Presentation New Method to Increase Network QOS Using OSPF's Link Weight

Ali Abulhallaj
Department of Computer Science, Andimeshk Branch, Islamic Azad University, Andimeshk, Iran
hallaj@gmail.com

Abstract

Nowadays, the internet is used various functions such as Radio, Television, and Telephone. For appropriate usage, we need network solutions that aim to provide end-users with Quality of Service (QoS) support. The networking research community proposed distinct QoS aware architectures and specific traffic control mechanisms that provide distinct service levels to networked applications(Wang,2001).

Accordingly, Internet Service Providers (ISPs) must have Service Level Agreements (SLAs) (Liu et al, 2005) with their clients and with other ISPs. If neglected, must pay financial penalties. To achieve that, administrators must do some important configuration in their networks to assure that correct resource provisioning is achieved in the ISP domain. For example, these configurations may vary based on ISPs because The QoS solutions are different. Also it may includes distinct tasks such as, specifying the forwarding treatment given at each network nodes (e.g. the per-hop-behaviors(Davie et al, 2002)(Wroclawski et al,1999) of DiffServ (Blake,1998)). defining traffic classification and packet marking rules at network edges, configuring admission control based mechanisms(Tsuchiya,2003), configuring routing algorithms(Slattery et al,2008) or finer-grain QoS solutions, enabling signaling protocols allow per flow resource reservation(Zhang et al,1997). In addition, there is more than one solution to achieve QoS aware network infrastructure. In general, any solutions require number of components working together.

Keywords: QoS, OSPF, Evolutionary Algorithms, EA, Optimization.

I. Introduction

According to the above, some components have more importance in any QoS capable infrastructure. One of these components has the ability to control the data path followed by packets traversing a given domain. Intra-domain routing protocols and Multi-Protocol Label Switching (MPLS)(Awduche et al,2002)(Davie et al,2000) are two different strategies can use for this purpose. MPLS has some deficiency when it used in the context of packet switching. The first, it adds significant complexity to the IP model when comparing with the simplicity of some routing protocols. Flow state has to be stored in every router of the path. The second, MPLS failure recovery mechanisms are considered more complex than the typical router convergence ones. Finally, it represents a considerable network management overhead.

Open Shortest Path First (OSPF) is the most common is used intra-domain internet routing protocol (Moy, 1997) (Pillay et al, 2008) (Thomas, 2003). Traffic is routed along shortest paths, flow split at nodes where several outgoing links and they are on shortest paths until destination. The weights of the links, can be changed by the network operator which used to compute the best path from each sources to each destinations using the well-known Dijkstra algorithm (Dijkstra, 1959). The only way administrators can affect the network behavior is the setting of OSPF weights. This choice is very important and may have a major impact in the

network performance. However, the simple rules are used, such as setting the weights inversely proportional to the link capacity. This approach often leads to sub-optimal network resource utilization.

An ideal way to improve the process of OSPF weight setting is to implement traffic engineering. It assumed, the administrator access to a matrix representing traffic demands between each pair of nodes in the network. This was the approach taken by Fortz et al(Fortz et al, 2000). Where this task was viewed as an optimization problem, by defining a cost function that it measures the network congestion. The same authors proved that this task is a NP-hard problem and proposed some local search heuristics that compared well with the MPLS model. Also the use of Evolutionary Algorithms (EAs) causes to improve these results (Ericsson ,2002). Both alternatives were also combined to create hybrid EAs including local search operators that were able to improve these results (Buriol et al, 2005).

One of important tasks to implement QoS aware networking is delay based constraints. But previous achievements didn't attend it. In this work, Parent-centric differential evolution algorithm(Pant et al,2009) were used to calculate link-state routing weights that optimize traffic congestion and delay requirements at the same time. For providing a QoS aware optimization framework, a mathematical model of the problem is proposed that regarding both congestion and delay constraints as a cost function. The DEPCX optimization algorithms to reach the optimal OSPF weights for each network link will use this. The Proposed algorithm is compared to other approaches such as EAs. Local search, common heuristics and multipurpose optimization algorithms. In comparison, the DEPCX is shown good results and can be used as the best alternative for this task.

I. Problem description

A. General description

The main objective of the proposed optimization framework is to provide network administrators with efficient OSPF link configurations, taking into account the users demands, the topology and other features of a given network domain (see Figure.1). It is assumed that client demands are mapped into a matrix, summarizing, for each source and destination routers, the given amount of bandwidth and end-to-end delay are required to be supported by the network domain. For instance, there are several techniques such as how to obtain traffic demand matrices (Davy et al, 2006) (Medina et al, 2002) which providing estimations regarding the overall QoS requirements within the given network domain.

