Multi-objective optimization scheme for multicast flows: a survey, a model and a MOEA solution

Ramon Fabregat
IIiA University of Girona
EPS – P4 - Campus Montilivi
Girona 17071 (Spain)
+ 34 972 41 84 84
ramon@eia.udg.es

Yezid Donoso
Universidad del Norte
Km.5 Vía Puerto Colombia
Barranquilla (Colombia)
+ 57 5 3509550
ydonoso@uninorte.edu.co

Benjamín Baran
Centro Nacional de Computación
National University of Asuncion
San Lorenzo (Paraguay)
+ 595 21 585550
bbaran@cnc.una.py

Fernando Solano
IIiA University of Girona
EPS – P4 - Campus Montilivi
Girona 17071 (Spain)
+ 34 972 41 84 75
fernando@eia.udg.es

Jose L. Marzo
IIiA University of Girona
EPS – P4 - Campus Montilivi
Girona 17071 (Spain)
+ 34 972 41 84 96
marzo@eia.udg.es

ABSTRACT
This paper presents a new traffic engineering load balancing taxonomy, classifying several publications and including their objective functions, constraints and proposed heuristics. Using this classification, a novel *Generalized Multiobjective Multitree model* (GMM-model) is proposed. This model considers for the first time multitree-multicast load balancing with splitting in a multiobjective context, whose mathematical solution is a whole Pareto optimal set that can include several results than it has been possible to find in the publications surveyed. To solve the GMM-model, a multi-objective evolutionary algorithm (MOEA) inspired by the Strength Pareto Evolutionary Algorithm (SPEA) is proposed. Experimental results considering up to 11 different objectives are presented for the well-known NSF network, with two simultaneous data flows.

Categories and Subject Descriptors
C.2.1 [**Network Architecture and Design**]: Network communications. C.2.2 [**Network Protocols**]: Routing protocols. C.2.3 [**Network Operations]**: Network management.

General Terms
Management, Design.

Keywords
multicast, splitting, traffic engineering, load balancing, multiobjective.

1. INTRODUCTION
Traffic engineering (TE) is concerned with improving the performance of operational networks, usually taking into account QoS (Quality of Service) requirements. The main objectives are to reduce congestion hot spots, improve resource utilization and provide adequate QoS for final users. These aims 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, giving the best possible service, i.e., minimum delay, packet losses, jitter, etc.

When load balancing techniques are translated into a mathematical formulation, a heuristic or a practical implementation, different conflicting objectives are found and hence they have been considered in the literature (more details are presented in Table 2) as minimizing:

- maximum or average link utilization [1-14];
- maximum, average and / or total hop count [1, 11-16];
- maximum, average and / or total delay [1, 8, 9, 11-14, 16-28];
- bandwidth consumption [3, 4, 7, 11-15, 17, 19, 23-26, 28, 29];
- flow assignation [4, 24];
- packet loss [17, 25, 26, 28];
- queue size [22];
- number of Label Switching Paths (LSPs) in a Multi-Protocol Label Switching (MPLS) implementation [16];
- jitter [17] and
- different cost functions [8-10, 18, 27, 28, 30-34].

Clearly, when all these objectives are considered, it can be seen that the problem is multiobjective, as already recognized by several authors [8, 9, 11, 18, 24-27]. It should be noted that before this multiobjective context was clearly recognized, several authors treated some objectives as restrictions with arbitrary a priori constraints, typically expressed as hard upper bounds on objectives such as the delay, hop count or bandwidth consumption, to name a few. Moreover, since single objective techniques such as linear programming, shortest path or genetic algorithms (GA) have been widely used, most influential works in this field (see Table 2) have studied this Multi-Objective Problem (MOP) as a Single-Objective Problem (SOP) with a *cost function* that combines several objectives using, as an example, a weighted sum [1, 4, 7, 12-14, 17, 23, 28]. It should be emphasized that an optimization problem is said to be a SOP if it optimizes a unique cost function, even though this cost function is a (typical linear) combination of several different objective functions, with several possible restrictions. It should be noted that in a *pure multi-objective context*, no objective needs to be considered as more important than the others, no a priori weighting of the objectives is needed and no a priori constraint on any objective is necessary; therefore, the solution of a MOP is usually a whole set of optimal compromised solutions, known as a Pareto set [35].

One interesting solution to the balancing alternative is the multipath approach, in which data is transmitted through different paths to achieve an aggregated, end-to-end bandwidth requirement. Several advantages of using multipath routing are discussed in [21]. 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. Furthermore, some authors have expanded this idea by proposing to split each flow into multiple subflows in order to achieve better load balancing [12, 13, 16, 36].

For a load balancing model to be general, unicast considerations are not enough and multicast should also be considered, as already proposed in [7-14, 17, 23-28, 30-34].

In all the different alternatives summarized when studying TE load balancing, a taxonomy is needed to clarify and better understand the problem; therefore, the next section proposes a *TE load balancing taxonomy*, which classifies previous publications., A *Generalized Multiobjective Multitree model* (GMM-model) inspired by this taxonomy is presented in section 3, mathematically formalizing the TE load balancing problem as a MOP. A *Multi-Objective Evolutionary Algorithm* (MOEA) which is able to solve the proposed GMM-model is presented in section IV, given its recognized ability for solving MOPs [8, 9, 35, 37]. Moreover, GMM-model considers a multitree-multicast load balancing problem with splitting in a multiobjective context, for the first time, using an evolutionary approach. To illustrate the resolution of the GMM-model using a MOEA, section 5 presents experimental results. Section 6 comments on the advantages of the GMM-model in comparison with previously published models. The final conclusions and future work are left for section 7.

