

Traffic Engineering Approaches Using Multicriteria Optimization Techniques

Pedro Sousa¹, Paulo Cortez³, Miguel Rio², and Miguel Rocha¹

¹ Dept. of Informatics, University of Minho, Portugal
`{pns,mrocha}@di.uminho.pt`

² Dep. of Electronic and Electrical Engineering, UCL, UK
`m.rio@ee.ucl.ac.uk`

³ Dep. of Information Systems, University of Minho, Portugal
`pcortez@dsi.uminho.pt`

Abstract. Nowadays, network planning and management tasks can be of high complexity, given the numerous inputs that should be considered to effectively achieve an adequate configuration of the underlying network. This paper presents an optimization framework that helps network administrators in setting the optimal routing weights of link state protocols according to the required traffic demands, contributing in this way to improve the service levels quality provided by the network infrastructure. Since the envisaged task is a NP-hard problem, the framework resorts to Evolutionary Computation as the optimization engine. The focus is given to the use of multi-objective optimization approaches given the flexibility they provide to network administrators in selecting the adequate solutions in a given context. Resorting to the proposed optimization framework the administrator is able to automatically obtain highly optimized routing configurations adequate to support the requirements imposed by their customers. In this way, this novel approach effectively contributes to enhance and automate crucial network planning and management tasks.

Keywords: Traffic engineering; Network design and network planning.

1 Introduction

Recently, several types of applications have been integrated over IP converged networks, increasing the requirements on the ability to provide adequate service levels. In order to achieve a generalized capable QoS infra-structure, many different Quality of Service (QoS) solutions and associated traffic control mechanisms were proposed, such as the examples of traffic prioritization and selective resource reservation solutions [1]. However, the fast growth of Internet users and events such as the proliferation of multimedia contents and the increasing use of P2P applications are being responsible for the change of traffic profiles in the Internet. Moreover, the heterogeneity of current Internet access networks including wireless technologies such as Wi-Fi, WiMAX, among others, along with the

mobility patterns of some users also pose new challenges to the wired supporting infrastructures, as regards to their correct planning and configuration.

In this context, it is crucial that network administrators carefully consider the traffic demands required to be supported by the network infra-structure during a given period of time, such as specific days of a week, particular periods during the year, seasonal intervals, etc. This information can be provided using distinct techniques, such as forecasting methods based on historical data, traffic accounting based mechanisms, demand matrix estimation techniques [6] [7] [8], among many others. Such inputs are of extreme importance to be used to enhance the provision of adequate QoS levels by the network. However, in general, achieving reasonable service quality requires several components of the network infrastructure working in a coordinated way. Moreover, and irrespective of the QoS mechanisms in place, there are other factors which also play a crucial role on the networking performance, such as the routing configuration of the network.

The optimization framework proposed in this work focuses on the Open Shortest Path First (OSPF) intra-domain routing protocol, an extremely popular and ease of implementation protocol [10]. In this protocol, the administrator sets weights to every link in the network, which then are used to compute the best paths from each source to destination pair, resulting on the nodes' routing tables [3]. This process has a major impact on the network performance. However, in practice, simple methods and heuristics are commonly used, but often leading to sub-optimal network resource utilization. To address these issues, innovative approaches were taken, many of them inspired by the work of Fortz et al. [2], where the OSPF weight setting process is implemented using Traffic Engineering (TE) techniques. Here, the routing configuration task is formulated as an optimization problem, by defining a cost function that measures the network congestion, assuming that the administrator has access to a matrix representing traffic demands between each pair of source and destination nodes of the network.

In previous work, the authors have enhanced such optimization efforts, proposing a new approach [4] [5] that also accommodates delay constraints, also crucial to implement QoS aware networking services. In such works, optimization algorithms were used to calculate link-state routing weights that optimize traffic congestion, while simultaneously complying with specific delay requirements. However, the objective of developing network management tools that could effectively help network administrators requires that more powerful, efficient and versatile optimization paradigms be used to underpin this complex optimization problem. Within this perspective, this work fosters the research efforts and the application of TE methods resorting to the use of Multiobjective Evolutionary Algorithms (MOEAs) as a means to provide near-optimal routing configurations able to be easily analyzed by the administrator and, if required, applied to the network environment. The devised optimization framework takes advantages of the nature of the MOEAs solutions, assuming the form of Pareto fronts which are, both from the administrator and the optimization engine perspectives, very appropriate to deal with this NP-hard multi-constrained problem.

