A Framework for Robust Traffic Engineering Using Evolutionary Computation

Vitor Pereira, Miguel Rocha, Paulo Cortez, Miguel Rio, Pedro Sousa

A Framework for Robust Traffic Engineering using Evolutionary Computation

Vitor Pereira[1], Miguel Rocha[2], Paulo Cortez[3], Miguel Rio[4], Pedro Sousa[1]

[1]Centro Algoritmi/Dep. of Informatics, University of Minho, Portugal, vmspereira@iol.pt, pns@di.uminho.pt
[2]CCTC/Dep. of Informatics, University of Minho, Portugal, mrocha@di.uminho.pt
[3]Centro Algoritmi/Dep. of Information Systems, University of Minho, Portugal, pcortez@dsi.uminho.pt
[4]Dep. of Electronic and Electrical Engineering, UCL, UK, m.rio@ee.ucl.ac.uk

Abstract. In current network infrastructures, several management tasks often require significant human intervention and can be of high complexity, having to consider several inputs to attain efficient configurations. In this perspective, this work presents an optimization framework able to automatically provide network administrators with efficient and robust routing configurations. The proposed optimization tool resorts to techniques from the field of Evolutionary Computation, where Evolutionary Algorithms (EAs) are used as optimization engines to solve the envisaged NP-hard problems. The devised methods focus on versatile and resilient aware Traffic Engineering (TE) approaches, which are integrated into an autonomous optimization framework able to assist network administrators. Some examples of the supported TE optimization methods are presented, including preventive, reactive and multi-topology solutions, taking advantage of the EAs optimization capabilities.
Keywords: Robust Traffic Engineering; Evolutionary Computation; Network Resilience; Autonomous Configuration

1 Introduction

Nowadays, IP-based networks are the main communication infrastructures used by a growing number of heterogeneous applications and services. This circumstance fostered the need for efficient and automated tools able to assist network management tasks and assuring the correct planing of resilient network infrastructures [1]. In this context, in order to attain acceptable network service levels there are several components that should be correctly configured and coordinated. Irrespective of the wide variety of specific solutions to enforce acceptable network performance, the efficient configuration of routing protocols still plays a vital role in the networking area. In fact, accurate routing configurations are essential to improve network resources usage, also allowing that upper layer protocols, applications and overlay systems have a trustable, resilient and optimized communication infrastructure.

The simplicity and popularity of some well known intra-domain routing protocols (e.g. such as OSPF or IS-IS) has motivated the appearance of seminal research work (e.g. [2]) involving Traffic Engineering (TE) approaches, aiming to attain near-optimal OSPF weight setting (OSPFWS) configurations for a given set of traffic demands, usually represented as a demand matrix. The results of such preliminary efforts have motivated several researchers to the improvement of such TE approaches. Moreover, the recent advances in traffic estimation techniques and the availability of tools within such purposes [3, 4] opened the opportunity for such theoretical approaches to be effectively applied in real network environments. The OSPFWS problem is by nature NP-hard and, among many other techniques, Evolutionary Algorithms (EAs) have been proposed to improve routing configurations [5]. Additionally, several studies highlighted the advantages of such enhanced configurations over traditional heuristics usually adopted by administrators [6], and their use in multi-constrained TE optimization contexts involving several QoS related constraints [7, 8]. However, many of such works, despite proving the efficiency of EA based optimization processes, still present some limitations, usually assuming static optimization conditions, not considering possible changes in specific optimization input parameters.

In this context, this work aims to contribute for devising more versatile and robust TE optimization mechanisms fostered by the use of EAs. In particular, the following topics summarize the main contributions of this work: *i)* the definition of EA based preventive TE methods able to deal a priori with network condition variations, such as demands variation and link failures; *ii)* the proposal of reactive TE optimization methods fostering the EA response time in achieving new configurations with a reduced instability impact in the infrastructure when conditions in the network change significatively; *iii)* support to multi-topology TE optimization techniques able to increase the traffic volumes supported by the network, and, *iv)* the integration of the devised methods in a freely available optimization tool to assist network administrators. As a result, this work clearly wides the applicability, versatility and robustness of existent TE optimizing approaches, resorting to fine-tuned EA based mechanisms, being a further step toward attaining autonomous and robust network optimization tools.