2. TAXONOMY OF RELATED WORK

To categorize the large amount of related work, we propose a novel taxonomy. Table 1 presents the proposed *TE Load Balancing Taxonomy*. First, it considers the flow type, 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 destinationscan 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 considering 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. Therefore, Table 2 indicates the column *Splitting* as *Not Applicable* (NA) with reference to unipath or unitree. Finally, optimization problems are also classified as a SOP or MOP considering 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 [35].

Even though, the taxonomy in Table 1 is very useful for presenting our model in the next section, it is not complete. For a future work we plan to consider other characteristics such as the existence (or not) of a reserved backup path/tree, and the ability to support dynamic membership changes, among others.

Table 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 may travel through **different** paths / trees
Splitting	No	A flow always travel the **same** path/tree
	Yes	A flow may be split in **several** subflows that may be delivered through different paths/trees
Objective Problem	Single-Objective Problem (SOP)	Only one generic cost function is considered
	Multiple-Objective Problem (MOP)	Several (conflicting) objective functions may be simultaneously optimized in a multiobjective context

Using the above Taxonomy, Table 2 classifies what could be considered as the most inspired papers for this work. Note that [17] is repeated because it considers unicast and multicast flows. The last column of Table 2 shows the methodology/heuristic proposed in publication. Clearly, most publications consider the load balancing problem as a SOP, therefore, they have mainly propose the use of 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 consider 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 (still no splitting). Later, several papers considering splitting began to appear in the year 2000. Lately, the multiobjective context of the TE problem has been slowly recognized [24], with an increased number of publications since

2003 (see Table 2 in bold for multiobjective publications). Now, we are going to speak about some related works.

In [19] 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.

In [1] 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.

In [2] 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.

In [3] 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).

In [15] 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.

In [20] 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.

In [21] 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.

In [4] 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.

In [5] 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.

In [6] 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.

In [29] 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.

In [22] 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.

In [16] 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.

In [7] 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).

In [24] 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.

In [25] 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.

Table 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	
[17]*	1999			X	X		X			X				X	X	Unicast	UP	NA	SOP	Genetic algorithms, Weighted Sum
[18]	2004			X							X								MOP	MOEA
[19]	1998			X	X												MP	No	SOP	Ford-Fulkason method
[1]	1998	X	X	X										X						Fuzzy logic, Weighted Sum
[2]	2002	X												X						Lineal programming and shortest path
[3]	2003	X			X									X						Lineal programming and shortest path
[15]	2003		X		X							X								(Lineal) multi-commodity network flow problem
[20]	2000			X										X				Yes	SOP	Non Lineal programming
[21]	2001			X										X						Max-flow and shortest path
[4]	2001	X			X	X								X						Lineal programming, Weighted Sum
[5]	2002	X										X		X						Mixed-integer programming
[6]	2002	X																		(Lineal) multi-commodity network flow problem
[29]	2003				X									X	X					Max-flow and Shortest Path
[22]	2004			X				X						X						Scheduling algorithm
[36] [16]	2002 2004		X	X		X			X											Shortest Path
[17]*	1999			X	X		X			X				X	X	Multicast	UT	NA	SOP	Genetic algorithms, Weighted Sum
[30] [31] [32]	1998 1999 1999										X									Genetic algorithms
[33]	1999										X		X							Genetic algorithms
[23]	2001			X	X															Genetic algorithms, Weighted Sum
[7]	2002	X			X							X		X	X					Mixed-integer programming, Weighted Sum
[24]	2002			X	X	X													MOP	MOEA based on NSGA
[25]	2003			X	X		X						X							MOEA
[26]	2004			X	X		X													MOEA based on NPGA
[27]	2004			X							X									MOEA based on SPEA
[8] [9]	2004	X		X							X									MOEA based on SPEA
[34]	2003										X						MT	No	SOP	Shortest Path Tree
[28]	2004			X	X		X			X	X									Genetic algorithms, Weighted Sum
[10]	2004										X									DIMRO Heuristic
[11]	2004	X	X	X	X														MOP	MOEA based on NSGA
[12] [13] [14]	2004 2003 2004	X	X	X	X									X	X			Yes	SOP	Non lineal programming, max-flow and shortest path tree, Weighted Sum
Present work		X	X	X	X	X		X	X	X				X	X				MOP	MOEA based on SPEA

UP: Unipath
MP: Multipath
UT: Unitree
MT: Multitree
NA: Not Applicable
SOP: Single-Objective Problem
MOP: Multiple-Objective Problem

3. GENERALIZED MULTIOBJECTIVE MULTITREE MODEL: MATHEMATICAL FRAMEWORK

Inspired by the above taxonomy, the present work proposes a *Generalized Multiobjective Multitree model* (GMM-model) in a pure multiobjective context that considers simultaneously for the first time, multicast flow, multitree, and splitting, as shown in the last row of Table 2.

In the studied multiobjective context, a MOP considers a set of n decision variables, q objective functions and m restrictions that can be expressed as [35]:

$$\textit{Optimize} \quad \mathbf{y} = \phi(\mathbf{x}) = [\phi_1(\mathbf{x}), \ldots, \phi_q(\mathbf{x})] \qquad (1)$$

subject to $e(\mathbf{x}) = [e_1(\mathbf{x}), \ldots, e_m(\mathbf{x})] \geq 0$

where $\mathbf{x} = [x_1, \ldots, x_n] \in \mathbf{X}$ is the decision vector and $\mathbf{y} = [y_1, \ldots, y_q] \in \mathbf{Y}$ the objective vector. $\mathbf{X}$ denotes the decision space while $\mathbf{Y} = \phi(\mathbf{X})$ the corresponding objective space. The set of restrictions $e(\mathbf{x}) \geq 0$ determines the set of feasible solutions $\mathbf{X_f} \subseteq \mathbf{X}$ and its corresponding set of objective vectors $\mathbf{Y_f} \subseteq \mathbf{Y}$. In general, there is no single "*best*" solution but a set of optimal solutions; therefore, a different optimality concept of should be established.