The paper proceeds with Section 2 giving the description of the problem and explaining the devised optimization model; Section 3 describes the experimental platform and the corresponding illustrative results; Section 4 discusses computational issues and software availability; Section 5 presents the conclusions.

2 Optimizing OSPF Routing Configurations

2.1 Problem Definition

The optimization framework proposed in this work aims to provide network administrators with efficient OSPF link weights, taking into account the users demands, the network topology and other features of the network domain. In OSPF, all links are associated with an integer weight. Every node uses these weights as an input to the Dijkstra algorithm [3] to calculate the shortest paths to all other nodes. All the traffic from a given source to a destination travels along the shortest path, except when two or more paths have the same length, in that case, traffic is divided among the arcs in these paths (load balancing) [12]. The framework assumes that client demands are mapped into a matrix¹ summarizing, for each source/destination edge router pair, a given required bandwidth and, if also defined, a target edge-to-edge delay to be supported by the network domain.

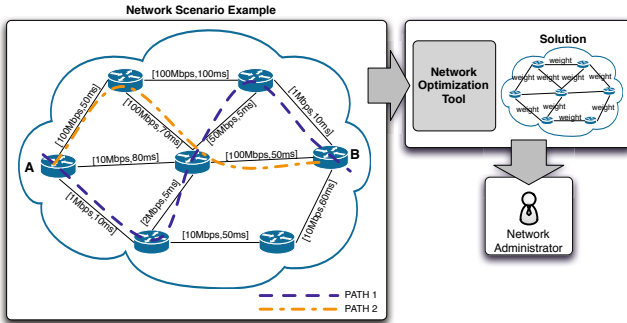


Fig. 1. Example of a network scenario

As an illustrative example, consider the network scenario included in Figure 1, involving only a single demand between the network $node_A$ and $node_B$. If such demand reflects only a delay target, the administrator should be able to use OSPF weights that result in a data path with the minimum delay between such nodes, i.e $path_1$. In a different perspective, if no delay requirements are assumed, and the unique constraint between such nodes is a given bandwidth target, the administrator would try to minimize the network congestion and assign OSPF weights to force a data path inducing the lowest level of losses in the traffic ($path_2$). These two distinct optimization aims would result in two

¹ There are several techniques to obtain traffic demand matrices which provide estimations regarding the overall requirements within a given domain (e.g. [6] [7]).

distinct sets of OSPF weights. Additionally, also considering that a given demand may have simultaneously bandwidth and delay constraints, then it is expected that the OSPF weight optimization process try to find a data path representing a trade-off between such requirements. The example of Figure 1 is, intentionally, extremely simple and only one demand was considered in the network domain. Assuming now that each router pair of a given Internet Service Provider (ISP) may have specific multi-constrained QoS requirements (i.e. traffic demands vs. delay restrictions), it is easy to understand how complex the problem can get, with the need of obtaining OSPF settings able to optimize multiple parameters.