The paper proceeds with Section 2, describing the devised optimization framework, highlighting its mains components, the underlying mathematical basis and the use of Evolutionary Algorithms; Section 3 illustrates some optimization capabilities of the framework, resorting to resilient aware EA optimization methods (with preventive and reactive approaches) and multi-topology based optimization processes; Section 4 presents the conclusions of the work.

2 A Framework for Robust Traffic Engineering

Figure 1 presents the conceptual architecture of the implemented TE optimization framework. As main inputs, the framework receives a description of the network topology, along with the expected traffic demands denoting the traffic volumes that, on average, traverse the network domain[1]. The framework inter-

[1] There are several techniques to obtain traffic demand matrices, which provide estimations about the overall traffic requirements imposed to a given domain (e.g. [3,4,9]).

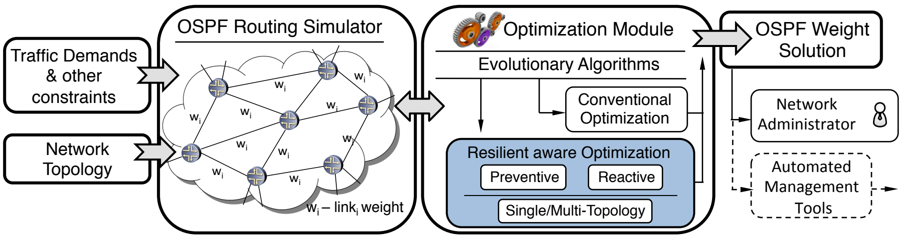


Fig. 1. Illustrative description of the devised Traffic Engineering Framework

nal core includes a routing simulation module that, for a given topology, routing weights and demand matrices, distributes the traffic along the network links, thus obtaining an estimation of the foreseeable congestion levels. The optimization module is the core of the framework and resorts to several mechanisms from the field of Evolutionary Computation, namely Evolutionary Algorithms (EAs). The optimization module achieves near-optimal OSPF weight setting solutions using conventional optimization approaches, where EAs seek for weight solutions able to efficiently accommodate the considered demands. This module includes a resilient aware optimization sub-module, considering resiliency issues that may affect the network (e.g. topology changes, link failures, variable traffic demands, etc.). For this purpose, this sub-module integrates both preventive and reactive optimization engines. Within the former approach, administrators may consider that some disruptive events will affect the network infrastructure and use preventive approaches to achieve solutions that, even in the presence of such events, assure acceptable network performance. In alternative, when preventive solutions are not possible to be considered, reactive methods focus on re-optimizing a given configuration considering the new operational conditions. Here, the objective is to foster the optimization process and assure that the new weight configuration has a reduced instability impact in the network. Moreover, the framework is also able to assume multi-topology optimization scenarios, to further improve network resources usage and robustness levels. As output, the system will provide network administrators with near-optimal weight setting solutions that might be used in subsequent configuration processes. If required, the optimization framework may also be integrated with automated management tools that, if conveniently tuned, can effectively contribute for the development of autonomous and resilient aware network infrastructures.

The presented framework, integrating all the mechanisms here described, allows a user friendly interaction with the devised methods hiding their inherent complexity. An open source version of the implemented framework is made available in `http://darwin.di.uminho.pt/netopt`.

2.1 Mathematical Formulation

The basic mathematical networking model used by the framework is a directed graph $G = (N, A)$, which represents routers by a set of nodes (N) and transmission links by a set of arcs (A). A solution to the OSPFWS problem is given by a link weight vector $\mathbf{w} = (w_a)$ with $a \in A$. OSPF requires integer weights from 1 to 65535 $\left(2^{16}-1\right)$ [10], but the range of weights can be reduced to smaller intervals $[w_{min}, w_{max}]$, the interval $[1, 20]$ is used in this work. This reduces the search space and increases the probability of equal cost paths [11]. Given a demand matrix D, consisting of several d_{st} entries for each origin and destination pair (s,t), where d_{st} is the amount of data traffic that enters the network at point s and leaves the network at point t, the problem consists in routing these demands over paths of the network, minimizing a given measure of network congestion. For each arc $a \in A$, the capacity is expressed by $c\,(a)$ and the total load by $\ell\,(a)$. For a given candidate weight vector w, for which the Dijkstra algorithm [12] determines the shortest paths, the total load over a is the sum of the $f_a^{(s,t)}$ terms, representing how much of the traffic demand between s and t travels over arc a. Here, the cost of sending traffic through arc a is given by $\Phi\,(\ell\,(a))$. The cost value depends on the utilization of the arc and the devised framework adopted the well known linear function proposed by Fortz and Thorup [11], presented in Equation 1. The objective of the OSPFWS problem is to distribute the traffic demands to minimize the sum of all costs, as expressed in Equation 2.