In this multiobjective context, a solution $u \in \mathbf{X}$ is better than another $v \in \mathbf{X}$ if and only if u is as good as v in every objective and strictly better in at least one objective. In this case, u dominates v, which is denoted as $u \succ v$. On the other hand, $u \sim v$ indicates that u and v are indifferent, i.e. neither u dominates v, nor v dominates u; therefore, it is not possible to decide which one is better when all q objectives are considered at the same time.

A decision vector $\mathbf{x}^* \in \mathbf{X_f}$ is Pareto optimal if it is not dominated by any other decision vector of $\mathbf{X_f}$. The set of all Pareto optimal solutions is known as a *Pareto set* (denoted as $\mathbf{X^*}$) and its corresponding set of objective vectors is known as a *Pareto Front*, denoted as $\mathbf{Y^*}$.

The proposed 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. 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.

$$\sum_{k=1}^{|K_f|} f_k = 1 \qquad (2)$$

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. $0 \leq X_{ij}^{f_k t} \leq 1$. 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 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 [7, 13]. In particular, the novel introduction of a subflow-index k gives an easy way to identify subflows and define LSPs in a MPLS implementation.

To fully define most objective functions referenced in Table 2, 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 used (value 1) or not (value 0) for transporting subflow f_k to destination node t, i.e.

$$Y_{ij}^{f_k t} = \left\lceil X_{ij}^{f_k t} \right\rceil = \begin{cases} 0, & \text{if } X_{ij}^{f_k t} = 0 \\ 1, & \text{otherwise} \end{cases} \qquad (3)$$

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 (1), the proposed GMM-model considers the following objective functions can be considered (see Table 2 for a summary of objective functions (OF) already used in the literature):

Maximal link utilization

$$\phi_1 = \max\{\alpha_{ij}\} \qquad (4)$$

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

4

Hop count, in several different flavors such as:

Total hop count

$$\phi_2 = \sum_{(i,j)\in E} \sum_{f\in F} \sum_{k\in K_f} \sum_{t\in T_f} Y_{ij}^{f_k t} \qquad (5)$$

Hop count average

$$\phi_3 = \frac{\sum_{(i,j)\in E} \sum_{f\in F} \sum_{k\in K_f} \sum_{t\in T_f} Y_{ij}^{f_k t}}{\sum_{f\in F} \sum_{k=1}^{|K_f|} |T_f|} \qquad (6)$$

Maximal hop count, which is useful for QoS assurance

$$\phi_4 = \max_{\substack{\forall f\in F,\\ \forall k\in K_f,\\ \forall t\in T_f}} \left\{ \sum_{(i,j)\in E} Y_{ij}^{f_k t} \right\} \qquad (7)$$

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

$$\phi_9 = \max_{\substack{\forall f\in F\\ \forall t\in T_f}} \{ \mathrm{H}_{ft} \} \qquad (8)$$

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

$$\phi_6 = \sum_{(i,j) \in E} \sum_{f \in F} \sum_{k \in K_f} \sum_{t \in T_f} d_{ij} \cdot Y_{ij}^{f_k t} \quad (9)$$

Average delay

$$\phi_7 = \frac{\sum_{(i,j) \in E} \sum_{f \in F} \sum_{k \in K_f} \sum_{t \in T_f} d_{ij} \cdot X_{ij}^{f_k t}}{\sum_{f \in F} \sum_{k=1}^{|K_f|} |T_f|} \quad (10)$$

Maximal delay, which is useful for QoS assurance

$$\phi_8 = \max_{\substack{\forall f \in F \\ \forall k \in K_f \\ \forall t \in T_f}} \left\{ \sum_{(i,j) \in E} d_{ij} \cdot Y_{ij}^{f_k t} \right\} \quad (11)$$

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

$$\phi_9 = \max_{\substack{\forall f \in F \\ \forall t \in T_f}} \{ \Delta_{ft} \} \quad (12)$$

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

$$\phi_{10} = \sum_{(i,j) \in E} \sum_{f \in F} \sum_{k \in K_f} b_f \cdot \max_{t \in T_f} \left\{ X_{ij}^{f_k t} \right\} \quad (13)$$

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

$$\phi_{11} = \sum_{f=1}^{|F|} |K_f| \quad (14)$$

As stated in equation (1), a MOP formulation usually considers *m* constraints (C), such as the ones that follow.

Flow conservation constraints [12, 13]:

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

$$\sum_{j=1}^{|N|} \sum_{k=1}^{|K_f|} X_{s_f j}^{f_k t} = 1 \quad (15)$$

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

$$\sum_{i=1}^{|N|} \sum_{k=1}^{|K_f|} X_{it}^{f_k t} = 1 \quad (16)$$

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

$$\sum_{j=1}^{|N|} X_{i_f j}^{f_k t} - \sum_{i=1}^{|N|} X_{i i_f}^{f_k t} = 0, \quad (17)$$

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

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

$$X_{ij}^{f_k t} = \begin{cases} X^{f_k} = \max_{\substack{t \in T_f, \\ (i,j) \in E}} \left\{ X_{ij}^{f_k t} \right\}, & if\ Y_{ij}^{f_k t} = 1 \\ 0, & if\ Y_{ij}^{f_k t} = 0 \end{cases} \quad (18)$$

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 introduced constraint, mapping of subflows to LSPs is easy.