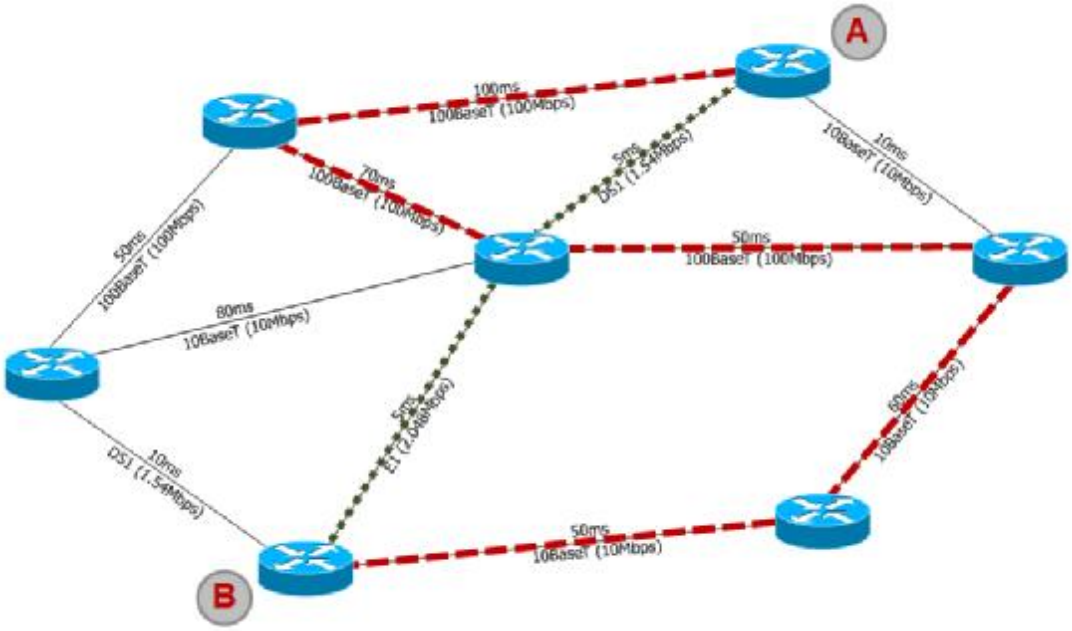

Figure.1. Example of an ISP network scenario

As can be seen the network scenario in Figure.1 and an individual demand between two network nodes (A and B). If this demand is expressed in terms of target delay, then the OSPF weight setting process should be able to provide data path with the minimum propagation delay between A and B (PATH ...). However, if delay requirements are not considered and the only constraint between A and B is given bandwidth requirement such as 9Mbps, then the OSPF setting process should try to minimize the network congestion and assign OSPF weights to force a data path inducing the lowest level of losses in the traffic (PATH ---). Considering now that a given demand has simultaneously bandwidth and delay constraints, then it is expected that the OSPF weight setting process should try to find data path representing a trade-off between the bandwidth and delay metrics. In addition, if one considers that after studying the QoS demands of the network domain users. Therefore, each routers pair of the domain may have specific multi constrained QoS requirements (i.e. congestion vs. delay demands), so it is easy to understand the complex nature of the problem of obtaining OSPF weights able to optimize the QoS levels of a given network domain.

B. Problem formulation

The general routing problem (Awduche et al, 2002), represents routers and transmission links by the set of nodes (N) and the set of arcs (A) in directed graph G = (N, A). In This model, c_a represents the capacity of each links $a \in A$. Additionally, a demand matrix D is available, where each element d_{st} represents the traffic demand between each pair of nodes s and t from N. It is assumed 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 load on each arc a (l_a) can be defined the following:

August.2013-Volume 1.No1
ISSN 2345-3397

$$l_a = \sum_{(s,t)\in N\times N} f_a^{st} \quad (1)$$

While the link utilization rate u_a is given by:

$$u_a = \frac{l_a}{c_a} \quad (2)$$

It is possible to define a congestion measure for each link ($\Phi_a = p(u_a)$)proposed by Fortz and Thorup(Fortz et al ,2000). It is depicted in Figure 1 where p is a penalty function p that it has small penalties for values near zero. However, as values approach the unity it becomes more expensive and exponential penalizes values above 1.