$$\Phi_a^{'}\,(x) = \begin{cases} 1 & for \;\; 0 \leq x/c\,(a) < 1/3 \\ 3 & for \;\; 1/3 \leq x/c\,(a) < 2/3 \\ 10 & for \;\; 2/3 \leq x/c\,(a) < 9/10 \\ 70 & for \;\; 9/10 \leq x/c\,(a) < 1 \\ 500 & for \;\; 1 \leq x/c\,(a) < 11/10 \\ 5000 & for \;\; x/c\,(a) \geq 11/10 \end{cases} \quad (1)$$

$$\Phi = \sum_{a \in A} \Phi_a\,(\ell\,(a)) \quad (2)$$

To enable results comparison among distinct topologies, a normalized congestion measure Φ^* is used. It is important to note that when Φ^* equals 1, all loads are below 1/3 of the link capacity, while when all arcs are exactly full the value of Φ^* is 10 2/3. This value will be considered by the framework as a threshold[2] that bounds the acceptable working region of the network. Some of the optimization mechanisms later discussed in this work introduce some variants to this base mathematical formulation and to the objective function Φ^*.

2.2 The use of Evolutionary Algorithms

The optimization problems addressed by the proposed framework are NP-hard and in such context EAs can be used to improve routing configurations, namely in the resilient aware sub-module of the framework. In general terms, in the proposed EAs, each individual encodes a solution as a vector of integer values, where each value corresponds to the weight of a link ($W_a, a \in A$). The objective function used to evaluate each individual (solution) in the EAs varies depending on the target of the optimization. As example, a simple conventional optimization approach might implement the minimization of the congestion according to

[2] For visualization, congestion values above such threshold (i.e. $\Phi^* > 10\ 2/3$) are marked with a gray filled area in the Tables and Figures of this paper.

Algorithm 1: Generic structure of EAs used in the Optimization Module

1 $t = 0$;
2 INITIALIZE $P(0)$;
3 EVALUATE $P(0)$;
4 **while** *t is less than Maximum number of generations* **do**
5   SELECT parents for reproduction;
6   APPLY REPRODUCTION operators to create offspring ;
7   EVALUATE offspring ;
8   SELECT the survivors from $P(t)$ to be kept in $P(t+1)$;
9   INSERT offspring into $P(t+1)$;
10     $t = t + 1$;
11 **endw**

Equation 2 mentioned before, using an initial population randomly generated in the $[1, 20]$ range. In each EA generation, two mutation and one crossover operators are used in the reproduction step to generate new individuals (offspring): *Random mutation*, *Incremental/decremental mutation* and *Uniform crossover*. A roulette wheel scheme is used in the selection procedure, firstly converting the fitness value into a linear ranking in the population. In each generation, 50% of the individuals are kept from the previous generation. In the experiments, a population size of 100 was considered. The EAs used in this work follow the generic structure given by Algorithm 1. Additional details about the evaluation strategies (lines 3, 7) and initial population filling strategies (line 2) are discussed in section 3, within the context of the presented illustrative mechanisms.

3 Illustrative Methods and Experimental Results

This section includes illustrative examples of resilient aware optimization methods supported by the framework, namely: *i)* preventive optimization methods to deal with heterogeneous traffic demand matrices; *ii)* preventive optimization methods for link failure scenarios; *iii)* a reactive optimization example fostering the EA convergence also reducing the impact of the new configurations and *iv)* multi-topology optimization approaches able to accommodate larger volumes of traffic in the infrastructure also improving the network resiliency to an increase of demands. In order to test the framework, several synthetic networks were generated with the Brite topology generator [13], using the Barabasi-Albert model, with a heavy-tail distribution and an incremental grow type. The link capacities uniformly vary in the interval $[1, 10]$ Gbits. For testing purposes, and considering the topologies characteristics, demands matrices instances (D) are automatically generated, tuned by a D_p parameter with values in $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ where larger values imply harder optimization problems[3].