Link capacity constraint [8, 9, 12, 13], $\forall i \in N, \forall j \in N,$:

$$\sum_{f=1}^{|F|} \sum_{k=1}^{|K_f|} b_f \cdot \max_{t \in T_f} \left\{ X_{ij}^{f_k t} \right\} \leq c_{ij} \quad (19)$$

Constraint on the maximum number of subflows, inspired by [38]:

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

a. constant maximum number [13]:

$$\sum_{k=1}^{|K_f|} \sum_{j \in N} Y_{ij}^{f_k t} \leq N_{\max} \quad (20)$$

b. or alternatively, depending on required bandwidth b_f [12]:

$$\sum_{k=1}^{|K_f|} \sum_{j \in N} Y_{ij}^{f_k t} \leq \left\lceil \frac{b_f}{\left\lceil \sum_{j \in N} c_{ij} \Big/ \sum_{j \in N} connection_{ij} \right\rceil} \right\rceil \tag{21}$$

In summary, the proposed GMM-model follows the general mathematical framework of any MOP, given by equation (1). In this context, inspired by the review in Table 2, this paper considers 11 objective functions given by equations (4) to (14), and 7 constraints given by (15) to (21). 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* has not been considered in this first proposal (see Table 2), but to include 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 $x^* \in \mathbf{X^*}$, i.e. any solution x' outside the Pareto set ($x' \notin \mathbf{X^*}$) is outperformed by at least one solution x^* of the Pareto set ($\exists x^* \succ x'$); therefore, x' can not outperform x^* even if not all the objective functions are considered. Consequently, under the same set of constraints, any previous model or algorithm (summarized in Table 2), 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 previously found using most of the already published alternatives that consider any subset of the proposed objective functions. Now it may be clear why we call this model *generalized.*

4. GMM RESOLUTION USING A MULTIOBJECTIVE EVOLUTIONARY ALGORITHM

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 [8, 9, 11, 18, 24-27]. A MOEA, as a genetic algorithm, is inspired by the mechanics of natural evolution (based on the survival of the fittest species) [35].

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 [39]. The process continues iteratively, replacing old populations with new ones, typically saving the best found solutions (which is known as elitism), until a stop condition is reached.

In this paper, an algorithm based on the *Strength Pareto Evolutionary Algorithm* (SPEA) [37] is proposed. It holds an evolutionary population **P** and an external set $\mathbf{P_{nd}}$ with the best Pareto solutions found. Starting with a random population **P**, the individuals of **P** evolve to optimal solutions that are included in $\mathbf{P_{nd}}$. Old dominated solutions of $\mathbf{P_{nd}}$ are pruned each time a new solution from **P** enters $\mathbf{P_{nd}}$ and dominates old ones.

4.1 Encondig

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 sufficiently studied in current literature (see Table 2). However, it should be mentioned that to our best knowledge, this paper is the first one that proposes an encoding process, as shown in Fig. 1, that allows the representation of several flows (unicast and / or multicast) with as many splitting subflows as needed to optimize a given set of objective functions.

In this proposal, each chromosome consists of |**F**| flows. Each flow *f*, denoted as *(Flow f)*, contains $|\mathbf{K_f}|$ 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)* used to send information about flow *f* to the set of destinations $\mathbf{T_f}$, while the second field represents the fraction f_k of the total information of flow *f* being transmitted. Clearly, equation (2) should be satisfied to assure that all information of flow *f* arrives to destinations $\mathbf{T_f}$.

Moreover, every tree *(Tree f,k)* consists of $|\mathbf{T_f}|$ different paths *(Path f,k,t)*, one for each destination $t \in \mathbf{T_f}$. Finally, each path *(Path f,k,t)* consists of a set of nodes N_l between the source node s_f and destination $t \in \mathbf{T_f}$ (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 was implemented to choose only one of these redundant path segments, making sure that any subflow satisfies the optimality criteria.

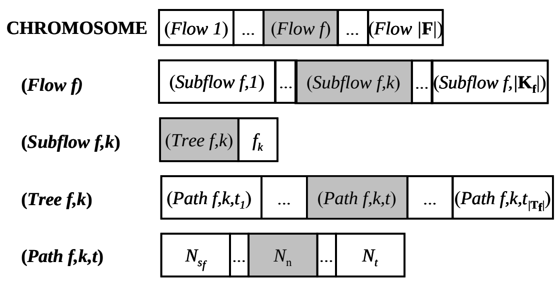


Figure 1. Chromosome representation.

One interesting advantage of the proposed encoding is that a valid chromosome completely satisfies restrictions (15) to (18), making verifying them unnecessary.

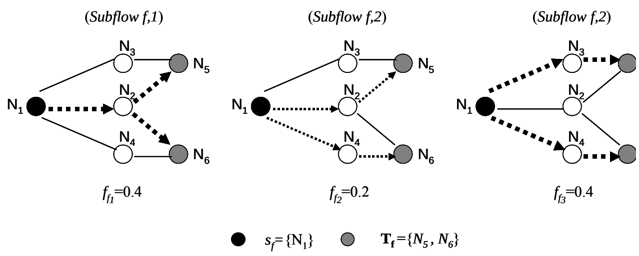
An example of the proposed encoding process is presented in Fig. 2 which shows a chromosome representation for a very simple network topology of six nodes (|**N**|=6) with a flow *f* between a source node s_f={N$_1$} and two destination nodes: $\mathbf{T_f}$ ={N$_5$, N$_6$}.

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 consist 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 = (N_1,N_2,N_5), *   /* path to destination N_5*
Path f,1,6 = (N_1,N_2,N_6))), *   /* path to destination N_6*
fraction f_1 = 0.4) *   /* subflow 1 transmits 40% of flow*
(Subflow f,2 = (*   /* second subflow*
Tree f,2=(
Path f,2,5 = (N_1,N_2,N_5), *   /* path to destination N_5*
Path f,2,6 = (N_1,N_4,N_6))), *   /* path to destination N_6*
fraction f_2 = 0.2) *   /* subflow 2 transmits 20% of flow*
(Subflow f,3 = (*   /* third subflow*
Tree f,3)=(
Path f,3,5 = (N_1,N_3,N_5), *   /* path to destination N_5*
Path f,3,6 = (N_1,N_4,N_6))), *   /* path to destination N_6*
fraction f_3 = 0.4) *   /* subflow 3 transmits 40% of flow*
)
]

Figure 2. Network topology and chromosome representation for a *flow f* with three subflows.

