

# Multiconstrained Optimization of Networks with Multicast and Unicast Traffic

Pedro Sousa<sup>1</sup>, Miguel Rocha<sup>1</sup>, Paulo Cortez<sup>2</sup>, and Miguel Rio<sup>3</sup>

<sup>1</sup> Dep. Informatics / CCTC - Univ. Minho - Braga - Portugal  
pns@di.uminho.pt mrocha@di.uminho.pt

<sup>2</sup> Dep. Information Systems/ Algoritmi - Univ. Minho - Guimarães - Portugal  
pcortez@dsi.uminho.pt

<sup>3</sup> University College London - London - UK  
m.rio@ee.ucl.ac.uk

**Abstract.** This paper presents an OSPF routing optimization framework taking into account a set of multiconstrained QoS requirements of the networking domain. The proposed optimization approach, based on Evolutionary Computation, is able to handle network scenarios with both unicast and multicast traffic, providing high quality configurations for single-topology or multi-topology routing approaches. The results clearly show the effectiveness of the devised optimization methods, allowing for the development of management tools automatically providing enhanced configurations to improve the QoS performance of the network.

## 1 Introduction

Nowadays, TCP/IP networks are facing the challenge of providing effective support to a number of advanced services with strict QoS requirements. Services such as Interactive TV, virtual reality, video-conferencing or video surveillance are examples of services that would gain from QoS enabled multicast content delivery. An effective network support to these services requires data delivery with minimal loss along with acceptable end-to-end delays, an even more crucial issue considering the cases of interactive applications requiring strict commitments regarding round-trip time delays. Another relevant point is the fact that the widespread use of multicast has never occurred in the Internet. However, in closed TCP/IP networks, where its scalability problems are not a deterrent, multicast has been used to support some advanced services.

In this context, Traffic Engineering (TE) techniques can be used to improve network performance by achieving near-optimal configurations for routing protocols. TE approaches can be classified into: Multi-Protocol Label Switching (MPLS) [3][2] and pure IP-based intra-domain routing protocols. In the case of MPLS, packets are encapsulated with labels at ingress points, that can be used to route these packets along an explicit label-switched path. However, the use of MPLS presents significant drawbacks when compared with traditional IP routing mechanisms. As regards intra-domain routing protocols, the most commonly

used today is Open Shortest Path First(OSPF)[12]. Here, the administrator assigns weights to each link in the network, which are then used to compute the best path from each source to each destination using the Dijkstra algorithm [4].

A number of studies have proposed TE procedures which optimize the weights of intra-domain routing protocols to achieve near optimal routing, taking as input the expected traffic demands. This was the approach adopted by the work of Fortz and Thorup (2000) [6] which proposed some local search heuristics to deal with this NP-hard problem. Another approach was the use of Evolutionary Algorithms (EAs) to improve these results [5]. Additional research has been carried out with the objective of pursuing multiconstrained QoS optimization for unicast traffic [11].

In this paper, EAs are employed to provide network administrators with OSPF weights able to optimize network behaviour, taking simultaneously into account unicast and multicast demands of a given domain (for an example see Fig. 1). The conducted study considers two distinct scenarios: (i) optimizing overall network congestion or (ii) optimizing both the overall network congestion and end-to-end delays for multicast traffic.

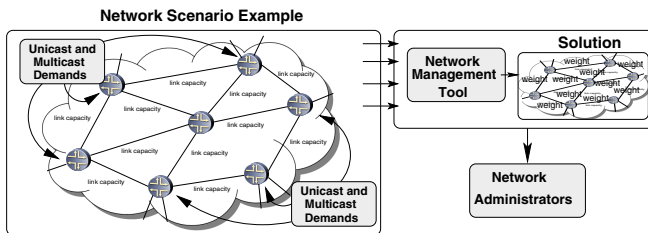


Fig. 1. Example of a network scenario

This work is based on the reasoning that in the optimization process both the unicast and multicast demands should be considered simultaneously, in contrast with previous work where optimization is performed in two distinct phases, the first for unicast traffic and the second devoted to multicast optimization [13]. Furthermore, it presents a novel approach that allows the simultaneous optimization of the overall network congestion and multicast end-to-end delays, considering scenarios that use traditional OSPF weights and also the possibility of using a multi-topology approach where two layers of weights are considered.