[3] Previous works showed that matrices generated with $D_p = 0.3, 0.4, 0.5$, induce very hard optimization problems and such instances will be used in the experiments.

Due to space constraints, the next sections results were taken from a network topology instance with 30 nodes and 55 links, but being representative of the usual behavior of the framework optimization methods in other topology instances. In all of the following illustrative examples, 10 runs of the corresponding EA were made, being the results shown the mean of the obtained values.

3.1 Preventive Optimization - Traffic Demands

The traditional routing problem deals with the selection of paths to route given amounts of demands between origin and destination routers, and some previous works assumed that the volume of traffic between each source-destination pair is known and fixed. However, the variety of services in the contemporary networks translates into traffic variations that hinder the planning and management of networks only based on static traffic demands. As example, traffic demands may follow many times periodic and foreseeable changes resulting in matrices with distinct levels of demands for distinct time periods [14, 15] or demand matrices that, despite inducing similar overall levels of traffic, may have quite distinct source-destination individual entries. Thus, with the proposed framework it is possible to seek for weight configurations promoting an acceptable level of congestion for a given set of considered traffic matrices. In this context, we redefine the OSPF weight setting task as a multi-objective problem. For a given network topology and a set of demand matrices D_i, the aim is to find a set of weights w that simultaneously minimize the functions $\Phi_i^*(w)$, where $\Phi_i^*(w)$ represents the function $\Phi^*(w)$ considering the traffic demands of matrix D_i. For a maximum of D_{max} matrices, the multi-objective optimization is achieved using the aggregated objective function of Equation 3, instantiating the evaluation steps in lines 3 and 7 of Algorithm 1. The α_i parameters ($\alpha_i \in [0,1]$, $\sum \alpha_i = 1$) will tune the importance of each partial objective in the optimization process.

A practical example of optimization considering two generic traffic matrices D_1, D_2 is now presented. In such scenario, Equation 4, with $\alpha_2 = 1 - \alpha_1$, represents the objective function that was used in Algorithm 1. Some results are presented in Table 1 for two pairs of demand matrices $\{D_1 = D0.5_a, D_2 = D0.4\}$ (distinct matrices with distinct overall levels of demands) and $\{D_1 = D0.5_a, D_2 = D0.5_b\}$ (distinct matrices with similar overall levels of demands). In the experiments, α_1 was set to 0.5, giving equal importance to both matrices (comparative values with $\alpha_1 = 1, 0$ are also shown). As observed in the presented results, an optimization process that is usually executed for a specific demand matrix may not be good enough for other distinct matrices. In Table 1, a single objective solution performed for the demand matrix $D0.5_a$, with congestion measure 2.894, is inadequate for the demand matrix $D0.5_b$, as the congestion measure reaches over 50. It is possible, using the proposed method, to obtain a suitable configuration for both matrices, only slightly compromising the congestion level in each individual scenario (last row of results in Table 1). Under this scheme, the administrator is able to fine tune adjustments, such as favoring one of the matrices and penalizing the other (α_1 is set to define such trade-off).

Table 1. Optimization results for two traffic demand matrices - $\Phi_i^*(w)$ values

D_1–D_2	$D0.5_a$–$D0.4$		$D0.5_a$–$D0.5_b$	
$\Phi_i^*(w)$	$\Phi_1^*(w)$	$\Phi_2^*(w)$	$\Phi_1^*(w)$	$\Phi_2^*(w)$
α_1=1	3.021	43.693	2.894	52.859
α_1=0	64.892	2.061	28.440	3.894
α_1=0.5	3.583	2.326	4.722	4.886

$$f(w) = \sum_{i=1}^{D_{max}} \alpha_i \Phi_i^*(w) \quad (3)$$

$$f(w) = \alpha_1 \Phi_1^*(w) + (1 - \alpha_1)\Phi_2^*(w) \quad (4)$$

3.2 Preventive Optimization - Link Failures

Besides traffic demands changes, other topology level events (e.g. such as a link failures) may also have a severe impact on the network performance, as traffic previously flowed through the failed link is shifted to other recalculated routes, which may cause congestion in parts of the network. In this section, we propose an EA based preventive optimization mechanism to improve network resilience to link failures. For a given network topology with n links the aim is to find a set of weights w that simultaneously minimize the function $\Phi_n^*(w)$, representing the congestion cost the network in the normal state, and other possible additional functions $\Phi_{n-i}^*(w)$, representing the congestion cost of the network when foreseeing that i specific links from the topology have failed[4]. The multi-objective optimization may use a generic objective function as in Equation 5, instantiating the EA evaluation steps (lines 3,7 of Algorithm 1), with $\alpha_i \in [0,1]$ and $\sum \alpha_i = 1$.