4.2 Initial Population

To generate an initial population **P** of valid chromosomes we have considered each chromosome at a time, building each (*Flow f*) of that chromosome at a time. For each (*Flow f*) we first generate a large enough set of different valid paths from source s_f to each destination $t \in \mathbf{T_f}$ (see line 1 of Fig. 7). Then, an initial |$\mathbf{K_f}$| is chosen as a reasonable random number that satisfies constraints on the maximum number of subflows given by equations (20) and (21). To build each of the |$\mathbf{K_f}$| subflows, we randomly generate a tree with its root in s_f and leaves in $\mathbf{T_f}$ by randomly selecting a path at a time for each destination, from the previously generated set of paths.

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

In this initialization procedure (see lines 2-3 of Fig. 7), 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 (19); otherwise, it is rejected and another chromosome is generated until the initial population **P** has the desired size P_{max}.

4.3 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 $\mathbf{P_{nd}}$) 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 [37]. In this case, the fitness for every member of $\mathbf{P_{nd}}$ 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 $\mathbf{P_{nd}}$ that dominate the individual considered. A *roulette selection operator* [39] is applied to the union set of $\mathbf{P_{nd}}$ and **P** each time a chromosome needs to be selected.

4.4 Crossover

We propose two different crossover operators: *flow crossover* and *tree crossover*. With *flow crossover* operator (line 10.a in Fig. 7), |**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 Fig. 3. A father may be chosen more than once, contributing with several flows to an offspring chromosome.

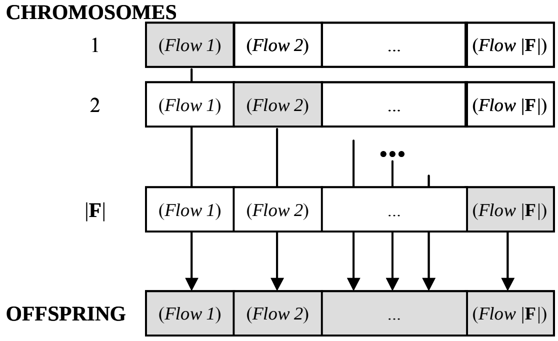


Figure 3. Flow crossover operator.

The *tree crossover operator* is based on a two-point crossover operator, which is applied to each selected pair of parent chromosomes (line 10.b in Fig. 7). In this case, the crossover is applied by doing tree exchanges between two equivalent flows of a pair of randomly selected parent-chromosomes, as shown in Fig. 4.

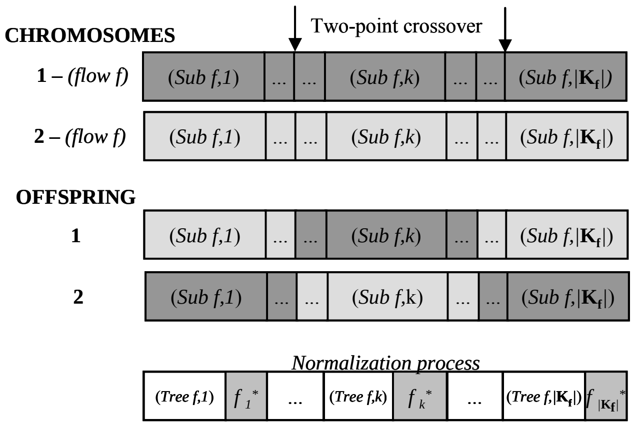


Figure 4. Tree crossover operator.

Tree crossover without normalization of the information fraction f_k usually generates infeasible chromosomes because equation (2) is not satisfied. Therefore, *a normalization process*:

$$f_k{}^* = f_k \Big/ \sum_{k=1}^{|K_f|} f_k \,, \qquad (22)$$

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

4.5 Mutation

A mutation operator is usually used to ensure that an optimal solution can be found with a probability greater than zero. This operator could improve the performance of an evolutionary algorithm, given its ability to continue the search of global optimal (or near optimal) solutions even after local optimal solutions have been found, not allowing the algorithm to be 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 11 in Fig. 7).

To apply a mutation operator, we first randomly choose a *(Flow f)* of the new offspring, in order to later select (also randomly) a *(Subflow f,k)* on 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 *segment mutation phase*, a (*Path f,k,t*) of (*Tree f,k*) is randomly chosen. At this point, a node N_j of (*Path f,k,t*) is selected as a *Mutation Point.* The *segment mutation phase* consists in finding a *new segment* to connect the selected *Mutation Point* to destination *t* (see Fig. 5), followed by the already explained (see *B - Initial Population*) *repair redundant segment process*, to achieve better chromosome quality.

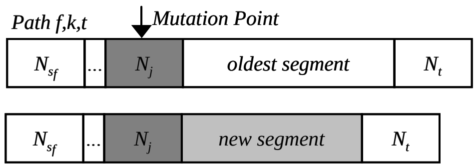


Figure 5. Segment mutation.

Finally, the *subflow fraction mutation phase* is applied to *(Subflow f,k)* by incrementing (or decrementing) flow fraction f_k in δ (see Fig. 6), followed by the *normalization process* that has already been explained, to assure that equation (2) is satisfied.