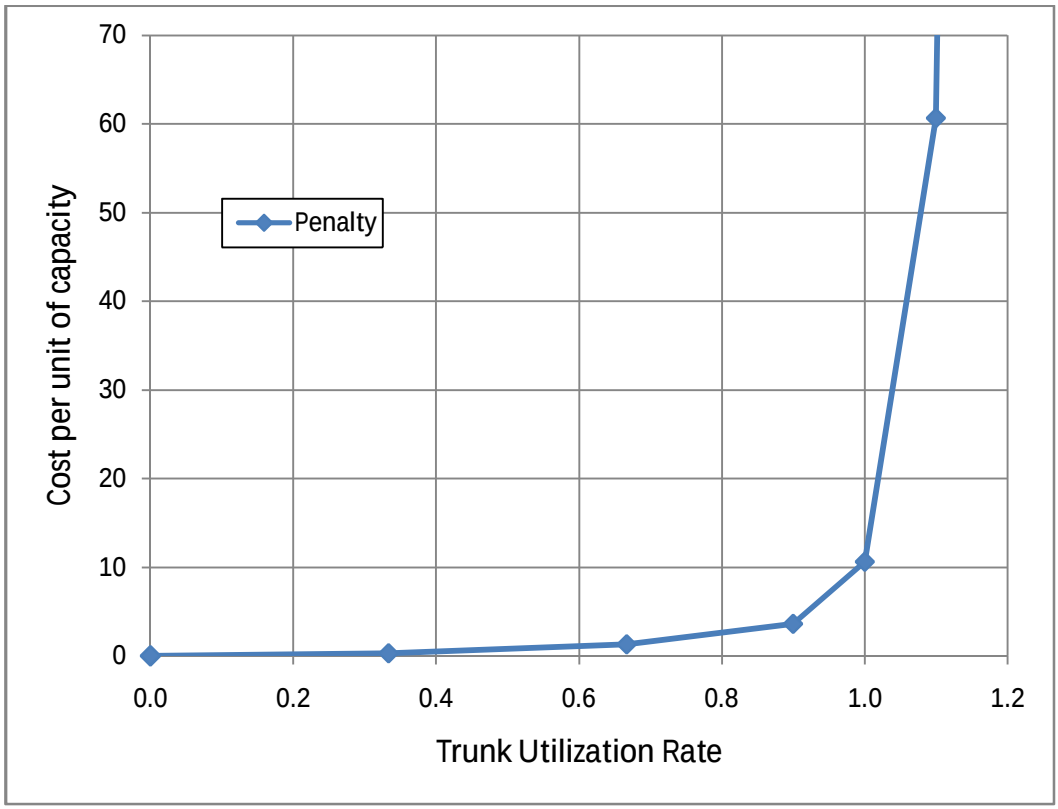

Figure.2. Piecewise linear function $F_a(l_a)$

$$p(x) = \begin{cases} x, & x \in [0, 1/3) \\ 3x - \frac{2}{3}, & x \in [1/3, 2/3) \\ 10x - \frac{16}{3}, & x \in [2/3, 9/10) \\ 70x - \frac{178}{3}, & x \in [9/10, 1) \\ 500x - \frac{1468}{3}, & x \in [1, 11/10) \\ 5000x - \frac{16{,}318}{3}, & x > 11/10 \end{cases} \quad (3)$$

Under this framework, it is possible to define a linear programming instance (Fortz et al ,2000) where the purpose is to set the value of the variables f_a^{st} that minimize the following objective function:

$$F = \sum_{a \in A} F_a \quad (4)$$

Subject to:

$$\sum_{u:(u,v)\in A} f_{(u,v)}^{(s,t)} - \sum_{u:(v,u)\in A} f_{(u,v)}^{(s,t)} = \begin{cases} -d_{st}, & if\ v = s \\ d_{st}, & if\ v = t, \\ 0\ , & otherwise \end{cases} \quad v, s, t \in N \quad (5)$$

$$l_a = \sum_{(s,t)\in N \times N} f_a^{st}, \quad a \in A \quad (6)$$

$$F_a \geq l_a, \quad a \in A \quad (7)$$

$$F_a \geq 3l_a - \frac{2}{3c_a}, \quad a \in A \quad (8)$$

$$F_a \geq 10l_a - \frac{16}{3c_a}, \quad a \in A \quad (9)$$

$$F_a \geq 70l_a - \frac{178}{3c_a}, \quad a \in A \quad (10)$$

$$F_a \geq 500l_a - \frac{1468}{3c_a}, \quad a \in A \quad (11)$$

$$F_a \geq 5000l_a - \frac{16318}{3c_a}, \quad a \in A \quad (12)$$

$$f_a^{(s,t)} \geq 0, \quad a \in A, s, t \in N \quad (13)$$

Where d_{st} means the value of D in rows and column t. Constraints (5) represent flow conservation in the network, ensuring the desired traffic flow is routed from source s to

destination t. Constraints (6) define the way load is calculated on each arc. Finally, constraints (7) – (12) define the penalties (cost) on each arc. The following, the optimal solution to problem is denoted by Φ_{Opt}.

The OSPF, all arcs are associated with an integer weight. Every node uses weights in the Dijkstra algorithm(Dijkstra , 1959) to calculate the shortest paths to all other nodes in the network, where each of these paths has a length equal to the sum of its arcs. All the traffic from a given source to a destination transmits along the shortest path. If there are two or more paths with the same length, between a given source and destination, traffic is evenly divided among the arcs in these paths (load balancing)(Moy,1998). It is assumed a given solution i.e. a weight assignment (w) and the corresponding utilization rates on each arc (u_a). In this case, the total routing cost is expressed by:

$$\Phi(w) = \sum_{a \in A} F_a(w) \tag{14}$$

For the loads and parallel penalties ($\Phi_a(w)$) is calculated based on the given OSPF weights. In this way, the OSPF weight-setting problem is equivalent to finding the optimal weight values for each link (w_{opt}), in order to minimize the function $\Phi(w)$. The congestion measure can be normalized over distinct topology scenarios that it divides by a scaling factor defined as:

$$F_{UNCAP} = \sum_{(s,t) \in N \times N} d_{st} h_{st} \tag{15}$$

Where h_{st} is the minimum hop count between nodes s and t. Finally, the scaled congestion measure cost is defined as:

$$F^*(w) = \frac{F(w)}{F_{UNCAP}} \tag{16}$$

and the following relationships hold(Fortz et al ,2000):