The formulation model sustaining the proposed optimization framework follows the traditional mathematical networking model, representing routers and transmission links by a set of nodes (N) and arcs (A), respectively, in a directed graph $G = (N, A)$ [9], with each arc having a specific bandwidth capacity and a propagation delay, both intrinsic in the network topology². Additionally, a demand matrix is available (D), where each element represents the traffic demand between a pair of nodes, allowing to calculate the total load on each arc. This value is used to define a congestion measure for each link, that can be used to compute a penalty function that exponentially penalizes high values of congestion (e.g. a function such the one used in [2]). The framework was also enriched with the possibility of including delay requirements for each pair of routers in the network. These are modeled as a matrix (DR) that, for each pair of nodes, gives the delay target for traffic between the origin and destination. In a similar way, a cost function was defined to evaluate the delay compliance for each scenario. This, in turn, allowed the definition of a delay minimization cost function. One illustrative example of the type of optimization problems addressed in our framework is to find the set of OSPF weights that simultaneously minimizes the cost functions associated with network congestion and with the edge-to-edge delay penalties of the network domain, making the problem addressed in this work clearly multi-objective. In the proposed model, given a specific network topology, a traffic demand matrix (D) and a delay requirements matrix (DR), the objective is to achieve a set of OSPF weights (w) that simultaneously minimize the functions $\Phi^*(w)$ and $\gamma^*(w)$, which are the penalty functions for congestion and edge-to-edge delays, respectively³. In this context, the cost of associated with a given weight solution w is evaluated using functions $\Phi^*(w)$ for congestion and $\gamma^*(w)$ for delays, with both functions normalized in the same range. The theoretical minimum value for the functions is 1; acceptable values for such network functions are in the range [1, 10] meaning that the traffic demands and delays restrictions (D , DR matrices) are accomplished by the routing configuration.

2.2 Multi-Objective Evolutionary Algorithms

This section explains how multi-objective Evolutionary Algorithms (MOEAs) are adapted to be used in this work. The multi-objective nature of this problem

² It was considered that the delay in each path is dominated by propagation delays. However, if required, queuing delays are also easily incorporated in the framework.

³ Details of these cost functions can be found in similar model formulations (e.g. [4]).

suggests MOEAs are good candidates for algorithms in the multi-objective optimization area, i.e. that return a set of solutions with distinct trade-offs between the two objectives, allowing the network administrator decide which solution to implement. Since the mid-1980's, MOEAs are being used to solve all kinds of multiple-criterion problems in distinct scenarios, being undoubtedly one of the most competitive approaches in this field [16]. The MOEAs chosen for this task are two of the most popular algorithms, namely the SPEA2 and the NSGA-II, widely accepted as two of the algorithms with the best overall performance.

In the proposed MOEAs based approach, each individual encodes a solution in the form of a vector of integer values. Here, each value (gene) corresponds to the weight of a link in the topology. Therefore, the size of the individual equals the number of links in the network. The individuals in the initial population are randomly generated, with link weights taken from an uniform distribution. To create new solutions, several reproduction operators were used, e.g. *crossover* and *mutation*. The *random mutation* replaces a given gene by a new value, randomly generated within the allowed range. The *incremental/decremental mutation* replaces a specific gene by the following or by the previous value (with same probabilities) constrained to the range of allowed values. The *uniform crossover* and *two-point crossover* are two standard crossover operators also applied to generate new solutions [11]. The fitness of an individual is derived from the analysis of functions $\Phi^*(w)$, for congestion, and $\gamma^*(w)$, for delays. On each algorithm iteration a sub-set of the best individuals proceeds to the next round, while a new subset of individuals is generated using the above mentioned reproduction operators. The process follows until the last iteration resulting in a set of solutions for the considered multi-objective problem. This process can be repeated several times (distinct runs) in order to obtain additional solutions.

The MOEAs return a Pareto front, i.e. a set of non-dominated solutions, for a given problem. When a solution is dominated by another one, it means that it is worse than the second in at least one of the objectives and it is not better in none. Given this, the Pareto front should be as near as possible to the optimal set of non-dominated solutions. In addition, it should also be as distributed as possible, i.e. it should cover the whole set of possible trade-offs between the optimization aims of the addressed problem. This specific characteristic of the Pareto front is adequate to be used in the context of optimization problems such the ones envisaged in this work. In this specific problem, the aim is to find routing configuration solutions where both objectives (congestion, delays) are within the range [1, 10], as previously stated. The network administrator can then check the solutions within the returned Pareto front and select which is the trade-off that is more reasonable in a given context. Therefore, with a single run of the algorithm a set of alternatives are provided, which brings important advantages both in the administrator results perception and computational efforts perspectives.