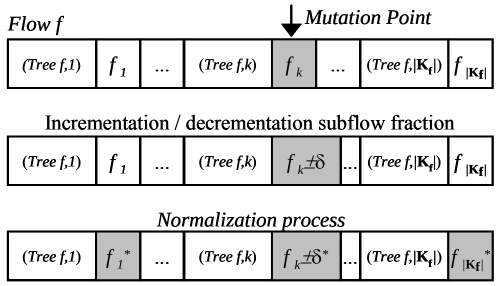


Figure 6. Subflow fraction mutation following by normalization process.

This MOEA is summarized in Fig. 7.

1	`Obtain initial set of valid paths`
2	`Generate the initial population` **`P`** `of size` P_{max}
3	`Normalize fractions and remove redundant segments for every chromosome in` **`P`**
4	`Initialize set` $\mathbf{P_{nd}}$ `as empty`
5	`DO WHILE A FINISHING CRITERIOM IS NOT SATISFIED {`
6	`Add non-dominated solutions of` **`P`** `into` $\mathbf{P_{nd}}$
7	`Remove dominated solutions in` $\mathbf{P_{nd}}$
8	`Calculate fitness of individuals in` **`P`** `and` $\mathbf{P_{nd}}$
9	`REPEAT` P_{max} `times {`

10	`Generate new chromosomes-set` **`C`** `using`
	`a` `Tree crossover with Selection (in` $\mathbf{P} \cup \mathbf{P_{nd}}$`) and` *`normalization process`*
	`b` `Flow crossover with Selection (in` $\mathbf{P} \cup \mathbf{P_{nd}}$`)`

11	`With probability` p_m `mutate set` **`C`** `and` *`normalization process`* `and remove redundant segments`
12	`Add to` **`P`** `valid chromosomes in` **`C`** `not yet included in` **`P`**
13	`END REPEAT`
14	`END WHILE`

Figure 7. Proposed Algorithm.

5. EXPERIMENTAL RESULTS

Although the aim of this work is to present a general model and not to discuss the best way to solve it, this section presents a simple problem and the corresponding experimental results using the proposed MOEA, as an illustration of what has been previously stated.

5.1 Network Topology

The chosen topology is the well-known 14-node NSF (National Science Foundation) network shown in Fig. 8 ($|\mathbf{N}|=14$) [8, 9, 12-14, 27]. The costs on the links represent the delays (d_{ij}) and all links are assumed to have 1.5 Mbps of bandwidth capacity (c_{ij} = 1.5 Mbps $\forall(i,j)\in E$).

Two flows with the same source, s_f=N_0, are considered. The egress subsets are $\mathbf{T_1}$={N_5, N_9} and $\mathbf{T_2}$={N_4, N_9, N_{12}}. The transmission rates are b_1=256 Kbps for the first flow and b_2=512 Kbps for the second flow.

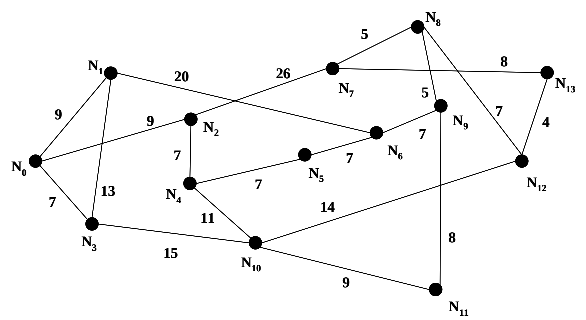


Figure 8. NSF network.

5.2 Resolution of the Test Problem

A complete set of found non-dominated solutions (best calculated approximation to the optimal Pareto set **X*** in a run) had 748 chromosomes. This large number is due to the large number of objective functions and the fact that the same set of subtrees with different fractions f_k of the flows are considered as different solutions because each one represents another compromise between conflicting objective functions. When needed, a clustering technique, that is included in the implemented SPEA, may be used to reduce the number of calculated non-dominated solutions to a maximum desired number [37]. Since there is no space to present all 748 non-dominated solutions, Table 3 shows some of the best calculated solutions considering one objective function at a time. Each row (identified by an ID given in the first column) represents a non-dominated solution whose chromosome is omitted to save space. The following 11 columns represent the different objective functions defined in equations (4) to (14). The cells in bold emphasize an optimal objective value. To better exemplify a solution, certain chromosomes were omitted from Table 3.

Table 3. Some calculated Pareto Front solutions

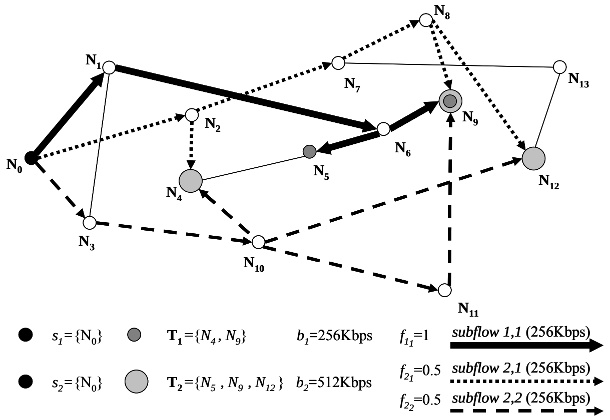
ID	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}	ϕ_{11}
1	16,7%	26	3,3	4	1	288	22,5	47	17	4096	3
2	33,3%	16	3,2	4	0	180	36,0	39	0	4096	2
3	25,0%	26	3,3	4	1	275	20,9	47	17	4608	3
4	33,3%	25	3,1	4	1	277	21,8	45	17	4608	3
5	33,3%	26	3,3	4	1	282	22,1	45	17	4352	3
6	33,3%	25	3,1	4	1	277	21,5	45	17	4864	3
7	33,3%	27	3,4	4	1	305	22,8	47	11	4224	3
8	33,3%	26	3,3	4	1	282	21,9	45	17	4480	3
9	25,0%	25	3,1	4	1	279	22,2	47	17	4352	3
10	33,3%	15	3,0	4	0	163	32,6	39	0	4608	2
11	33,3%	26	3,3	4	1	282	21,8	45	17	4544	3
12	33,3%	26	3,3	4	1	282	22,0	45	17	4416	3
13	25,0%	28	3,5	4	1	315	23,3	47	11	4480	3
14	33,3%	25	3,1	4	1	271	21,4	47	11	4352	3
15	33,3%	26	3,3	4	1	282	22,2	45	17	4288	3
16	50,0%	25	3,6	5	0	245	25,2	47	2	5792	3
17	50,0%	28	4,0	7	0	254	26,5	56	2	5312	3
18	50,0%	19	3,8	5	0	154	30,8	41	0	4864	2
19	33,3%	153	4,8	10	7	1545	7,1	104	77	6176	13