## 2 Problem Formulation

### 2.1 Unicast Traffic

In this section, a model for minimizing congestion in a network only with unicast traffic demands will be described. This is based on the framework proposed in [6]. The general routing problem [1] that underpins this work represents routers and links by a set of nodes ( $N$ ) and arcs ( $A$ ) in a directed graph  $G = (N, A)$ .

In this model,  $c_a$  represents the capacity of each link  $a \in A$ . A demand matrix  $D$  is available, where each element  $d_{st}$  represents the traffic demand between nodes  $s$  and  $t$ . For each arc  $a$ , the variable  $f_a^{(st)}$  represents how much of the traffic demand between  $s$  and  $t$  travels over arc  $a$ . The total unicast load  $l_a$ , the link utilization rate  $u_a$  and the congestion measure  $\Phi_a$  for arc  $a$  are given in Equations 1, 2 and 3, where  $p$  is a penalty function that has small values near 0, but as the values approach the unity it becomes more expensive and exponentially penalizes values above 1 [6].

In OSPF, all arcs have an integer weight, used by each node to calculate the shortest paths to all other nodes in the network, using the Dijkstra algorithm [4]. The traffic from a given source to a destination travels along the shortest path. If there are two or more paths with equal length, traffic is evenly divided among the arcs in these paths (load balancing) [9]. Let us assume a given a weight assignment, and the corresponding values of  $u_a$ . In this case, the total routing cost is expressed by Eq. 4 for the loads and penalties calculated based on the given OSPF weights. In this way, the *OSPF weight setting problem* is equivalent to finding the optimal weight value for each link, in order to minimize  $\Phi$ . The congestion measure can be normalized ( $\Phi^*$ ) over distinct scenarios to values in the range [1,5000]. It is important to note that in the case when all arcs are exactly full ( $l_a = c_a$ ), the value of  $\Phi^*$  is  $10^2/3$ , a value that will be considered a threshold that bounds the acceptable working region of the network.

$$l_a = \sum_{(s,t) \in N \times N} f_a^{st} \quad (1) \quad u_a = \frac{l_a}{c_a} \quad (2) \quad \Phi_a = p(u_a) \quad (3) \quad \Phi = \sum_{a \in A} \Phi_a \quad (4)$$

## 2.2 Multicast Demands

A model that considers minimizing congestion only considering multicast traffic in the network will be described, that is based on the work by Wang and Pavlou (2007) [13]. If there are unicast and multicast demands, this model can be used to perform a two-step optimization process (explained in Section 3.2). The multicast demands are given for a set of  $G$  groups, where for each group  $g \in G$  the following parameters are defined: a root node  $r_g$ , a bandwidth demand  $M_g$  and a set of receivers ( $V_g$ ). The multicast optimization problem is typically defined as the computation of a bandwidth constrained Steiner tree, with the objective of minimizing overall bandwidth consumption, using integer programming. The target is to instantiate a number of binary decision variables:  $y_a^g$ , are equal to 1 if link  $a$  is included in the multicast tree for group  $g$ ; and  $x_a^{g,k}$  are equal to 1 if link  $a$  is included in the multicast tree for group  $g$ , in the branch from the root node to receiver  $k$ . The objective is to minimize the overall bandwidth consumption ( $L_1$ ) as expressed by Eq. 5.

$$L_1 = \sum_{g \in G} \sum_{a \in A} M_g y_a^g \quad (5)$$

The deployment of the obtained Steiner trees can be enforced by using an explicit routing overlay, through MPLS on a per-group basis. An alternative

with some advantages, previously discussed, is to consider that the routing will be achieved by using an intra-domain protocol such as OSPF. In this case, the tree for a given group will be built from the shortest paths between the root node and each receiver. Therefore, the values assigned to  $y_a^g$  variables will be computed as follows:  $y_a^g$  is equal to 1 if link  $a$  is in the shortest path from the root node  $g$  to at least one of the receivers in  $V_g$ , and is equal to 0 otherwise.