2.3 Comparative Heuristics

In addition to the above mentioned optimization techniques, a number of traditional heuristic methods were also implemented, namely: *InvCap* - sets each

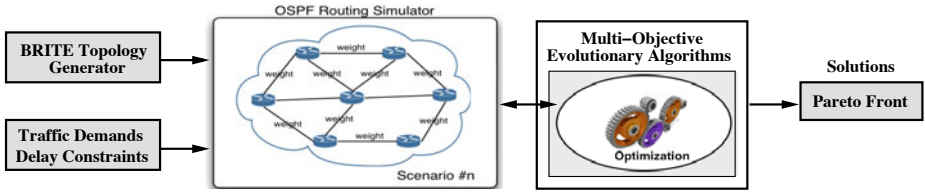


Fig. 2. Testbed for performance evaluation of the proposed framework

link weight to a value inversely proportional to its capacity; $L2$ - sets each link weight to a value proportional to its Euclidean distance; $Random$ - a number of randomly generated solutions are analyzed and the best is selected; and $Unit$ - sets every link weight to one. These heuristic are used for comparative terms, i.e. to assess the order of magnitude of the improvements obtained by the MOEAs.

3 Performance Analysis

Figure 2 shows the experimental testbed used to assess the performance of the devised link weight setting optimization solution. The BRITE [13] topology generator was used to create distinct synthetic networks. The optimization results presented in this section were taken from two networks with $N \in \{30, 50\}$ nodes, with a node average degree of $m = 4$, resulting in topologies with 110 and 190 links, respectively. The capacity of the links varies in the interval $[1, 10]$ Gbits and the topology generation follows the Barabasi-Albert model⁴.

In order to generate the traffic demands and delay constraints matrices two parameters (each one with three distinct values) were used, $D_p \in \{0.1, 0.2, 0.3\}$ and $DR_p \in \{3, 4, 5\}$, allowing to tune the difficulty levels of traffic demands and delay requirements, respectively. The precise values of D_p and DR_p parameters were selected in accordance with the overall congestion and delay constraints levels which are intended to be imposed to each network instance, given their particular characteristics. In the devised test methodology, scenarios assuming higher values for the D_p parameter and, simultaneously, lower values of the DR_p parameter are the ones harder to comply⁵. Based on the network topology, the demand matrices and on an initial weight assignment to the network links, a proprietary OSPF simulator will distribute the traffic along the paths, turning possible the computation of the $\Phi^*(w)$ and $\gamma^*(w)$ values. The optimization module of Figure 2 will then resort to the MOEA optimization approach explained before to find the solution for the optimization problem. In the following sections the results were obtained with the optimization module operating with the NSGA-II algorithm.

⁴ A heavy-tail distribution was used along with an incremental grow type (the parameters HS and LS have values of 1000 and 100, respectively).

⁵ Higher values of D_p mean that higher traffic demands are being considered, being harder to comply. In counterpoint, higher values for the DR_p mean that higher values for edge-to-edge delays requirements are being considered, being easier to comply.

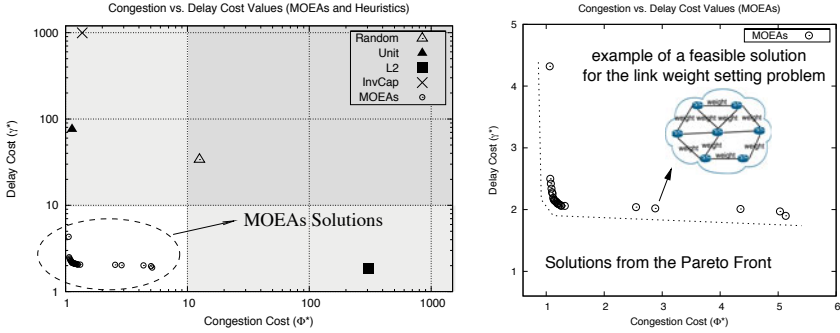


Fig. 3. a) Comparative perspective of the MOEAs and heuristics results (logarithmic scale); b) MOEAs results (i.e. only the white area from Fig. 3 a))

