Intelligent Environments 2016
P. Novais and S. Konomi (Eds.)

doi:10.3233/978-1-61499-690-3-298

Optimizing Load Balancing Routing Mechanisms with Evolutionary Computation

Vítor PEREIRA [a], Miguel ROCHA [b] and Pedro SOUSA [a,1]
[a] *Centro Algoritmi / Dept. Informatics, University of Minho, Portugal, Vitor.Pereira@algoritmi.uminho.pt, pns@di.uminho.pt*
[b] *Centre of Biological Engineering / Dept. Informatics, University of Minho, Portugal, mrocha@di.uminho.pt*

Abstract. Link State routing protocols, such as Open Shortest Path First (OSPF), are widely applied to intra-domain routing in todays IP networks. They provide a good scalability without lost of simplicity. A router running OSPF distributes traffic uniformly over Equal-cost Multi-path (ECMP), enabling a better distribution of packets among the existent links. More recently, other load balancing strategies, that consider non even splitting of traffic, have been put forward. Such is the case of the Distributed Exponentially-weighted Flow SpliTting (DEFT), that enables traffic to be directed through non equal-cost multi-paths, while preserving the OSPF simplicity. As the optimal link weight computation is known to be NP-hard, intelligence heuristics are particularly suited to address this optimization problem.

In this context, this work compares the solutions provided by Evolutionary Algorithms (EA) for the weight setting problem, considering both ECMP and DEFT load balancing alternatives. In addition to a single objective network congestion optimization problem, both load balancing schemes are also applied to a multi-objective optimization approach able to attain routing configurations resilient to traffic demand variations.

Keywords. Traffic Engineering, Evolutionary Algorithms, Link-State Routing, Load Balancing

1. Introduction

Congestion avoidance in whole or part of an IP network is one the the most important problems for Internet Traffic Engineering (TE). Distinct proposals, that use diverse strategies, have emerged in the networking research community targeting optimal traffic congestion levels on a network. Some approaches are reactive and implement control mechanisms that try to avoid congestion, by performing online traffic measurements and adapting traffic flows at the edge of the network. Other solutions, enabled by new trends like Software Defined Networking (SDN) [1], maximize the network utilization with hybrid deployments [2]. There are also preemptive approaches that optimize path flows, between sources and destinations, based on known or estimated traffic requirements [3].

[1]Corresponding Author: Pedro Sousa, E-mail: pns@di.uminho.pt

The core of the problem lies in the improvement of resource management and, in this context, routing protocols play an important role.

In a link state context, to optimize congestion is to carefully adjust link weights. Link state routing protocols, such as Open Shortest Path First (OSPF) [4], compute the best routes between sources and destinations, minimizing the sum of weights assigned to each link in the path. Those paths are then used to forward network traffic. When more than one shortest path exists to a same destination, OSPF performs Equal-Cost Multi-Path (ECMP) [5], splitting traffic evenly along the multiple paths with equal cost. As a consequence, a better distribution of traffic is achieved using the available resources. However, this advantage of OSPF can also be considered an obstacle to optimal routing, as it is unable to implement non even load balancing of traffic across multiple routes.

An extension to link-state protocols, the Distributed Exponentially-weighted Flow SpliTting (DEFT) [6], was put forward, enabling routers to direct traffic on non shortest paths. DEFT performs an uneven distribution of traffic along available paths, thus allowing the network to attain even lower congestion levels. As SDN technology is increasingly being used in production networks, DEFT can also be considered a good solution for traffic load balancing in SDN enabled switches that use, for example, the OpenFlow Protocol [7], without the burden of heavy configurations and maintenance.

Given the above mentioned, this work focuses on the intelligent optimization of link-state routing protocols using Evolutionary Algorithms (EAs) [8] to obtain near optimal link weight configurations. In particular, two alternative traffic load balancing strategies, able to be used on such link-state routing approaches, are analyzed: the ECMP mechanism and a traffic splitting variant based on the DEFT strategy. We firstly analyze the benefits of using the routing configurations provided by EA based optimization engines on each of the mentioned load balancing alternatives, and compare them with configurations provided by commonly used weight setting heuristics. In order to understand the corresponding advantages and potential gains of each mechanism, a performance comparison between ECMP and DEFT based strategies is also conducted. Finally, a multi-objective optimization approach, able to improve the network resilience to traffic demands variations, is presented and the results analyzed for each of the studied alternatives.