In previous work [13], EAs have been proposed to optimize OSPF weights for multicast traffic. The objective function used in this case is based on the overall network load ( $L_1$ ) but also on the excessive bandwidth allocated to overloaded links ( $L_2$ ), that can be given by:

$$L_2 = \sum_{a \in A} [w_a \sum_{g \in G} (M_g y_a^g) - c_a] \quad (6) \quad w_a = \begin{cases} 0, & \text{if } \sum_{g \in G} M_g y_a^g \leq c_a \\ 1, & \text{otherwise} \end{cases} \quad (7)$$

The EA's fitness is, therefore, given by:

$$f(L_1, L_2) = \frac{\mu}{\alpha L_1 + \beta L_2} \quad (8)$$

where  $\mu$ ,  $\alpha$  and  $\beta$  are constants, whose values are set to  $10^7$ , 1 and 10 respectively.

### 2.3 Unified Congestion Model with Unicast and Multicast Demands

In this work, a unified approach will be proposed that is able to reach OSPF weights that optimize the network congestion measure, simultaneously considering unicast and multicast demands. In this case, the multicast load  $ml_a$  for a given link  $a$  is given by Eq. 9.

The values of  $y_a^g$  will be calculated from the OSPF weights as explained in the previous section. So, the total load on a given arc  $a$  is given by  $l_a = ml_a + ul_a$  where  $ul_a$  is the unicast load in arc  $a$  (given by  $l_a$  in the previous section). It should be noted that  $l_a$  here takes the meaning of the total load in the network, while in Section 2.1,  $l_a$  is only used for unicast loads since in that case those were the only loads considered. After calculating the overall values of  $l_a$  for all links, the process proceeds as described in Section 2.1, in order to reach the normalized congestion measure  $\Phi^*$ . Another interesting measure of the network performance in this scenario is the excessive bandwidth in overloaded links (BOL). This is a generalization of  $L_2$  but now applied to the global loads and not only to the multicast traffic. (Eq. 10).

$$ml_a = \sum_{g \in G} M_g y_a^g \quad BOL = \sum_{a \in A} z_a (l_a - c_a) \quad z_a = \begin{cases} 0, & \text{if } l_a \leq c_a \\ 1, & \text{otherwise} \end{cases} \quad (11)$$

(9)
(10)

### 2.4 Modeling Delays in Multicast Traffic

In this section, the previous model is enriched by considering also end-to-end delays associated with multicast traffic. Delay requirements were defined at the group level: a maximum allowed delay  $md_g$  is defined for each group  $g$ , that

is applied to every pair (root, receiver) in that group. This means that the maximum allowed delay must be respected for all receivers.

A cost function was developed to evaluate the delay compliance for each scenario. The *delay compliance ratio* for a given receiver  $k$  in a group  $g$  is defined as in Eq. 12 where  $s$  is the root node  $r_g$  and  $t$  is the receiver  $k$  of group  $g$ . In this expression,  $Del_{st}$  stands for the average delay of the traffic between the nodes  $s$  and  $t$ , a value calculated by considering all paths between  $s$  and  $t$  with minimum cost and averaging the delays in each. The delay in each path is the sum of the propagation delays in its arcs and queuing delays in the nodes along the path. Note that in some network scenarios the latter component might be neglected (e.g. if the propagation delay component has an higher order of magnitude than queuing delays<sup>1</sup>).

A penalty for delay compliance can be calculated using function  $p$  ( $\gamma_{kg}$  in Eq. 13). This, in turn, allows the definition of a delay minimization cost function, given a set of OSPF weights ( $w$ ) (Eq. 14) where the  $\gamma_{kg}(w)$  values represent the delay penalties for each end-to-end path, given the routes determined by the OSPF weight set  $w$ .

$$dc_{kg} = \frac{Del_{st}}{md_g} \quad (12) \quad \gamma_{kg} = p(dc_{kg}) \quad (13) \quad \gamma(w) = \sum_{g \in G} \sum_{k \in V_g} \gamma_{kg}(w) \quad (14)$$

This function can be normalized dividing the values by the sum of all minimum end-to-end delays (for each pair of nodes the minimum end-to-end delay -  $minDel_{st}$  - is the delay of the path with minimum possible overall delay):

$$\gamma^*(w) = \frac{\gamma(w)}{\sum_{g \in G} \sum_{k \in V_g} minDel_{st}} \quad (15)$$

where, as before,  $s$  is the root node  $r_g$  and  $t$  is the receiver  $k$  of group  $g$ .

