max_{t\in T_f}\left(X_{ij}^{tf}\right) + r_4 \sum_{f\in F} \sum_{t\in T_f} \sum_{(i,j)\in E} v_{ij} Y_{ij}^{tf} \qquad \text{(P-MHDB model)}$$

(B.13)

Subject to

$$\sum_{(i,j)\in E} X_{ij}^{tf} - \sum_{(j,i)\in E} X_{ji}^{tf} = 1 \quad ,t\in T_f, f\in F, i=s \tag{B.14}$$

$$\sum_{(i,j)\in E} X_{ij}^{tf} - \sum_{(j,i)\in E} X_{ji}^{tf} = -1 \quad ,i,t\in T_f, f\in F, i\in T_f \tag{B.15}$$

$$\sum_{(i,j)\in E} X_{ij}^{tf} - \sum_{(j,i)\in E} X_{ji}^{tf} = 0 \quad ,t\in T_f, f\in F, i\neq s, i\notin T_f \tag{B.16}$$

$$\sum_{f\in F} bw_f.\max_{t\in T_f}(X_{ij}^{tf}) \le c_{ij}.\alpha \quad ,\alpha\ge 0,(i,j)\in E \tag{B.17}$$

where

$$\sum_{j\in N} \sum_{t\in T_f} Y_{ij}^{tf} \le \left\lceil \frac{bw_f}{\left\lfloor \sum_{j\in N} c_{ij} \Big/ \sum_{j\in N} connection_{ij} \right\rfloor} \right\rceil \quad ,i\in N, f\in F \tag{B.18}$$

$$X_{ij}^{tf} \in \Re, 0 \le X_{ij}^{tf} \le 1 \tag{B.19}$$

$$Y_{ij}^{tf} = \left\lceil X_{ij}^{tf} \right\rceil = \begin{cases} 0 & ,X_{ij}^{tf} = 0 \\ 1 & ,0 < X_{ij}^{tf} \le 1 \end{cases} \tag{B.20}$$

$$\sum_{i=1}^{m} r_i = 1, \quad r_i \in \Re, \quad r_i \ge 0, m > 0 \tag{B.21}$$

The Multi-objective function (P- MHDB model) (B.13) is the same as that presented in chapter 3.

B. 5.2. Dual multi-objective problem

In this section, we present the dual problem of the primal problem P-MHDB proposed earlier and demonstrate that we can find a tree of a set of trees that are the SPT (Shortest Path Tree). In the primal problem, we assigned the delay v_{ij} and the capacity c_{ij} for each link *(i,j)* (see Fig. B.1). Before we present the theoretical results, let us first illustrate, using a simple example with only one egress node, the relationship between the primal and dual problem.

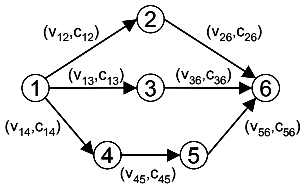


Fig B.1 Metrics for the primal problem

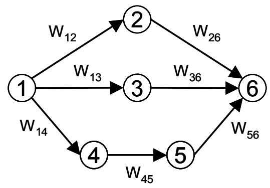


Fig B.2 Metrics for the dual problem

In the dual problem, we define an associated weight w_{ij} to the link *(i,j)*. In this case, the objective consists in maximizing the distance d_t^{tf} from the ingress node to each egress node.

The value found when solving the objective function indicates a lower bound for the distances associated with the egress nodes and therefore, we can find the SPT. If multiple shortest path trees exist for a particular flow, these trees have an equal cost. When we refer to a tree, we mean any one of these trees.

The dual of P-MHDB can be formulated as.

$$\max \sum_{f \in F} \sum_{t \in T_f} d_t^{tf} \quad \text{(D-MHDB model)} \tag{B.22}$$

Subject to

$$d_j^{tf} - d_i^{tf} \le bw_f . w'_{ij} + r_2 + r_3 . bw_f + r_4 . v_{ij}, \quad (i,j) \in E, t \in T_f, f \in F \tag{B.23}$$

$$\sum_{(i,j) \in E} c_{ij} w'_{ij} = r_1 \tag{B.24}$$

$$w'_{ij} \geq 0, \quad (i,j) \in E, t \in T, f \in F, r_1, r_2, r_3, r_4 > 0, \\ bw_f > 0 \tag{B.25}$$

By lemma 3 $d_s^{tf} = 0, t \in T_f, f \in F$ (B.26)

Note that the shortest path tree dual problem (D-SPT) is a special case of the D-MHDB model. By applying the same proof methodology used for the P-SPT, we can to demonstrate that the P-MHDB is an SPT.

Lemma 7: (complementary slackness) By applying lemma 6: If $\overline{X}_{ij}^{tf} > 0$, then $\overline{d}_j^{tf} - \overline{d}_i^{tf} = bw_f.w'_{ij} + r_2 + r_3.bw_f + r_4.v_{ij}$.

In the same way as the D-SPT problem, if we assume that the variables, $\overline{w}'_{ij}$, are constant, then the value of $\overline{d}_i^{tf}$ is the shortest path distance from ingress node s to node *i* with respect to link weight, $bw_f.w'_{ij} + r_2 + r_3.bw_f + r_4.v_{ij}$. Therefore, the total length of the shortest path tree for flow *f* is given by $\sum_{t \in T_f} d_t^{tf}$. Let $\overline{w}_{ij} = bw_f.w'_{ij} + r_2 + r_3.bw_f + r_4.v_{ij}$, hence $\overline{w}_{ij} > 0$.