Some solutions presented in Table 3 are clearly non-dominated because they are the best ones in at least one objective, like the ones with ID=1 with the minimum value of ϕ_1 or ID=10 with the minimum value of ϕ_2 and ϕ_3 to just name a few. However, most solutions are non-dominated because they are different compromise solutions. As an example, solutions from ID=1 to

ID=15 are all optimal considering ϕ_4, but each one represents a different compromise between conflicting objective functions.

In this example when a flow is not split into subflows, we potentially need the least amount of LSPs (ϕ_{11}=2) and there is no hop count or delay variations between subflows (ϕ_5=0, ϕ_9=0), as shown in solutions with ID=2, 10, 18 and 20. However, it is possible to have a delay variation (ϕ_9>0) even when there is no hop count variation (ϕ_5=0) if at least one flow is split (ϕ_{11}>2), like the non-dominated solutions with ID=16 and 17.

Fig. 9 presents the chromosome solution of ID=1 in Table 3 showing that flow 1 of b_1=256 Kbps from source s_1 ={N_0} to **T_1**={N_5, N_9} is transmitted without splitting (fraction 1_1 = 1.0) while flow 2 of b_2=512 Kbps from source s_2 ={N_0} to **T_2**={N_4, N_9, N_{12}} is split into two subflows transmitting 256 Kbps per subflow (fraction 2_1 = fraction 2_2 = 0.5).



[
(Flow 1 =
(Subflow 1,1 = (
Tree 1,1=(
Path 1,1,5 = (N_0,N_1,N_6,N_5),
Path 1,1,9 = (N_0,N_1,N_6,N_9))),
fraction 1_1 = 1.0)
)
(Flow 2 =
(Subflow 2,1 = (
Tree 2,1=(
Path 2,1,4 = (N_0,N_2,N_4),
Path 2,1,9 = (N_0,N_2,N_7,N_8,N_9),
Path 2,1,12 = (N_0,N_2,N_7,N_8,N_{12}))),
fraction 2_1 = 0.5)
(Subflow 2,2 = (
Tree 2,2=(
Path 2,2,4 = (N_0,N_3,N_{10},N_4),
Path 2,2,9 = (N_0,N_3,N_{10},N_{11},N_9),
Path 2,2,12 = (N_0,N_3,N_{10},N_{12}))),
fraction 2_2 = 0.5)
)
]

Figure 9. Chromosome and subflow representation of non-dominated solution (ID=1 in Table 3).

5.3 Correlation Analysis

A correlation analysis between each pair of objective functions was also performed to get an idea of the real necessity of using (or not) that large a number of objective functions. A very large correlation clearly means that if one objective function is optimized another one with a high correlation is also indirectly optimized.

Table 4 presents these correlation values between the 11 objective functions given by equation (4) to (14), considering the whole experimental set of 748 non-dominated solutions.

Table 4. Correlations between objective functions

	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}	ϕ_{11}
ϕ_1	-0,22	-0,16	-0,24	-0,26	-0,25	0,19	-0,28	-0,19	0,09	-0,20
ϕ_2		0,69	0,77	0,80	1,00	-0,80	0,78	0,78	0,56	0,98
ϕ_3			0,78	0,48	0,66	-0,35	0,62	0,41	0,52	0,59
ϕ_4				0,82	0,76	-0,61	0,90	0,73	0,47	0,72
ϕ_5					0,81	-0,81	0,85	0,93	0,34	0,80
ϕ_6						-0,81	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

As shown in Table 4 there are very large correlations between some objectives, such as:

- the total hop count (ϕ_2) and total delay (ϕ_6), with a correlation of almost 1, this is easy to understand given that a longer path usually implies more delay;
- the total hop count (ϕ_2) and number of subflows (ϕ_{11}), with a value of 0.98, given that the use of splitting implies the use of multiple-routes and therefore, more links;
- the maximal hop count (ϕ_4) and maximal delay (ϕ_8), with a correlation of 0,90, because the longest path in the hop count normally has the longest delay. In fact, equation (7) and (11) are very similar. Since the same reasoning applies for the minimal path, it is also easy to understand the high correlation of 0,93 for
- the hop count variation (ϕ_5) and delay variation (ϕ_9). Finally,
- the total delay (ϕ_6) and the number of subflows (ϕ_{11}) with 0.98, as a logical consequence of the high correlation of both objective functions with the total hop count (ϕ_2).

More experimental results are needed to make a final conclusion, but it is clear that all objective functions are not really needed at the same time. We have considered them for sake of completeness, just to make sure that an optimal solution from a previous work that consider a given objective will also be a solution of the GMM-model.

6. MODELS COMPARISON

Given that one of the main contributions of this work is the formalization of the GMM-model as a general model, it is interesting to compare it to another recently published model, like the MHDB-model [12-14]. In 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, 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 2) are not good enough for finding all the solutions of a Pareto set in multiobjective non-convex problems, as stated in [35].
- The GMM-model finds better solutions (considering more objective functions) or at least, ones that are just as good.
- Correlations between the considered objective functions prove that they are not all needed in practical cases, but using them insures that solutions are all Pareto optimal considering any desired subset of objective functions.
- 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 [40].
- 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.