It is possible to define a new optimization problem that is multi-objective, where the aim is to find the set of OSPF weights ( $w$ ) that simultaneously minimizes the functions  $\Phi^*(w)$  and  $\gamma^*(w)$ . The algorithms described in the following sections use a linear weighting scheme where the cost of the solution is given by:

$$f(w) = \alpha \Phi^*(w) + (1 - \alpha) \gamma^*(w), \alpha \in [0, 1] \quad (16)$$

This scheme is effective since both cost functions are normalized in the same range and the parameter  $\alpha$  can be tune the trade-off between both components.

### 3 Optimization Algorithms

#### 3.1 Evolutionary Algorithms

Evolutionary Algorithms (EAs) [8] are a popular family of optimization methods, inspired in the biological evolution. These methods work by evolving a

<sup>1</sup> In this work experiments the network queuing delays at each network node were not considered. However, if required, they might be approximated resorting to queuing theory and taking into account the scheduling mechanisms in use and the capacity and utilization rates of the output links of the network nodes along the path.

population, i.e. a set of individuals, each encoding solutions to a target problem in an artificial chromosome. Each individual is evaluated through a fitness function, that assigns it a numerical value, corresponding to the quality of the encoded solution. EAs are stochastic methods due to their selection process. In fact, individuals selected to create new solutions are taken from the population using probabilities. Highly fit individuals have a higher probability of being selected, but the less fit still have their chance.

In the proposed EA, each individual encodes a solution in a direct way, i.e. as a vector of integer values, where each value corresponds to the weight of an arc in the network (the values range from 1 to  $w_{max}$ ). Therefore, the size of the individual equals the number of links in the network. If multiple topologies are used, i.e. different sets of weights for unicast and multicast, the size of the individual is twice the number of links and the two sets of weights are encoded linearly, i.e. the first  $L$  genes encode the weights for unicast traffic, while the latter  $L$  links encode the weights for multicast ( $L$  is the number of links).

The weight values for individuals in the initial population are randomly generated, taken from a uniform distribution. In order to create new solutions, several reproduction operators were used, more specifically two mutation and one crossover operator: *i) Random Mutation*, replaces a given weight value by a random value, within the allowed range; *ii) Incremental/decremental Mutation*, replaces a given weight value  $w$  by  $w + 1$  or by  $w - 1$  (with equal probabilities); *iii) Uniform crossover*, a standard crossover operator [8].

The operators are all used to create new solutions with equal probabilities. The selection procedure is done by converting the fitness value into a linear ranking in the population, and then applying a roulette wheel scheme. In each generation, 50% of the individuals are selected from the previous generation, and 50% are bred by the application of the genetic operators over selected parents. A population size of 100 individuals was considered.

### 3.2 Optimization Approaches

Three distinct optimization approaches are compared, with the aim to optimize OSPF weights in networks where both unicast and multicast demands are available. All these methods use EAs as the optimization engine. The first method is a **2-step optimization process (2S)**, based on the proposal from Section 2.2 [13], that can be described as: *i)* the OSPF weights are optimized (using EAs) to minimize congestion penalties ( $\Phi^*$ ) only taking into account the unicast demands; *ii)* the bandwidths used for each link in unicast traffic are deduced from the link capacities and *iii)* a second optimization process is conducted, where a different set of weights is calculated from multicast traffic only, by running a new EA with  $f(L_1, L_2)$  (Eq. 8) as the fitness function. This method can only be used in the optimization of network congestion and assumes that a protocol that allows multiple sets of weights, each for a distinct type of traffic, is deployed. This is the case, for instance, of the multi-topology protocol MT-OSPF [10].