$$1 \leq F^*_{opt} \leq F^*(w_{opt}) \leq 5000 \tag{17}$$

It is important to note that when Φ^* equals 1, all loads are below 1/3 of the link capacity. In the case when all arcs are exactly full the value of Φ^* is 10 2/3. This value will be considered as threshold that bounds the acceptable working region of the network. To enable an enlarged set of QoS constraints, an extension to this model is proposed in this work. This enrichment allows the inclusion of delay requirements for each pair of routers in the network. These are modeled as a matrix DR, that for each pair of nodes (s, t) $\in N \times N$ (where $d_{st} > 0$) gives the delay target for traffic between origin s and destination t (denoted byDR_{st}). A cost function was developed to evaluate the delay compliance for each scenario (a set of OSPF weights). This function takes into account the average delay of the traffic between the two nodes ($Delay_{st}$), the value calculated by considering all paths between s and t with minimum cost and averaging delay for any paths.

It was considered the scenarios where this work would be applicable, the delay in each path is dominated by the component given by propagation delays in its arcs, and that queuing delays can be neglected. However, if it required, queuing delays can be introduced in the model by approximating its values resorting to queuing theory(Bolch,2006) and taking into account the following parameters at each node. The capacity of the corresponding output links, their utilization rates, and more specific technical parameters such as the mean packet size and the overall queue size associated with each link.

The delay compliance ratio for a given pair (s, t) ∈N × N is as following:

$$\widetilde{Del}_{st} = \frac{Delay_{st}}{DR_{st}} \tag{18}$$

A penalty for delay compliance can be calculated using function p. Therefore, the τ_{st} function is defined according to the following:

$$\tau_{st} = p(\widetilde{Del}_{st}) \tag{19}$$

This, in turn, allows the definition of a delay minimization cost function, given a set of OSPF weights (w):

$$\tau(w) = \sum_{(s,t)\in N\times N} \tau_{st}(w) \tag{20}$$

where the τ_{st} (w) values represent the delay penalties for each end-to-end path, the given routes determined by the OSPF weight set w.

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 min $Delay_{st}$ is calculated as the delay of the path with minimum possible overall delay):

$$\tau^*(w) = \frac{\tau(w)}{\sum_{(s,t)\in N\times N} \underline{Del}_{st}} \tag{21}$$

It is possible to define the optimization problem addressed the work that is clearly multi-objective. Indeed, given a network represented by a graph G = (N, A), a demand matrix D and a delay requirements matrix DR, the aim is to find the set of OSPF weights (w) that simultaneously minimizes the functionsΦ^*(w) and τ^*(w). When a single objective is considered the cost of a solution w is calculated using functions Φ^*(w) for congestion and τ^*(w) for delays. For multi-objective optimization, all algorithms described in the following section use a linear weighting scheme where the cost of the solution is given by:

$$f(w) = \alpha F^*(w) + (1-\alpha)\tau^*(w), \qquad \alpha \in [0,1] \tag{22}$$

This scheme can be effective since both cost functions are normalized in the same range. The parameter can be used to tune the trade-off between both components of the cost function.

II. Algorithms for OSPF weight setting

A. Differential Evolution

In this algorithms[15], we took only three parent vectors $\vec{a} \neq \vec{b} \neq \vec{c}$, such that $\vec{a} = \overrightarrow{X_{best,G}}$ and $\vec{b}$ and $\vec{c}$ are randomly selected from the rest of the population.$\overrightarrow{X_{best,G}}$represent the point with best fitness function value. The reason for selecting the point with best fitness function value is to create an elite type model so that the probability of getting an offspring near the best parent increases. We have considered the generation of a single offspring approximately the best parent vector. In the proposed algorithms, the mutant vector is generated as follows:

$$V_{i,G+1} = \overrightarrow{X_{best,G}} + w_c d_p + \sum_{i-p}^{u}{}_{i-1} w_n \overline{D}, \overrightarrow{e_i} \quad (23)$$

In the next phase that is reproduction, a new candidate for the next generation is produced. The newly generated offspring will have the genes of both the parents. Since one of the parents is the one having the best fitness function value found so far, the probability of the offspring getting good genes increases. Pseudo code of the ProDEPCX is:

TABLE I
PSEUDOCODE OF PRODEPCX

Algorithm: pseudo code of ProDEPCX
1. Randomly initialize the population PG
2. REPEAT Until search converged
3. REPEAT for each individual I∈PG
4. Mutation operation:
4.1 Perform mutation using equation(5) with a probability PD, or
4.2 Perform original mutation with Equation (1) (with probability 1−PD).
5. Crossover operation.
6. Evaluation of the function.
7. Selection.
8. ENDREPEAT
9. ENDREPEAT.

The advantage of parent-centric approach is increasing the probability of the offspring to be generated near the parent. It is self-adaptive type approach where a new solution vector keeps moving towards the optimum systematically. If the parents are located close to each other, the offspring are located densely around the parents whereas if the parents are located far from each other, the offspring are sparsely located. Since the parents are selected at random and offspring are generated around them, also it maintains the diversity of the population.