3.1 Illustrative Analysis of MOEAs Performance

As an initial illustrative example of the MOEAs optimization capabilities we select one particular instance of a network with 30 nodes and 110 links, with $D_p = 0.1$ and $DR_p = 3$. In this context, a brief analysis of the Pareto front returned by the MOEA is presented, along with comparative analysis of the performance obtained from the use of commonly used heuristics. This example will be complemented, in the next section, with additional optimization results for other network instances. However, in general terms, the conclusions drawn for this particular instance are representative of the overall performance capabilities of the proposed optimization framework. Figure 3 a) shows a particular subset of the solutions obtained by the MOEAs in the selected scenario and, for comparative terms, the ones obtained by common heuristics described before. Figure 3 a) has two distinct areas, the first one corresponds to solutions assuming routing configurations able to obey the considered traffic and delay demands (i.e. the white area, where cost function values are lower than 10), and a second area where the routing solutions lead to quality degradation of the network, with overloaded links or with the target delays requests not being assured by the network (gray filled area)⁶.

As observed in Figure 3 a) the MOEA available in the proposed solution is able to provide the network administrator with a set of near-optimal routing configuration solutions for the network domain. In opposition, it is also noticeable that all the results of the heuristics for this instance lie outside the admissible range (i.e. outside the white area), and some of them with penalties which are one or two orders of magnitude higher than the ones obtained by MOEAs. This means that none of the heuristics is able to provide acceptable routing configurations⁷.

⁶ Note that in the dark gray filled area none of the requests are accomplished.

⁷ Note that in Fig. 3 a) a logarithmic scale is used, meaning that points outside the white area represent, in fact, extremely poor quality routing solutions.

Figure 3 b) magnifies the white area of Figure 3 a) and now only the MOEAs solutions are plotted. As observed, the administrator achieves a set of near-optimal configuration solutions resulting from the Pareto front of the optimization process, all lying inside of the white area. Each one of such points (solutions) is associated with a routing weights table able to be used by the administrator according with the desired trade-off between the optimization objectives.

3.2 Optimization Results

This section analyzes the MOEAs optimization results for ten distinct network scenarios. The examples were taken from two networks with $N \in \{30, 50\}$ nodes and using five distinct combination of (D_p, DR_p) parameters, in this case the set $\{(0.1, 3), (0.1, 5), (0.2, 4), (0.3, 3), (0.3, 5)\}$. In the considered scenarios, the behavior of the heuristics is similar to the describe before, i.e. they are not capable of achieving acceptable performance, meaning that these solutions are completely outside of the white area of the graphs. For that reason, such results are not included in the following analysis.

Figure 4 shows the Pareto fronts obtained for each of the considered scenarios. It is important to note that results were obtained in the first runs of the MOEA optimization procedures, i.e. only a single run of the optimization algorithm has been performed for each of the network instances. As observed in Figure 4, for most of the scenarios the first run of the MOEA was sufficient to find acceptable results i.e. Pareto fronts with solutions (i.e. weight settings) in the white area of the figures. The administrator is then able to select these solutions in order to reach near-optimal routing configurations obeying to the imposed constraints.

As expected, the harder optimization scenarios are the ones imposing higher requirements regarding both the traffic demands and delay restrictions (e.g. $D_p = 0.3, DR_p = 3$). In such scenarios, the Pareto front patterns returned by the MOEAs are not so close to the graph origin as in other network configurations. As observed in Figure 4, the first run of the MOEA was not sufficient to find near-optimal configurations for the last two scenarios with $N = 50$ and (D_p, DR_p) values of $(0.3, 3)$ and $(0.3, 5)$. To improve such preliminary results additional runs of the MOEAs could be used to generate other weight setting configurations overcoming the performance obtained in previous runs.

In order to illustrate the previous reasoning, Figures 5 a), b) and c) plot additional optimization results obtained in other runs of the MOEAs for three specific scenarios (including the instances with lower quality results in the first run). The new Pareto fronts depicted in Figures 5 a), b) and c) are compared with the values obtained in the first runs (observed before in Figure 4). The analysis clearly shows an improvement of the Pareto front patterns for each scenario, containing now several points in the white regions of the figures. This behavior is visible in Figures 5 a), b) and c) and corresponds to a generalized displacement of the Pareto fronts to the feasible configuration area.