As a practical optimizing example, it is assumed the specific case where the administrator intends to preventively protect the failure of the link with the highest traffic load, which may configure one of the worst single link failure scenarios for congestion. In this case, the objective function integrates a first state where all links are functional and another where it foresees that the link has failed. For each candidate solution w, the proposed EA algorithm assesses the congestion level of the network without failure (Φ_n^*) and assuming the failure (Φ_{n-1}^*), as expressed by Equation 6, with $\alpha_{n-1} = 1 - \alpha_n$. In our proposal, we consider a flexible factor α_i that can assume any value in the range [0, 1], instead of using a fixed weighting factor, thus giving more flexibility to network administrators. As example, [16, 17] present alternatives approaches that resort to other optimization methods, also differing in the configuration of the weighting factor. In Equation 6, when $\alpha_n = 1$, the optimization is only performed for the normal state topology, without any link failures, whereas when using $\alpha_n = 0.5$ the same level of importance is given to the two topology states. However, as the link failure optimization can compromise the network congestion level in a normal state, a network administrator may wish to focus on the performance of the normal state network, e.g. using a α_n value between 0.5 and 1, at the expense of the congestion level in a failed state, that may not occur.

[4] The administrator may select such i candidate links (i=$|Z|$, $Z \subset A$) based on a given criteria, such as link failure probabilities, topology related criteria, link loads, etc.

Table 2. Preventive link failure optimization with $\alpha_n = 1$ and 0.5

Demands	Without failure optim. α_n=1		With failure optim. α_n=0.5	
	Before Failure	After Failure	Before Failure	After Failure
D0.3	1.401	25.242	1.466	1.493
D0.4	1.712	35.524	1.720	1.882
D0.5	3.682	160.043	4.745	4.165

$$f(w) = \alpha_n \Phi_n^*(w) + ... + \alpha_{n-i}\Phi_{n-i}^*(w) + ... \quad (5)$$

$$f(w) = \alpha_n \Phi_n^*(w) + (1 - \alpha_n)\Phi_{n-1}^*(w) \quad (6)$$

$$SPC_{(s,t)} = \frac{\left| SP_{1(s,t)} \cap SP_{2(s,t)} \right|}{max\left(|SP_{1(s,t)}|, |SP_{2(s,t)}| \right)} \quad (7)$$

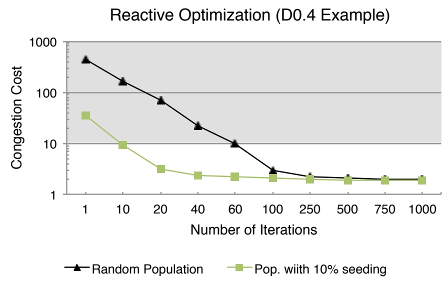
A set of experiments was devised to illustrate this approach, and the results are presented in Table 2. The algorithm was applied to the network topology considering traffic demands levels of $D0.3$, $D0.4$ and $D0.5$, and weighting factors 1 (without link failure optimization as a reference value) and 0.5. Comparing the results of Table 2, one can observe that, under the proposed mechanism, a slightly worse behavior of the network congestion level in its normal state, is largely compensated by a large gain in scenarios of link failure. For $\alpha = 0.5$, with an almost imperceptible penalty in the congestion level (e.g. from 1.712 to 1.720 in the D0.4 instance), the gains on the congestion levels of the link failure network are very significant, reducing from 35.524 (absolutely outside of the acceptable network working region) to 1.882 in the same demand instance. The results obtained in all the demand instances clear indicate the obvious advantages of administrators resorting to this preventive link failure optimization method.