Since OSPF weights are integer, it is necessary to round the values used in the DE before the evaluation. In the experiments, the population size was 20.

B. Evolutionary Algorithms

Evolutionary Algorithms (EAs) (Tsuchiya, 2003) can address the problems defined in the previous section. In the EA, each individual encodes a solution as a vector of integer values, where each value (gene) corresponds to the weight of a link (arc) in the network (the values range from 1 to wmax). Therefore, the size of the individual equals the number of links in the

network. The individuals in the initial population are generated randomly, with the arc weights taken from a uniform distribution.

In order to create new solutions, several reproduction operators were used, more specifically two mutation and one crossover operator:

- Random mutation, replaces a given gene by a random value, within the allowed range.
- Incremental/decremental mutation, replaces a given gene by the next or previous value (with equal probabilities) within the allowed range.
- Uniform crossover, a standard crossover operator (Zhang et al, 1997) is suit when the order of the variables in the individual (solution) is not important. This operator works by taking two parents as input and generating two offspring. For each position in the genome, a binary variable is randomly generated: if its value is 1, the first offspring takes the gene from the first parent in that position, while the second offspring takes the gene from the second parent; if the random value is 0, the roles of the parents are reversed.

The proposed EA, whenever a new individual needs to be creation, one of the previous reproduction operators is selected, with equal probabilities. Unlike traditional EAs, only one operator is applied to create each new individual. In this context, the crossover internal probability is set to 1, meaning that when it is used the crossover will always be applied. The mutation operators always use an internal probability of 0.01, which means that when they are used to create offspring, they will modify in average 1% of the genes in the parent.

The overall structure of the EA is given by:

(1) Generate and evaluate the initial population (P0).
(2) While the termination criteria is not met:
  (i) Select from Pt individuals for reproduction.
  (ii) Apply the reproduction operators to breed the offspring and evaluate them.
  (iii) Insert the offspring into the next population (Pt+1).
  (iv) Select the survivors from Pt to be kept in Pt+1.

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 kept from the previous generation, and 50% are bred by the application of the reproduction operators. In the experiments, a population size of 100 individuals was considered.

C. Local search

A local search (LS) scheme was devised to improve the quality of a solution and works as follows: taking a set of weights wi , a link is randomly selected to start the process. Firstly, it increases the value of this weight by 1, if this leads to a better solution. This process is repeated while the solution improves. If the first increase operation did not lead to better solution, the decrease is tried and repeated while the solution improves. The process is

repeated for the next position, until all positions have been tested. The overall process is repeated while the solution improves.

Based on LS operator was devised a multi-start LS (MS-LS) algorithm. It starts with a random solution. Therefore, it applies the LS operator. This process is repeated and the best solution found is kept. The process is terminated when a maximum number of solutions have been evaluated.

D. Heuristic methods

A number of heuristic methods were implemented to assess the order of magnitude of the improvements obtained by the proposed methods when it is compared with the traditional weight setting heuristics:

- InvCap – sets each link weight to a value inversely proportional to its capacity.
- L2 – sets each link weight to a value proportional to the Euclidean distance.
- Random – a number of randomly generated solutions are analyzed and the best is selected.

III. Experiments and results

A. Simulation

To evaluate the proposed algorithms, a number of experiments were conducted. We use Level3 as experimental platform to this work and it is presented in figure 3.

August.2013-Volume 1.No1
ISSN 2345-3397

Figure 3: Level3 network backbone in USA

All simulation was made by OPNET academic version. A set of 3 networks was created, varying the number of nodes (N = 7, 32, 56). This resulted in networks ranging from 11 to 64 links. The link bandwidth was based on real model and between 1 and 10 Gbits/s.

Next, the demand and delay constraints matrices (D and DR) were generated. For each network, a set of three distinct D and DR matrices were created. A parameter (Dp) was considered, representing the expected mean of congestion in each link (values for Dp in the experiments were 0.1, 0.2 and 0.3). For DR matrices, the strategy was to calculate the average of the minimum possible delays, over all pairs of nodes.

The termination criteria was the same for all optimization algorithms (EAs, DEPCX and MS-LS) consisting in the maximum number of solutions evaluated. In all cases, w_{max} was set to 20. For all the stochastic algorithms, 10 runs were executed in each case.

The results are divided into two sets based on the cost function used. The first considering a single objective cost function, for the optimization of network congestion. The latter considers the case of simultaneous optimization of congestion and delays. In all figures presented in this section, the data was plotted in a logarithmic scale, given the exponential nature of the penalty function adopted.

B. Congestion

Since the number of experiments is quite high, it was decided to show aggregate results that can be used to draw conclusions. Table shows the results for all the networks, averaged by the demand levels (Dp), including in the last line the overallmeanvalue. It is clear that the results get worse with the increase of Dp, as would be expected. Figure plots the same results in a graphical way, showing in the white area the acceptable working region, i.e. the congestion requirements are satisfied, whereas an increasing level of grey is used to identify regions with increasing levels of service degradation.

TABLE II
RESULTS FOR THE OPTIMIZATION OF CONGESTION (Φ*)—AVERAGED BY DEMAND LEVELS.