The remaining alternatives are based on the model proposed in Sections 2.3 and 2.4. Using this model, two different optimization approaches may be followed:

**Single topology (ST)**, i.e. a single set of OSPF weights is used for both types of traffic demands and **Multiple topologies (MT)**, i.e. two sets of OSPF weights are used, one for unicast traffic and the other for multicast demands. These two methods can address both the problem of network congestion, where the fitness function is simply  $\Phi^*$ , calculated as shown in Section 2.3, as well as to optimize both congestion and multicast end-to-end delays. In the latter case, the cost function is given by the Eq. 16 (in this work  $\alpha$  was set to 0.5).

## 4 Experiments and Results

To evaluate the proposed algorithms, a number of experiments were conducted. The experimental platform is presented in Fig. 2. All algorithms and the OSPF routing simulator were implemented using the Java language. A set of 3 network topologies was created using the Brite topology generator [7], varying the number of nodes ( $N = 30, 80, 100$ ) and the average degree of each node was kept ( $m = 4$ ). This resulted in networks ranging from 110 to 390 links. The link bandwidth was generated by a uniform distribution between 1 and 10 Gbits/s. The networks were generated using the Barabasi-Albert model, using a heavy-tail distribution and an incremental grow type (parameters  $HS$  and  $LS$  were set to 1000 and 100). Next, the unicast demand matrices ( $D$ ) were generated (two distinct matrices for each network). A parameter ( $D_p$ ) was considered, representing the expected mean of congestion in each link (values for  $D_p$  were 0.2 and 0.3).

The generation of the multicast traffic demands was based on the following: Firstly, for each network the number of groups  $G$  was set equal to the number of nodes. The root node for each group was randomly chosen from the set of nodes (with equal probabilities). For each group, the number of receivers was generated from the range  $[2, n/2]$ , where  $n$  is the number of nodes. The set of receivers  $V_g$  was created with the given cardinality, by randomly selecting a set of nodes different from the root. Finally, the demand  $M_g$  was generated taking a parameter ( $R$ ) into account.  $R$  is defined as the ratio between the total multicast demands and the total unicast demands. Given  $R$  and given the unicast demands, a target is calculated for the total multicast demands. The group demands are generated by dividing the target value by the different groups in an uneven way, so that groups with different demands are created resulting in a more plausible scenario. In the experiments, the value of  $R$  was set to 0.5.

For the generation of the delays for each group, the strategy was to calculate the average of the minimum possible delay between the root node and all

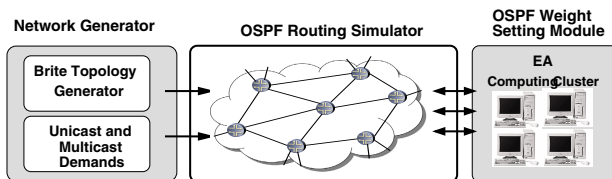


Fig. 2. Experimental platform for EA's performance evaluation

**Table 1.** Results for the optimization of the overall congestion

Nodes	Demands ( $D_p$ )	Metric	ST	MT	2S
30	0.2	$\Phi^*$	1.38	1.31	1.83
		BOL	0	0	42
	0.3	$\Phi^*$	3.32	2.83	7.78
		BOL	257	128	875
80	0.2	$\Phi^*$	1.44	1.38	1.74
		BOL	0	0	7
	0.3	$\Phi^*$	2.63	2.50	3.97
		BOL	227	247	821
100	0.2	$\Phi^*$	1.35	1.31	1.69
		BOL	0	0	7
	0.3	$\Phi^*$	4.51	3.18	4.54
		BOL	2122	665	1113

receivers of the group. A noise value is added that can change this value by  $\pm 25\%$ . A parameter ( $DR_p$ ) was considered, representing a multiplier applied to the previous value (values for  $DR_p$  in the experiments were 3 and 4). The termination criteria for all optimization approaches consisted in a maximum number of solutions evaluated. This value ranged from 100000 to 300000, increasing linearly with the number of links. For all cases,  $w_{max}$  was set to 20 and 20 runs were executed in each instance and the results presented are the means.