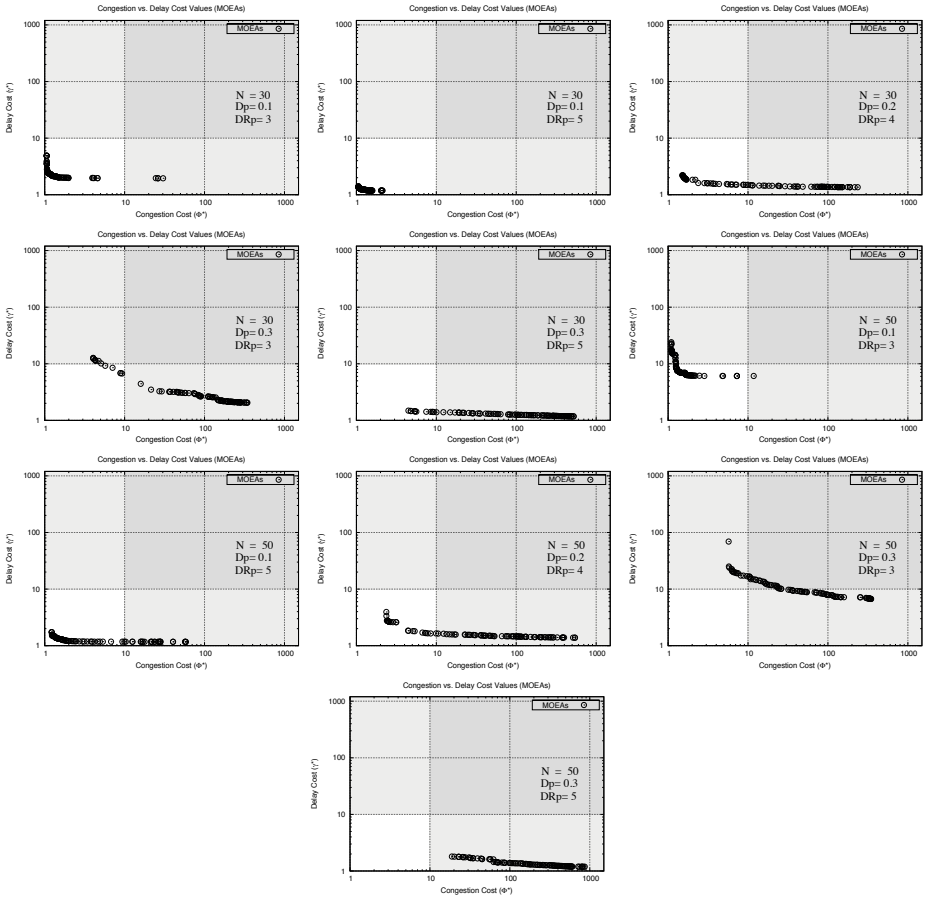


Fig. 4. MOEA optimization results - Pareto fronts - for two synthetic networks ($N = 30, 50$) with distinct combinations of D_p and DR_p parameters (first runs)

3.3 Single-Objective vs. Multi-Objective Evolutionary Algorithms

This section compares the proposed framework approach with other approaches in the area which assume this problem under a single objective optimization perspective. In such works, a linear weighting scheme could be used to denote the overall cost of the solution, as expressed in Equation 1.

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

When using a single objective evolutionary algorithm, the user specifies the parameter (α) defined above, that determines the importance that is given to each objective (congestion and delays). Examples of performance analysis of this approach can be found in [4], for a large set of distinct QoS constrained scenarios. Although this strategy has obtained acceptable results, it suffers from one