3.3 Reactive Optimization Approaches

The previous illustrated methods may have some inherent limitations, assuming that the network administrator has in advance some knowledge about foreseeable events that may affect the network. Thus, the proposed framework also integrates reactive optimization methods, providing new weight configurations whenever a new re-optimization is triggered. Here, the objectives are *i)* foster the EA convergence to timely provide new appropriate configurations and *ii)* achieve weight solutions with a reduced instability impact in the infrastructure.

For this purpose, after analyzing several optimization examples, it was observed that reactive optimization methods can be often fostered using a special filling strategy of the initial EA population. In this context, the framework saves approximately 10% of individuals from the final population of the previous optimization process and integrates such individuals in the initial population, whenever a new reactive optimization is triggered (i.e. changing the default behavior of line 2 of Algorithm 1). In addition to a faster convergence, this method also tends to assure new weights with a reduced instability impact in the infrastructure, being such estimation also provided to the administrator. The metric presented in Equation 7, $SPC_{(s,t)}$, assesses the changes in the shortest paths between two nodes, (s, t), for two distinct configurations, where $SP_{1(s,t)}$ and $SP_{2(s,t)}$ represent a set of links which integrate the (s, t) shortest paths in the current and in

Iter.	$D0.4$ Rand. Populat.	$D0.4$ Pop.10% seeding	$D0.5$ Rand. Populat.	$D0.5$ Pop.10% seeding
1	449.749	35.824	756.911	160.043
10	166.583	9.263	564.174	135.724
20	71.439	3.105	359.052	99.458
40	22.061	2.374	192.239	60.948
60	9.666	2.245	130.543	45.726
100	3.012	2.110	75.011	25.887
250	2.193	1.958	26.328	10.488
500	2.027	1.909	10.477	8.263
750	1.990	1.887	7.104	7.712
1000	1.956	1.882	6.248	4.022
	$\Delta_{iter.} \approx -80\%$		$\Delta_{iter.} \approx -50\%$	



	$D0.4$ Rand.	$D0.4$ Pop.10%	$D0.5$ Rand.	$D0.5$ Pop.10%
APC	0.7641	0.890	0.683	0.856
	$\Delta_{APC} \approx +17\%$		$\Delta_{APC} \approx +25\%$	

Fig. 2. Reactive link failure optimization: Iterations and APC metric ($D0.4$, $D0.5$)

the new configuration, respectively. The arithmetic mean of $SPC_{(s,t)}$ for all (s,t) pairs, with $s, t \in N$ and $s \neq t$, is denoted by *Average Path Change* (APC), with values in the interval $[0, 1]$. APC values close to 1 represent routing configurations not imposing significant changes to the already established paths, whereas for lower values a higher impact in the network is expected.

As an example, Figure 2 shows the behavior of the EA based reactive link failure optimization, after the link with the highest load has failed (for $D0.4$, $D0.5$ instances). The values are plotted against the conventional optimization approach for a reference baseline. As shown, the proposed reactive strategy has a faster convergence, reducing significantly the number of iterations required to achieve acceptable weight settings[5]. In this case, a decrease in the order of 80% and 50% in the considered instances. Also, in addition to a faster response, the APC values comparison included in Figure 2 clearly shows that the re-optimized configurations have a lower instability impact in the network, comparatively to the conventional optimization, with APC values of 0.890 and 0.856, i.e. improvements in the order of 17% and 25% considering the baseline references.

3.4 Multi-topology Optimization Approaches

This section illustrates other of the framework optimization capabilities, taking as example multi-topology approaches. In this case, it is assumed that the network administrator is only focused on studying the viability of such techniques as a means to maximize network resources usage and improve the infrastructure resilience to demands grow. For that, network edge routers may assume a given pre-defined strategy to internally classify and split traffic among several routing topologies, e.g. a flow level division approach assuring that packets within

[5] It is worth to mention that each EA single iteration involves the generation of several new individuals and the computation of the corresponding fitness functions.