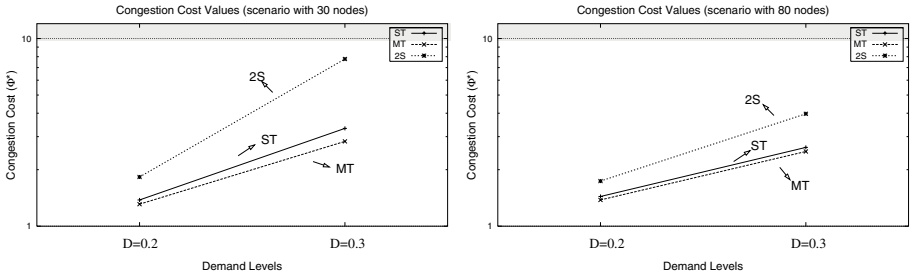
Two sets of experiments were conducted in the aforementioned instances, regarding: (i) the optimization of the overall network congestion and (ii) the simultaneous optimization of congestion and multicast end-to-end delays. The results are given and discussed in the next two sub-sections.

#### 4.1 Results for Congestion

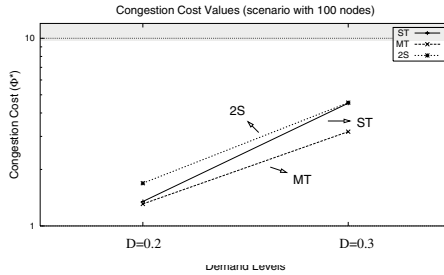
In this case, there were 6 scenarios (3 networks, 2 values of  $D_p$ ), 3 optimization approaches (2S, ST, MT). Table 1 shows the results for this task. The first column gives the number of nodes in the network; the second shows the demand generation parameter ( $D_p$ ); the third shows the performance metrics; the remaining columns give the results of each approach.

A different perspective is shown in Figs. 3 and 4 where the congestion measure ( $\Phi^*$ ) for the three networks is plotted. In each plot, the two scenarios in terms of demands are shown. The values are shown in a logarithmic scale, given the exponential nature of the penalty function. It is clear from these results that the 2S approach leads to sub-optimal results, in terms of overloaded links and network congestion (visible both in the BOL and  $\Phi^*$ ). Both the MT and ST show better results, being able to keep the network in an acceptable behaviour. When comparing ST and MT, the results are quite near, i.e. the gain obtained by using MT is not impressive, although it increases with the value of  $D_p$  (the problem is harder). In practical terms, this means that if the network has lots of bandwidth resources and low demands, it is probably not worth to pay the cost of deploying a multi-topology protocol. On the other hand, using this kind of protocol allows the network to support higher demands with the same resources.





**Fig. 3.** Results for congestion measure - network with a) 30 nodes and b) 80 nodes



**Fig. 4.** Results for congestion measure - network with 100 nodes

## 4.2 Results for Congestion and Delays

In this case, the number of considered scenarios increases to 12, since two values of  $DR_p$  are considered. The optimization approaches in this case are only MT and ST, since the 2S method can not be applied in this case. The results of the experiments are summarized in Tables 2, 3 and 4, for each of the networks. In each table, the first two columns show the parameters  $D_p$  and  $DR_p$ , the third gives the metrics and the last two show the results of both algorithms.

A different perspective is shown in Figs. 5 and 6. In these figures a distinct data representation is used, with the congestion values represented in the  $x$ -axis and the delays cost values in the  $y$ -axis. As before, the white area represents the acceptable working region, meaning that the proposed routing configurations are able to support the traffic demands of the domain and simultaneously obey the delay requirements of the considered multicast groups.

Note that in this perspective, acceptable OSPF configurations are expected to be harder to find. As an example, for the the case of the multi-topology approach, the OSPF weights devised to handle the multicast traffic are expected to provide network paths able to support both the traffic demands and the delay requirements of the multicast groups. In the same perspective, scenarios only using a single level of OSPF weights will be faced with the challenge of finding a single level of link weights able to induce both unicast and multicast paths able to satisfy both the overall traffic demands and the multicast delays requirements.