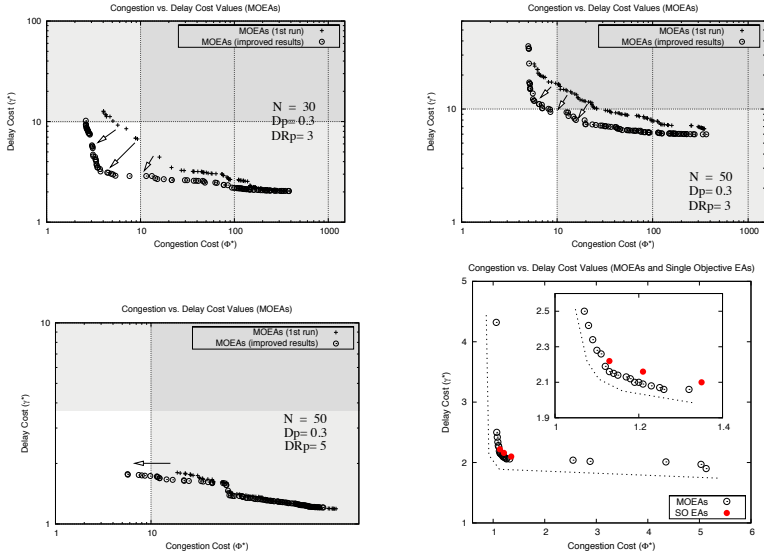


Fig. 5. a) b) c) Improved results for three particular network instances of Figure 4 (in runs 2, 23 and 16, respectively); d) MOEAs and single objective algorithm results

main drawback, since it assumes that there is one single trade-off that is optimum. Therefore, the algorithms typically return one single solution that has to be implemented by the administrator. To be able to analyze several distinct trade-offs between the two objectives, the user needs to execute different runs of the algorithm using different values of the parameter α . Moreover, for specific network configurations several tradeoffs between the congestion and delays requirements are not possible to be achieved, and the administrator will have the doubt about which values are admissible to tune the α parameter.

For comparative analysis, in addition to the MOEA used in the example of Figure 3 also a single objective evolutionary algorithm was run using the objective function of Equation 1. The single-objective algorithm was run with three distinct values of the parameter α (0.25, 0.5 and 0.75). Figure 5 d) shows the results obtained, showing a set of selected solutions obtained by the MOEA and also the best solution obtained by each of the single-objective evolutionary algorithms. It is important to note that all the solutions shown for the multi-objective optimization are obtained in a single run, while the solutions for the single objective need three distinct runs. Looking at the Figure 5 d), and although a direct comparison of the results obtained by the alternative approaches is not trivial, it is clear that running the MOEA the network administrator is provided with a set of alternatives, from where he can choose the best option, reflecting the ideal trade-off between the two objectives. The single objective evolutionary algorithm is normally restricted to a certain area of the working region of the network, making difficult the tuning process of the importance of each

objective. Thus, the MOEA approach of the proposed framework presents advantages not only regarding the computational efforts perspective, but also improves the quality and the diversity of the results provided to the administrator.

4 Computational Efforts, Applicability and Availability

Even taking into account the complexity of the NP-hard problems here discussed, acceptable computational times were obtained when assessing the MOEA optimization process. As illustrative examples, on a current end-user computational platform (e.g. with a Core i7 processor) a single MOEAs run for the considered network examples required a computational time in the order of some minutes. As obvious, this value varies according with the network topology size and, when considering even harder optimization problems, the need of using several MOEAs runs may considerably increase the required computational times. In this perspective, if required, the use of computational clusters environments might be also an alternative to foster the process of obtaining high quality solutions.

It is important to note that, as known, frequent changes to the network routing configurations may lead to network instability. In this context, short-time updates to the routing configurations of a domain are not the scenarios envisioned by this work. Instead, the devised optimization framework was conceived with the aim of being an useful tool to assist network operators in the process of adjusting routing configurations for specific, stable and well known time periods, in conformity with previous studies expressing the demands required to be accomplished. This will allow that, in a first network configuration phase, a more efficient distribution of the traffic in the domain could be achieved. As obvious, other QoS oriented mechanisms are then expected to be applied, in a complementary way, in order to provide a finer-grain control of the service levels supported by the infra-structure. Another related area that could take advantage of the proposed solution, although not explored in this work, is the field of autonomic/self-management networks, as the devised optimization framework might also be useful to be integrated in such automated environments.

A preliminary version of the software implementing the proposed optimization framework was included in a traffic engineering platform which is made available in the home page accessible at <http://darwin.di.uminho.pt/netopt>.