D_p	ProDEPCX	EA	MS-LS	InvCap	L2	Random
0.1	1.05	1.06	1.15	1.6	214	69.1
0.2	1.14	1.21	1.62	59.9	765	465
0.3	1.59	1.84	6.68	337	1273	872
Average	1.26	1.37	3.15	132.83	750.67	468.70

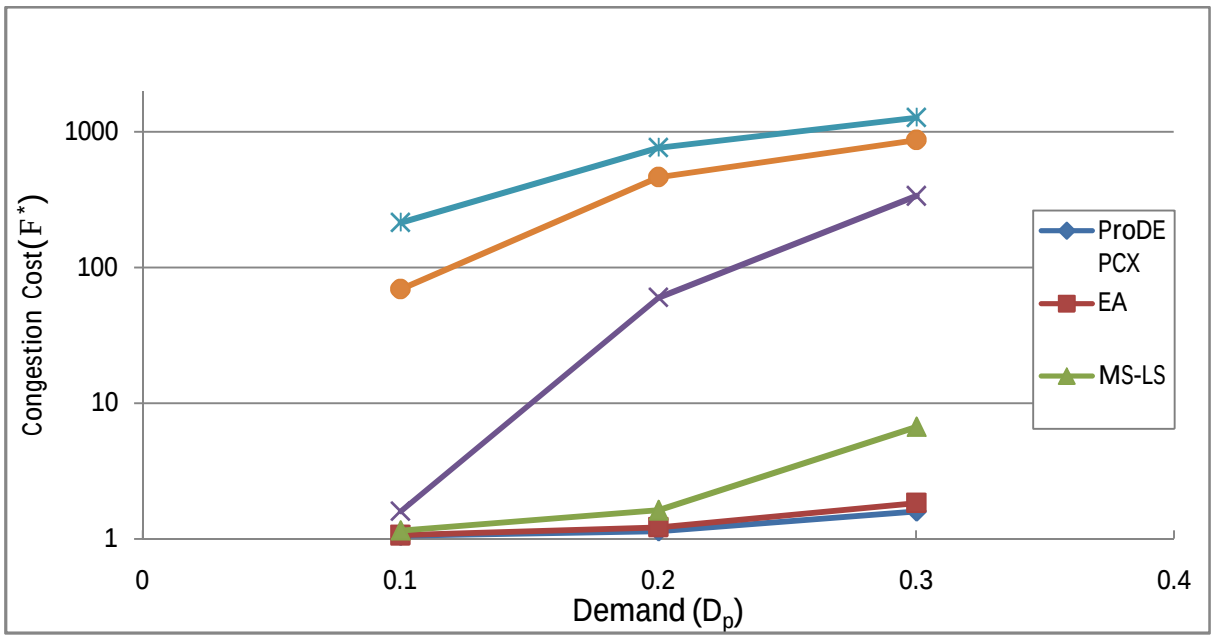

Figure .4. Graphical representation of the results obtained by the different methods in congestion optimization (averaged by Dp).

The comparison between the methods shows a superiority of the ProDEPCX that achieves solutions which manage a very reasonable behavior in all scenarios. The heuristics manage very poorly even InvCap, quite it is used in practice and gets poor results when Dp is 0.2 or 0.3, which means that the optimization with the ProDEPCX assures good network behavior in scenarios where demands are at least 200% larger than the ones where InvCap would assure similar levels of congestion. The results of EA and MS-LS are acceptable but significantly worse than the ones obtained by the ProDEPCX and the gap increases with larger values of Dp.

C. Simultaneous optimization of congestion and delays

In this section, the results for the simultaneous optimization of congestion and delays, using a linear weight combination of the objectives are discussed. The results are presented in terms of the values for the two objective functions ($\Phi^*$and τ^*), since the value of f for these solutions can be easily obtained and is not relevant to the analysis. In a first stage, only the value of $\alpha = 0.5$ will be considered, thus considering each aim to be of equal importance. Table shows the results averaged by the demand level (Dp) for all the algorithms. From the table, it is clear that the ProDEPCX outperforms all other algorithms, followed by the EA and MS-LS. The heuristics behave quite badly, when both aims are taken into account. Since, in the heuristic methods, the solution is built disregarding the cost function, the results for multi-objective optimization only change due to the increase in the Dp parameter. The onlyweackness of ProDEPCX is when traffic is very low that EA may better.

TABLE III

RESULTS FOR THE SIMULTANEOUS OPTIMIZATION OF CONGESTION AND DELAYS—AVERAGED BY DP, WITHα = 0.5.