Finally, considering the example in section V and its experimental results, it can be emphasized that by using the MHDB-model it would only be possible to find one solution (as the one of Fig. 9) but not a whole Pareto set as the one with 748 solutions found by the GMM-model.

7. CONCLUSIONS

This paper has presented a novel taxonomy of traffic engineering load balancing, classifying 35 publications including their objective functions, constraints and proposed heuristic. Using this classification (given in Table 2), it is clear that no previous work has studied TE load balancing considering multicast flow, multitree and splitting simultaneously in a multiobjective context.

As a consequence, in this work 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 easily combine objective functions in a unique cost function that can be combined with restrictions on some objectives, as an upper bound on the delay, hop count, etc. However, any optimal single objective solution, like the ones proposed in several previous papers, would be a solution of the GMM-model or dominated by a Pareto solution of it 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 new encoding process to represent multitree-multicast solutions using splitting. This MOEA found a set of 748 non-dominated solutions for a very simple multicast test problem based on the well-known NSF network. A correlation analysis of this set of non-dominated solutions was also included, emphasizing that several objective functions are highly correlated and therefore, not really needed for some practical applications.

The GMM-model was compared to a previous MHDB-model, proving to have several advantages such as: a whole Pareto set of solutions that are better when all objective functions are considered, and a clear identification of subflows that makes mapping LSPs for a MPLS implementation easy.

For future work, we plan to improve MOEA solving more complex problems, considering different topologies and including some objective functions that still haven't been considered, such as *Packet Loss*. We also plan to formulate an extended model that includes backup paths and dynamic multicast group considerations with an efficient algorithm that can solve the dynamical inclusion of new destinations or leave of existing nodes, in a given multicast tree; and therefore, the taxonomy presented will be extended to include these characteristics.

8. REFERENCES

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

[2] B. Fortz, M. Thorup. "Optimizing OSPF/IS-IS weights in a changing world". *JSAC, February 2002.*

[3] A. Sridharan, R. Guerin, C. Diot. "Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks". *INFOCOM 2003.*

[4] Y. Wang, Z. Wang, L. Zhang. "Internet Traffic Engineering without Full Mesh Overlaying". *INFOCOM 2001.*

[5] Y. Lee, Y. Seok, Y. Choi, C. Kim. "A Constrained Multipath Traffic Engineering Scheme for MPLS Networks". *ICC 2002.*

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

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

[8] J. Crichigno, B. Barán. “Multiobjective Multicast Routing Algorithm”. *IEEE ICT 2004.*

[9] J. Crichigno B. Barán. “Multiobjective Multicast Routing Algorithm for Traffic Engineering”. *IEEE ICCCN 2004.*

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

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

[12] Y. Donoso, R. Fabregat, J. Marzo. “Multi-Objective Optimization Algorithm for Multicast Routing with Traffic Engineering”. *IEEE ICN 2004.*

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

[14] R. Fabregat, Y. Donoso, J.L. Marzo, A. Ariza. “A Multi-Objective Multipath Routing Algorithm for Multicast Flows. *SPECTS 2004.*

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

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

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

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

[19] N.S.V. Rao, S.G. Batsell. “QoS Routing via multiple Paths using Bandwidth Reservation”. *INFOCOM 1998.*

[20] S. Vutukury, J.J. Garcia. "A traffic engineering approach based on minimum-delay routing". *ICCN 2000.*

[21] J.C. Chen, and S.H. Chan. "Multipath Routing for Video Unicast over Bandwidth-Limited Networks". *GLOBECOM 2001.*

[22] C. Cetinkaya, E. Knightly. “Opportunistic Traffic Scheduling Over Multiple Network Paths," *INFOCOM 2004.*

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

[24] A. Roy, N. Banerjee, S. Das. “An Efficient Multi-Objective QoS-Routing Algorithm for Wireless Multicasting”. *INFOCOM 2002.*

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

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

[27] J. Crichigno, B. Barán. “A Multicast Routing Algorithm using Multiobjective Optimization”. *IEEE ICT 2004.*

[28] L. Layuan, L. Chunlin. “QoS Multicast Routing in Networks with Uncertain Parameter”. *IEEE IPDPS 2003.*

[29] H. Cho, J. Lee, B. Kim. “Multi-path Constraint-based Routing Algorithms for MPLS Traffic Engineering”. *ICC 2003.*

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

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

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

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

[34] Y. Cui, K. Xu, J. Wu. “Precomputation for Multi-constrained QoS Routing in High-speed Networks”. *INFOCOM 2003.*

[35] D. A. van Veldhuizen. “Multiobjective Evolutionary Algorithms: Classifications, Analyses and New Innovations”. *Ph.D. thesis. Air Force Institute of Technology, 1999.*

[36] C. Kim; Y. Choi; Y. Seok. Y. Lee. “A Constrained Multipath Traffic Engineering Scheme for MPLS Networks”. *ICC 2002.*

[37] E. Zitzler and L. Thiele. “Multiobjective Evolutionary Algorithm: A comparative case study and the Strength Pareto Approach”. *IEEE Trans. Evolutionary Computation. Vol. 3, No. 4, 1999.*

[38] M. Krishnamoorthy, A.T. Ernst. “Comparison of algorithms for the degree constrained minimum spanning tree”. *Journal of heuristics, Vol. 7, 2001.*

[39] D. E. Goldberg. “Genetic Algorithms in Search, Optimization and Machine Learning”. *Addison-Wesley Publishing, 1989.*

[40] F. Solano, R. Fabregat, Y. Donoso, "Sub-flow assignment model of multicast flows using multiple p2mp LSPs". *CLEI 2004.*