Theorem 2: Let λ_f be any tree from ingress node *s* to *t*, $t \in T_f$ determined by the P-MHDB optimal solution $\overline{X}_{ij}^{tf}$. Then λ_f is a shortest path tree with respect to link weights $\{\overline{w}_{ij}\}$.

Proof: Let λ_f be a tree for flow *f* determined by the P-MHDB optimal solution. For every link *(i,j)* on λ_f, we have $\overline{X}_{ij}^{tf} > 0$. This implies using lemma 7 that $\overline{d}_j^{tf} - \overline{d}_i^{tf} = \overline{w}_{ij}$. From theorem 1, λ_f is a shortest path tree from *s* to *t*, $t \in T_f$ with respect to link weights $\{\overline{w}_{ij}\}$. ■

In summary, we have shown that for the multi-objective function, the optimal trees can be reproduced as shortest path trees based on certain positive link weights, which can be derived through the optimal dual solution. This characteristic is based on constraints (B.14) and (B.15) and on the multi-objective function (B.13) of the primal problem formulation, which together produce constraint (B.23) of the dual problem which indicates the distance between a pair of adjacent nodes.

Appendix C Publications

Journals

- [DON05] Y. Donoso, R. Fabregat, JL. Marzo. "Multicast Routing With Traffic Engineering: A Multi-Objective Optimization Scheme And A Polynomial Shortest Path Tree Algorithm With Load Balancing." Annals of Operations Research. Kluwer Publisher. 2005.

- [DON04c] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Algorithm for Multicast Routing with Traffic Engineering." Telecommunication Systems Journal. Kluwer Publisher. 2004.

International Conferences

- [DON05a] Y. Donoso, R. Fabregat, F. Solano, JL. Marzo, B. Baran. "Generalized Multiobjective Multitree model for Dynamic Multicast Groups". IEEE ICC. Seoul. Korea. May 2005.

- [SOL05] F. Solano, R. Fabregat, Y. Donoso, JL. Marzo. "Asymmetric Tunnels in P2MP LSPs as a Label Space Reduction Method". IEEE ICC 2005. Seoul, Korea. May 2005.

- [SOL04a] F. Solano, R. Fabregat, Y. Donoso, JL. Marzo. "Mapping subflows to p2mp LSPs". IEEE International Workshop on IP Operations & Management (IPOM 2004). Beijing, China. October 2004.

- [FAB04a] R. Fabregat, Y. Donoso, F. Solano, JL. Marzo. "Multitree Routing for Multicast Flows: A Genetic Algorithm Approach". Setè Congrés Català d'Intel·ligència Artificial (CCIA'2004), Universitat Autònoma de Barcelona, Barcelona (Spain). October 2004.

- [SOL04] F. Solano, R. Fabregat, Y. Donoso. "Subflow assignment model of multicast flows using multiple p2mp LSPs". CLEI (Congreso Latinoamericano de Estudios en Informática) 2004. Arequipa, Perú. October 2004. Selected as Best Papers.

- [FAB04a] Y. Donoso, R. Fabregat, JL. Marzo, A. Ariza. "A Multi-Objective Multipath Routing Algorithm for Multicast Flows." International Symposium on Performance

Evaluation of Computer and Telecommunication Systems (SPECTS'04). San Jose, USA. July 2004.

- [DON04a] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Model and Heuristic Algorithm for Dynamic Multicast Routing". IEEE & VDE Networks 2004. Vienna, Austria. June 2004.

- [DON04b] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Scheme for Dynamic Multicast Groups". IEEE ISCC (The 9th IEEE Symposium on Computers and Communications). Alexandria, Egypt. June 2004.

- [DON04] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Algorithm for Multicast Routing with Traffic Engineering". IEEE 3rd International Conference on Networking ICN'04, Guadeloupe, French Caribbean, March 2004. Selected as Best Papers.

- [DON04d] Y. Donoso, R. Fabregat, JL. Marzo. "Multi-Objective Optimization Model and Heuristic Algorithm for Multipath Routing of Static and Dynamic Multicast Group." III Workshop on MPLS Networks. Girona, Spain. March 2004.

- [DON04e] Y. Donoso, R. Fabregat, JL. Marzo. "Multicast Routing with Traffic Engineering: a Multi-Objective Optimization Scheme and a Polynomial Shortest Path Tree Algorithm with Load Balancing". Proceedings of CCIO -2004 Cartagena de Indias, Colombia, March 2004. Operation Research National Award given by the Colombian Operations Research Society. March 2004.

- [DON03] Y. Donoso, R. Fabregat. "Multi-Objective Scheme over Multi-Tree Routing in Multicast MPLS Networks". IFIP/ACM Latin America Networking Conference 2003 (LANC03), IFIP-TC6 and ACM SIGCOMM, La Paz (Bolivia). October 2003.

- [DON03a] Y. Donoso, R. Fabregat. "Ingeniería de Tráfico aplicada a LSPs Punto-Multipunto en Redes MPLS". CLEI (Congreso Latinoamericano de Estudios en Informática) 2003. La Paz, Bolivia. October 2003.

Research Reports

- [BAR04] B. Barán, R. Fabregat, Y. Donoso, F. Solano, JL. Marzo. "Generalized Multiobjective Multitree model". Proceedings of IIiA Research Report, ref. IIiA 04-08-RR, Institut d'Informàtica i Aplicacions, University of Girona. September 2004.