D	ProDEPCX		EA		MS-LS		InvCap		L2		Random	
	F^*	t^*	F^*	t^*	F^*	t^*	F^*	t^*	F^*	t^*	F^*	t^*
0.1	1.18	2.04	1.19	1.91	1.65	3.99	1.4	253	214	1.73	84	109
0.2	1.42	2.23	1.51	2.38	3.31	8.21	53.6	253	765	1.73	469	140
0.3	2.29	2.87	2.67	3.49	14.8	15.63	314	253	1273	1.73	941	151
Average	1.63	2.38	1.79	2.59	6.59	9.28	123	253	751	1.73	498	133

A different view is offered by Figure where the results are plotted with the two objectives in each axis. The results averaged by Dp are shown. In these graph, the good overall network behavior of the solutions provided by the ProDEPCX is clearly visible, regarding the network behavior in terms of congestion and delays, and when compared to all other alternative methods. It is easy to see that no single heuristic is capable of acceptable results in both aims simultaneously. L2 behaves well in the delay minimization but fails completely in congestion; InvCap is better on congestion but fails completely in the delays. The EA gets results that are in an acceptable range, but are always significantly worse than those of the ProDEPCX when traffic increase, and MS-LS does not manage good results when the problem instances get harder.

To study the impact of the parameter α, three distinct values were tested: 0.25, 0.5 and 0.75. The value of 0.5 considers each aim to be of equal importance, while the 0.25 favors the minimization of delays and the 0.75 will give more weight to congestion. In this analysis, only the ProDEPCX and the EA (the two best algorithms in the previous case) will be considered. In Table IV, the results obtained were summarized averaging by the parameter α. The first column represents the parameter α; the next two indicate the results for the ProDEPCX algorithm and, finally, the last two give the results of the EA for both congestion and delays, each with an extra information indicating the percentage by which this results exceed the ones obtained by the corresponding algorithm under a single objective cost function.

TABLE IV
OVERALL RESULTS FOR SIMULTANEOUS OPTIMIZATION OF CONGESTION AND DELAYS—AVERAGED BY α

α	ProDEPCX		EA	
	F^*	t^*	F^*	t^*
0.25	1.92	2.29	2.06	2.34
0.5	1.63	2.38	1.79	2.59
0.75	1.57	2.76	1.62	2.97

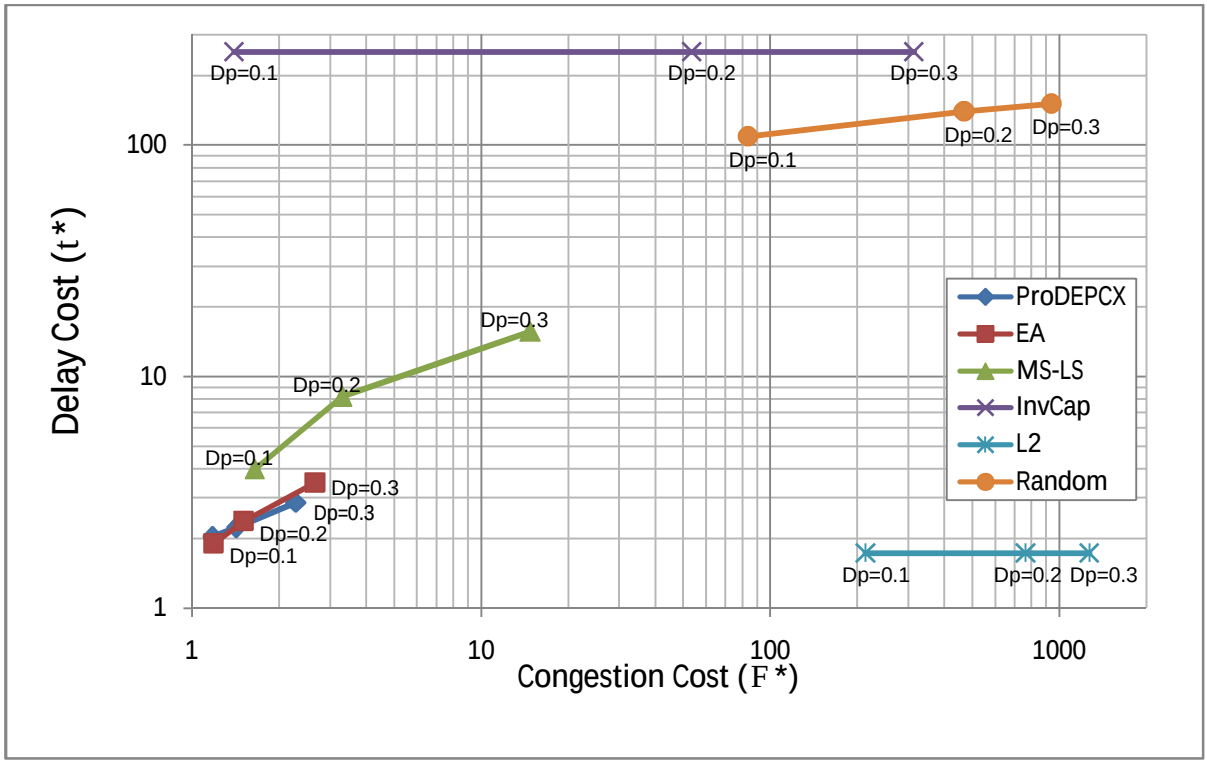

Figure .5. Graphical representation of the results obtained by the different methods in the multi-objective optimization with α = 0.5: averaged by Dp

The results is shown in this table make clear the effect of parameter α, once it is possible to observe different trade-offs between the two objectives. Indeed, when α increases the results on congestion improve, while the reverse happens to the delay minimization. The ProDEPCX provides smoother changes being able to offer better results with all configurations. In particular, the intermediate value of α (0.5) provides a good compromise between the two objectives. In this case, the overall results show decrease in the congestion performance and the delays minimization, both when comparing to single objective optimization. These values are quite good, since in this case both aims have to obey simultaneously even if they are contradictory. In fact, a decrease in the performance, when compared to single objective optimization would always be expected. If the absolute average

values for both cost functions are taken into account this indicates a quite acceptable network performance, well within the defined working region.