Furthermore, complementing the mentioned topics, this work also provides a contribution for the automated network management area. An intelligent optimization framework was developed by the authors, being used to study the performance of the ECMP and DEFT strategies and to calculate near-optimal weights configurations. Thus, this tool can assist network administrators in the task of obtaining robust routing configurations to attain optimized network infrastructures.

The paper proceeds with Section 2, describing the mathematical model, the DEFT splitting mechanism and the addressed TE optimization problems. Section 3 introduces a summary of EAs and exposes some of their advantages, while Section 4 presents an overall description of the used framework, as well as the necessary configurations, and Section 5 presents some illustrative optimization scenarios for single and multi-objective problems dealing with network congestion issues, and discusses the obtained results. Finally, Section 6 draws conclusions from the obtained results and outlines future work.

2. Mathematical Model and Problems Formulation

This section describes the mathematical model that sustains this study. We firstly formally describe the network representation and the DEFT load balancing mechanism. A description of a congestion measurement function follows, that is used as objective function in the optimization of both load balancing mechanisms.

In this model, network topologies are defined as directed graphs $G(N,A)$, where N represents a set of nodes, and A a set of arcs, with capacity constrains c_a for each $a \in A$. The amount of traffic routed on the arc a, with source s and destination t, is denoted as $f_a^{(s,t)}$. We define the utilization of an arc a as $u_a = \frac{\ell_a}{c_a}$ where ℓ_a is the sum of all flows $f_a^{(s,t)}$ that travel over it.

2.1. Distributed Exponentially-weighted Flow Splitting

As mentioned, ECMP based mechanisms are used by some routing protocols (e.g. OSPF) to evenly split traffic along multiple paths with equal cost weights. Alternative proposals, however, try to diminish the overall congestion levels in the network, by exploring the usage of non equal paths. One of those approaches is explored here, namely the DEFT strategy.

The DEFT load balancing strategy assigns flows to a next-hop with a probability that decreases exponentially with the additional length of the path, when compared with the shortest path. Other uneven traffic splitting proposals for OSPF exist, such as [9,10]. However, they present a relevant limitation. If only link weights are available, routers are unable to independently compute the traffic splitting fractions. They must rely on solutions that involve, for example, a distributed and synchronized load balancing management, which increases communication overhead in the network [11]. DEFT on the other hand, only requires configuring the link weights. Although DEFT also includes a link weight setting optimization mechanism, the here presented proposal differs from the previous by applying DEFT to Multi-Objective Optimization Problems (MOOP) and by resorting to EAs as optimization engines.

The distance from a node u to a node t when traffic is routed thought a node v is expressed as $d_v^t + w_{u,v}$, where d_v^t is the shortest distance from the next-hop v to t, and $w_{u,v}$ is the weight of the link (u,v). The extra length of the path from u to t through v, when compared to the shortest path, is obtained by Eq. 1, and denoted as $h_{u,v}^t$. The flow proportion on the outgoing link (u,v) destined to t, at u , is computed by Eq. 3, where the exponential function Γ, Eq. 2, that decreases with the extra length $h_{u,v}^t$ of a path, maps it into the $[0,1]$ range. As the extra length $h_{u,v}^t$ is an integer value, a parameter p is required to scale the penalizing function Γ to a range $[w_{\min}, w_{\max}]$ of possible weights.

In Figure 1, an example is provided where traffic needs to be routed from node u to node t. The only next-hop on a shortest path from u to t is node v_2. Therefore, and using an ECMP load balancing scheme, all the traffic would be forward by v_2. However, DEFT enables routers to forward traffic on non-shortest paths. All adjacent nodes whose shortest distance to t is less than $d(u,t)(=11)$ will be considered as next-hop. The node v_3 is in such conditions, $d(v_3,t) = 9$, and thus, with DEFT strategy, an extra path with destination t can be used, and a fraction of the traffic destined to t can be forwarded through node v_3. The computed proportional factors $\Gamma\left(h_{u,v_2}^t\right)$ and $\Gamma\left(h_{u,v_3}^t\right)$ (Eq. 2 with $p = 1$) translate into fractional splits (Eq. 3), respectively 88% and 12%.