a specific flow are maintained in the same logical topology to avoid packet reordering at end systems. In this optimization mode, the proposed framework resorts to a distinct mathematical model. Given a physical topology represented by the graph $G = (N, A)$, T logical topologies are defined as $G_\tau = (N_\tau, A_\tau)$ with $N_\tau \subseteq N$, $A_\tau \subseteq A$ and $\tau = 1..T$. To model a possible traffic balancing approach, the demands D are uniformly distributed[6] among several D_τ traffic matrices, which are mapped to the T logical topologies, where each d_{st}^{τ} element represents traffic with origin s and destination t that traverses the topology τ. In this multi-topology perspective each logical topology has associated a set of weights, w_τ, ruling the shortest paths computation over such topology and, consequently, the traffic distribution within the network. For optimization purposes, the selected EA uses individuals that aggregate all the w_τ weighting sets, i.e. a vector of integers in the form of $w = \left(w_{(1,1)}, ..., w_{(n,1)}, w_{(2,1)}, ..., w_{(n,T)}\right)$, with $n = |A|$. After the shortest paths computation, for each arc $a \in A$, $f_{st,a}^{\tau}$ represents the traffic from s to t that traverses the arc a in the logical topology τ. For a given specific τ topology, the partial load of arc a derived from such logical topology is represented by $\ell_\tau(a)$, as in Equation 8. The total load of arc a in the physical topology, $\ell(a)$, is then the sum of all partial loads, as in Equation 9. On the proposed EA each candidate solution w is then evaluated using the function Φ^*.

$$\ell_\tau(a) = \sum_{(s,t) \in N \times N} f_{st,a}^{\tau} \qquad (8)$$

$$\ell(a) = \sum_{\tau = 1..T} \ell_\tau(a) \qquad (9)$$

This optimization mode is illustrated resorting to a scenario were the considered network topology is under very heavy traffic constraints, assuming for that purpose a $D0.6$ demand matrix. In this example, a conventional single topology routing approach is not able to find weight settings able to completely accommodated such traffic volumes. Table 3 shows the framework optimization results of the multi-topology approach (including baseline values for a single topology). Such results show that the EA was able to find weight settings perfectly accommodating all the traffic demands, simply by considering an additional topology. This is explained by the EA ability to find weight settings that impose, for each source/destination pair, a considerable dissimilarity level between the shortest paths computed on each of the considered logical topologies. Such perception is further corroborated by Table 4, with the APC values among distinct topologies, where, for each pair of compared topologies, the shortest paths differ roughly in the order of 40%. This justifies why more versatile traffic distribution processes could be achieved with a correctly configured multi-topology routing approach.

An additional results visualization is given by a specific framework interface, showing the link loads distributions, comparing the network behavior when optimized by a conventional method and when using a multi-topology approach, with T=4 (Fig. 3). As seen, in the multi-topology case a more efficient use of link capacities is achieved, Fig. 3b, comparatively to the single topology scenario, Fig. 3a, where a considerable number of links have insufficient capacity to hold the network traffic volumes ([0,1] values denote uncongested links). This method

[6] Other alternatives might be assumed by the framework optimization model, depending on the traffic load distribution techniques adopted in the network.

Table 3. Congestion cost for multi-topology optimization ($D0.6$)

Number of Topologies (T)	Dem. D0.6
1 (T_1: conventional opt.)	34.270
2 ($T_1 + T_2$)	6.230
3 ($T_1 + T_2 + T_3$)	5.926
4 ($T_1 + T_2 + T_3 + T_4$)	5.338

Table 4. Shortest paths comparison (APC values) with T=4 topologies ($D0.6$)

Topology	T_1	T_2	T_3	T_4
T_1	-	0.591	0.639	0.598
T_2	0.591	-	0.590	0.584
T_3	0.639	0.590	-	0.699
T_4	0.598	0.584	0.699	-

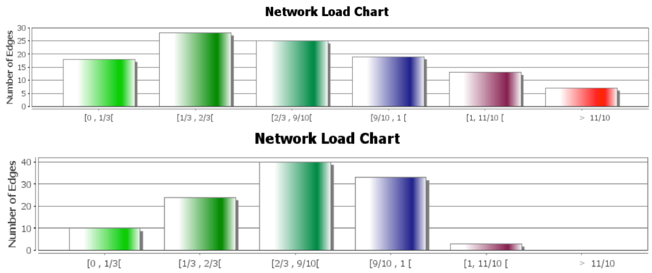


Fig. 3. Link loads distribution for a scenario with traffic demands of $D0.6$: a) without multi-topology optimization b) with optimization for a network with 4 logical topologies

can be used by administrators to optimize multi-topology routing protocols (e.g. Multi-topology OSPF) in order to increase the network ability to support larger traffic volumes, without having to upgrade the existent infrastructure capacity.