**Table 2.** Results for congestion and delay optimization - network with 30 nodes

Demands ( $D_p$ )	Delays ( $DR_p$ )	Metric	ST	MT
0.2	4	$\Phi^*$	1.65	1.38
		BOL	2	0
		$\gamma^*$	1.36	1.12
0.2	3	$\Phi^*$	1.74	1.38
		BOL	2	0
		$\gamma^*$	1.66	1.41
0.3	4	$\Phi^*$	7.35	3.62
		BOL	985	211
		$\gamma^*$	5.05	1.32
0.3	3	$\Phi^*$	24.7	3.24
		BOL	3039	162
		$\gamma^*$	1.91	1.54

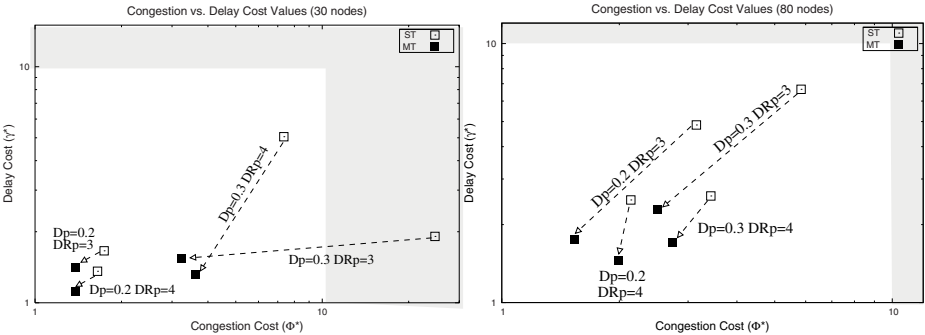
**Table 3.** Results for congestion and delay optimization - network with 80 nodes

Demands ( $D_p$ )	Delays ( $DR_p$ )	Metric	ST	MT
0.2	4	$\Phi^*$	2.14	1.99
		BOL	195	151
		$\gamma^*$	2.49	1.45
0.2	3	$\Phi^*$	3.15	1.54
		BOL	449	0
		$\gamma^*$	4.85	1.75
0.3	4	$\Phi^*$	3.43	2.73
		BOL	737	298
		$\gamma^*$	2.58	1.70
0.3	3	$\Phi^*$	5.84	2.50
		BOL	2139	184
		$\gamma^*$	6.66	2.29

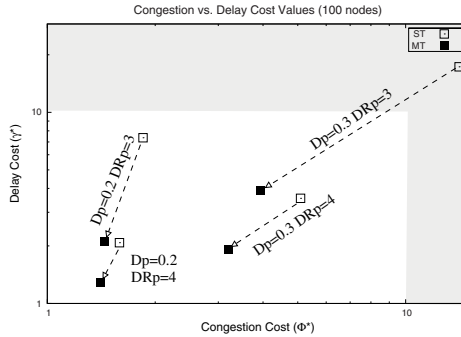
**Table 4.** Results for congestion and delay optimization - network with 100 nodes

Demands ( $D_p$ )	Delays ( $DR_p$ )	Metric	ST	MT
0.2	4	$\Phi^*$	1.59	1.41
		BOL	0	0
		$\gamma^*$	2.08	1.28
0.2	3	$\Phi^*$	1.85	1.44
		BOL	15	0
		$\gamma^*$	7.36	2.10
0.3	4	$\Phi^*$	5.09	3.21
		BOL	2715	809
		$\gamma^*$	3.55	1.92
0.3	3	$\Phi^*$	14.1	3.93
		BOL	7437	1356
		$\gamma^*$	17.3	3.90

Taking into account the results of Figs. 5 and 6 some conclusions might be drawn. As expected, scenarios assuming lower levels of traffic demands and delay requirements (e.g.  $D_p = 0.2$  and  $DR_p = 4$ ) present better QoS results in all of the considered scenarios (both for the ST and MT approaches). In the same perspective, for harder assumptions of network demands (e.g.  $D_p = 0.3$  and  $DR_p = 3$ ) most of the results show an increase in the congestion and delay



**Fig. 5.** Results for congestion vs delay - network with a) 30 nodes and b) 80 nodes



**Fig. 6.** Results for congestion vs delay - network with 100 nodes

costs. In particular, in scenarios with high levels of traffic demands and delays, some of the ST results fall into the gray filled area of the figures, representing a network behavior with some degree of packet loss and/or delay violations.