5 Conclusions

This paper describes an evolutionary computation based traffic engineering framework, allowing the network administrator to improve the QoS levels on IP networks by obtaining near-optimal OSPF routing configurations. As an important contribution for the traffic engineering research efforts, the proposed framework integrates efficient multi-objective optimization algorithms from the evolutionary computation area. The presented results corroborate the idea that this optimization solution brings important advantages and enlarges the set of available options to network administrators, in shorter computation times,

increasing the capacity of making informed decisions regarding the trade-offs between the different factors at stake. The supporting framework is currently under rapid development and additional functionalities may be also target of the MOEA optimization perspective. The class-based [15] and multicast [14] optimization mechanisms are two examples of developments which are suitable to be improved taking advantage of the presented MOEA based optimization approach.

References

1. Wang, Z.: *Internet QoS: Architectures and Mechanisms for Quality of Service*. Morgan Kaufmann Publishers, San Francisco (2001)
2. Fortz, B., Thorup, M.: *Internet Traffic Engineering by Optimizing OSPF Weights*. In: *Proceedings of IEEE INFOCOM*, pp. 519–528 (2000)
3. Dijkstra, E.W.: *A note on Two Problems in Connexion with Graphs*. *Numerische Mathematik* 1(269-271) (1959)
4. Sousa, P., Rocha, M., Rio, M., Cortez, P.C.: *Efficient OSPF Weight Allocation for Intra-domain QoS Optimization*. In: Parr, G., Malone, D., Ó Foghlú, M. (eds.) *IPOM 2006*. LNCS, vol. 4268, pp. 37–48. Springer, Heidelberg (2006)
5. Rocha, M., Sousa, P., Cortez, P., Rio, M.: *Quality of Service Constrained Routing Optimization using Evolutionary Computation*. *Applied Soft Computing* 11(1), 356–364 (2011)
6. Medina, A., et al.: *Traffic matrix estimation: Existing techniques and new directions*. *Computer Communication Review* 32(4), 161–176 (2002)
7. Davy, A., Botvich, D., Jennings, B.: *An Efficient Process for Estimation of Network Demand for QoS-Aware IP Network Planning*. In: Parr, G., Malone, D., Ó Foghlú, M. (eds.) *IPOM 2006*. LNCS, vol. 4268, pp. 120–131. Springer, Heidelberg (2006)
8. Gunnar, A., Johansson, M., Telkamp, T.: *Traffic matrix estimation on a large IP backbone: a comparison on real data*. In: *IMC 2004: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement*, pp. 149–160 (2004)
9. Ahuja, R., Magnanti, T., Orlin, J.: *Network Flows*. Prentice Hall, Englewood Cliffs (1993)
10. Moy, J.: *RFC 2328: OSPF version 2* (April 1998)
11. Michalewicz, Z.: *Genetic Algorithms + Data Structures = Evolution Programs*, 3rd edn. Springer, USA (1998)
12. Moy, J.: *OSPF, Anatomy of an Internet Routing Protocol*. Addison Wesley, Reading (1998)
13. Medina, A.L.A., Matta, I., Byers, J.: *BRITE: Universal Topology Generation from a Users Perspective*, Tech. Rep. 2001-003 (January 2001)
14. Sousa, P., Rocha, M., Cortez, P., Rio, M.: *Multiconstrained Optimization of Networks with Multicast and Unicast Traffic*. In: Pavlou, G., Ahmed, T., Dagiuklas, T. (eds.) *MMNS 2008*. LNCS, vol. 5274, pp. 139–150. Springer, Heidelberg (2008)
15. Sousa, P., Rocha, M., Rio, M., Cortez, P.: *Class-Based OSPF Traffic Engineering Inspired on Evolutionary Computation*. In: Boavida, F., Monteiro, E., Mascolo, S., Koucheryavy, Y. (eds.) *WWIC 2007*. LNCS, vol. 4517, pp. 141–152. Springer, Heidelberg (2007)
16. Coello, C.: *A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques*. *Knowledge and Information Systems* 1(3), 129–156 (1999)