4 Conclusions

This work addressed the proposal of versatile and resilient aware EA based TE approaches. The devised mechanisms include preventive approaches, providing network administrators with resilient routing configurations and reactive methods, fostering the response time of the optimization framework, while reducing the instability impact on the existent infra-structure. Other advanced approaches, dealing with multi-topology schemes, were also devised to attain improved network resources usage. It is worth to mention that even with modest end-user computational platforms (e.g. Core 2 Duo/Core i3/etc. processors) the presented NP-hard optimization examples required computational times roughly in the order of some minutes. As obvious, when considering even harder optimization problems, a considerable increase in computational times is expected. In such more demanding scenarios, if administrators need to re-optimize a given configuration, the devised reactive TE approach is an important asset to foster the optimization process and timely provide new near-optimal configurations. Future work will address the definition of other optimization methods widening the framework optimization scope, and the development of additional graphical interfaces allowing to easily define, from the administrator perspective, the network topology submitted to the TE optimization framework.

Acknowledgments

This work is partially funded by FEDER Funds through the Programa Operacional Fatores de Competitividade - COMPETE and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-022674.

References

1. Lee, K., Lim, F., Ong, B.: Building Resilient IP Networks. Cisco Press (2012)
2. Fortz, B.: Internet Traffic Engineering by Optimizing OSPF Weights. In: Proc. of IEEE INFOCOM, pages 519–528 (2000)
3. Cariden Technologies. Building Traffic Matrices: Introduction to MATE Flow Collection. White Paper - Version 2. (October 2012)
4. Davy, A., et al: An Efficient Process for Estimation of Network Demand for Qos-aware IP Networking Planning. In: 6^{th} IEEE Intern. Workshop on IP Operations and Management, IPOM 2006, LNCS 4268, pages 120–131. Springer-Verlag (2006)
5. Ericsson, M., Resende, M., Pardalos, P.: A Genetic Algorithm for the Weight Setting Problem in OSPF Routing. Combinatorial Optimization, 6:299–333 (2002)
6. Sousa, P., Rocha, M., Rio, M., Cortez, P.: Efficient OSPF Weight Allocation for Intra-domain QoS Optimization In: 6th IEEE Intern. Workshop on IP Operations and Management, IPOM 2006, LNCS 4268, pages 37-48, Springer-Verlag (2006)
7. 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)
8. Sousa, P., Cortez, P., Rio, M., Rocha, M.: Traffic Engineering Approaches Using Multicriteria Optimization Techniques. In: Proc. 9th Int. Conf. on Wired/Wireless Internet Communications, pp. 104-115, Springer, LNCS 6649 (2011)
9. Medina, A. et al: Traffic Matrix Estimation: Existing Techniques and New Directions. Computer Communication Review, 32(4):161–176 (2002)
10. Moy, J.: OSPF Version 2. RFC 2328 (Standard) (April 1998)
11. Fortz, B., Thorup, M.: Optimizing OSPF/IS-IS Weights in a Changing World. IEEE Journal on Selected Areas in Communications, 20(4):756–767 (2002)
12. Dijkstra, E.: A Note on Two Problems in Connexion with Graphs. Numerische Mathematik, 1(1):269–271 (1959)
13. Medina, A., Lakhina, A., Matta, I., Byers, J.: BRITE: Universal Topology Generation from a User's Perspective. Technical report 2001-003 (January 2001)
14. Feldmann, A. et al: Deriving Traffic Demands for Operational IP Networks: Methodology and Experience. IEEE/ACM Trans. on Net., 9(3):265–280 (2001)
15. Cortez, P., Rio, M., Rocha, M., Sousa, P.: Multiscale Internet Traffic Forecasting using Neural Networks and Time Series Methods. Expert Systems, 29(2):143–155 (2012)
16. Fortz, B., Thorup, M.: Robust Optimization of OSPF/IS-IS Weights. In: Proceedings of the International Network Optimization Conference, pp.225-230 (2003)
17. Sqalli, M., Sait, S., Asadullah, S.: OSPF Weight Setting Optimization for Single Link Failures. Int. Journal of Computer Networks & Comm., 3(1):168–183 (2011)