In addition, the effectiveness of the MT approach can be clearly assessed observing the fact that results obtained from the MT approach are shifted to the lower left corner of the figures (black squares), thus with lower values for the congestion and delay cost functions when comparing to ST. It is also important to note that even providing low quality results when compared with the MT approach, the ST performance is still acceptable for a large part of the studied scenarios. This allows to conclude that under low or medium network demands the optimization framework presented in this work is able to find single topology network configurations assuring a good overall QoS performance in the domain.

Finally, it should be highlighted the quality of the MT results in all the scenarios, independently of the difficulty levels of the demands. This means that even if heavy traffic demands and strict multicast delay requirements are considered, acceptable QoS performances is obtained, corroborating the viability and effectiveness of the proposed optimization framework.

## 5 Conclusions

The optimization of OSPF weights brings important tools for traffic engineering, without demanding modifications on the basic network model. This work presented EAs for routing optimization in networks with unicast and multicast demands. Resorting to a set of network configurations and unicast/ multicast demands, it was shown that the proposed EAs were able to provide OSPF weights that can lead to good network behaviour, in terms of the overall network performance, as well as regarding multicast end-to-end delays.

The proposed approach was favourably compared to a 2-step optimization procedure, proposed in previous work, that leads to sub-optimal results in terms of network congestion and overloaded links. The advantages of using a multi-topology protocol in these scenarios were also studied and it was concluded that these are most advantageous when the network resources are limited.

The main contribution of this work is the capability of optimizing the OSPF weights considering all factors involved (i.e. all types of traffic). Using the proposed methods, the network administrator can decide if a multi-topology protocol is needed or simply use a standard implementation of OSPF.

## Acknowledgments

This work is supported by the Portuguese FCT project PTDC/EIA/64541/2006.

## References

1. Ahuja, R.K., Magnati, T.L., Orlin, J.B.: *Network Flows*. Prentice Hall, Englewood Cliffs (1993)
2. Awduche, D., Jabbari, B.: Internet traffic engineering using multi-protocol label switching (MPLS). *Computer Networks* 40, 111–129 (2002)
3. Davie, B., Rekhter, Y.: *MPLS: Multiprotocol Label Switching Technology and Applications*. Morgan Kaufmann, USA (2000)
4. Dijkstra, E.: A note on two problems in connexion with graphs. *Numerische Mathematik* 1, 269–271 (1959)
5. Ericsson, M., Resende, M.G.C., Pardalos, P.M.: A Genetic Algorithm for the Weight Setting Problem in OSPF Routing. *J. of Combinatorial Optimization* 6, 299–333 (2002)
6. Fortz, B., Thorup, M.: Internet Traffic Engineering by Optimizing OSPF Weights. In: *Proceedings of IEEE INFOCOM*, pp. 519–528 (2000)
7. Medina, A., Lakhina, A., Matta, I., Byers, J.: BRITE: Universal Topology Generation from a User’s Perspective. Technical Report 2001-003 (January 2001)
8. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd edn. Springer, USA (1996)
9. Moy, J.: *OSPF, Anatomy of an Internet Routing Protocol*. Addison Wesley, Reading (1998)
10. Psenak, P., Mirtorabi, S., Roy, A., Nguyen, L., Pillay-Esnault, P.: Multi-topology (MT) routing in OSPF. IETF RFC 4915 (June 2007)
11. Rocha, M., Sousa, P., Rio, M., Cortez, P.: Qos constrained internet routing with evolutionary algorithms. In: *Proc. IEEE Conference Evolutionary Computation*, pp. 9270–9277. IEEE Press, Los Alamitos (2006)
12. Thomas, T.M.: *OSPF Network Design Solutions*. Cisco Press (1998)
13. Wang, N., Pavlou, G.: Traffic Engineered Multicast Content Delivery Without MPLS Overlay. *IEEE Transactions on Multimedia* 9(3) (April 2007)