TABLE V
RESULTS FOR THE SIMULTANEOUS OPTIMIZATION OF CONGESTION AND DELAYS—AVERAGED BYTHE NUMBER OF NODES
($\alpha = 0.5$).

Nodes	ProDEPCX		EA	
	F^*	t^*	F^*	t^*
7	1.65	2.27	1.75	2.39
32	1.56	2.41	1.83	3.12
56	1.69	2.45	1.78	2.25

Table confirms the good scalability properties of both the ProDEPCX and the EA. The results are almost constant for the different network sizes (in this case, measured by the number of nodes).

IV. Conclusions and further work

The optimization of OSPF weights brings important tools for traffic engineering, without demanding modifications on the basic network model. This work presented Evolutionary Computation approaches for multi-objective routing optimization in the Internet. Resorting to a set of network configurations that each constrained by bandwidth and delay requirements, it was shown that the proposed Parent Centric Differential Evolution Algorithms (ProDEPCX) were able to provide OSPF weights that can lead to good network behavior. The performance of ProDEPCX was compared with other algorithms (Evolutionary Algorithms, local search, heuristics) clearly showing its superiority.

The proposed optimization framework, although requiring some computational effort, can be achieved in useful time (e.g. for Level3 network with 56 nodes, a good solution can be obtained by the ProDEPCX in less than half an hour). If very distinct traffic profiles occur in around day, the corresponding matrices can be used to optimize distinct OSPF weights. Furthermore, the adaptation to new solution is always faster than running from scratch, since a good solution is available to boost the search and therefore any change that demands re optimization will be achieved rapidly. Obviously, we can say that the proposed framework could be implemented in a straightforward way in a real world scenario.

One important topic for future work is the integration of distinct classes of QoS demands in the proposed optimization model. On this topic, the Internet Engineering Task Force (IETF) has proposed standards on Multi-topology Routing aiming at providing different paths for different types of traffic.

References

Awduche, D.O.Jabbari, B., 2002, Internet traffic engineering using multi-protocol label switching (MPLS). Computer Networks, **40**(1): p. 111-129.
Blake, S., et al., 1998, An architecture for differentiated services.
Bolch, G., 2006, Queueing networks and Markov chains: modeling and performance evaluation with computer science applications: Wiley-Blackwell.
Buriol, L., et al., 2005, A hybrid genetic algorithm for the weight setting problem in OSPF/IS-IS routing. Networks, **46**(1): p. 36-56.
Davie, B., et al., 2002, An expedited forwarding PHB (per-hop behavior).
Davie, B.S.Rekhter, Y., 2000, MPLS : technology and applications San Francisco: Morgan Kaufmann Publishers. xiv, 287 p.
Davy, A., Botvich, D.Jennings, B., 2006, An efficient process for estimation of network demand for QoS-aware IP network planning. Autonomic Principles of IP Operations and Management: p. 120-131.
Dijkstra, E.W., 1959, A note on two problems in connexion with graphs. Numerische mathematik, **1**(1): p. 269-271.
Ericsson, M., Resende, M.G.C.Pardalos, P.M., 2002, A genetic algorithm for the weight setting problem in OSPF routing. Journal of Combinatorial Optimization, **6**(3): p. 299-333.
Fortz, B.Thorup, M., 2000, Internet traffic engineering by optimizing OSPF weights. IEEE.
Liu, Y., Tham, C.K.Jiang, Y., 2005, Conformance analysis in networks with service level agreements. Computer Networks, **47**(6): p. 885-906.
Medina, A., et al., 2002, Traffic matrix estimation: Existing techniques and new directions. ACM.
Moy, J., 1997, OSPF version 2.
Moy, J.T., 1998, OSPF: anatomy of an Internet routing protocol: Addison-Wesley Professional.
Pant, M., Ali, M.Singh, V., 2009, Parent-centric differential evolution algorithm for global optimization problems. Opsearch, **46**(2): p. 153-168.
Pillay-Esnault, P.Lindem, A., 2008, OSPFv3 Graceful Restart.
Slattery, T., Burton, W.Ebrary, I., 2000, Advanced IP routing in Cisco networks: McGraw-Hill.
Thomas, T., 2003, OSPF network design solutions. Recherche, **67**: p. 02.
Tsuchiya, T., 2003, Call admission control with QoS class modification. IEICE transactions on communications, **86**(2): p. 682-689.
Wang, Z., 2001, Internet QoS: Architectures and mechanisms for quality of service: Morgan Kaufmann Publishers.
Wroclawski, J., et al., 1999, Assured Forwarding PHB Group.
Zhang, L., et al., 1997, Resource reservation protocol (RSVP)--Version 1 functional specification. Resource.