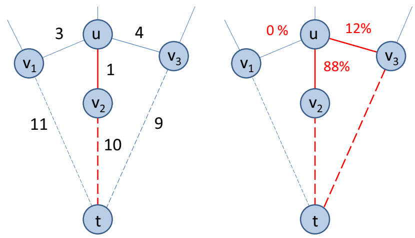

Figure 1. DEFT traffic splitting example

$$h^t_{u,v} = d^t_v + w_{u,v} - d^t_u \tag{1}$$

$$\Gamma\left(h^t_{u,v}\right) = \begin{cases} e^{-\frac{h^t_{u,v}}{p}}, & \text{if } d^t_v < d^t_u \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

$$P\left(h^t_{u,v}\right) = \frac{\Gamma\left(h^t_{u,v}\right)}{\sum_{(u,i)\in A}\Gamma\left(h^t_{u,i}\right)}, \tag{3}$$

2.2. *Objective Function*

A well known piecewise linear cost function Φ_a, proposed by Fortz and Thorup [12], is used to heavily penalize over-utilized links. Given a weight setting configuration w, traffic requirements, modeled as a matrix D, are distributed onto the available network resources considering the computed paths. For each link, the Φ_a function computes a congestion cost considering its utilization ratio u_a. The derivative of Φ_a is defined as:

$$\Phi'_a = \begin{cases} 1 & for\ 0 \leq u_a < 1/3 \\ 3 & for\ 1/3 \leq u_a < 2/3 \\ 10 & for\ 2/3 \leq u_a < 9/10 \\ 70 & for\ 9/10 \leq u_a < 1 \\ 500 & for\ 1 \leq u_a < 11/10 \\ 5000 & for\ u_a \geq 11/10 \end{cases} \tag{4}$$

$$\Phi = \sum_{a\in A} \Phi_a \tag{5}$$

A congestion measure for the entire network, Φ, can than be obtained by adding the congestion cost of all links, as expressed by Eq. 5. To enable results comparison between distinct topologies, a normalized congestion measure Φ^* is used. It is important to note that when all arcs are exactly full the value of Φ^* is 10 2/3. This value will be considered, in the experimental section of this paper, as a threshold that bounds the acceptable working region of the network. Congestion values above 10 2/3 mean that, for a specific weight setting configuration, the network topology is unable to support the considered traffic demands. For simplicity, in this document, all mentions of the congestion measure function Φ refer to its normalized form.

2.3. *Problems Formulation*

The single and multi-objective optimization problems here addressed, aim to optimize the network congestion for known static traffic demands (single-objective) or when changes on the traffic requirements are assumed (multi-objective). Both optimization problems can be formulated as presented, respectively, in P1 and P2.

P1: Given a network topology and a traffic demand matrix, the aim is to find a weight setting w that minimizes the function Φ which evaluates the network congestion cost.

P2: Given a network topology and two demand matrices D_1 and D_2, the aim is to find a weight setting w that simultaneously minimizes the functions $\Phi_1(w)$ and $\Phi_2(w)$, that represent, respectively, the congestion function Φ considering the traffic demands of matrix D_1 and D_2.

In Section 5 some results for both single and multi-objective optimization are presented, as well as the congestion values obtained for each of the two distinct load balancing schemes, ECMP and DEFT.

3. Evolutionary Algorithms

Evolutionary algorithms (EAs) have amply shown their promise in solving various search and optimization problems since the mid-80th. Derived from the principles of natural selection and evolutionary theory, they are particularly suited to address NP-hard problems that have multiple conflicting goals. Such is the case of the TE multi-objective optimization problems here pursued.

An EA transforms a population of individual solutions, each with an associated fitness value, into a new generation, using the Darwinian principle of survival of the fittest. Each solution in the search space of the problem is encoded into a representation suitable for the EA. In this particular case, each individual encodes a routing solution as a vector of integer values, where each value corresponds to the weight of a link in the network. At each iteration of the EA, a new generation of solutions is obtained by applying reproduction and mutation operators. A new population is then built from the pool of all solutions by means of a selection mechanism.

Multi-objective optimization problems are those for which there are simultaneously at least two objective functions. These distinct, and in most cases, conflicting goals give rise a set of trade-off solutions with equivalent quality. Multi-objective EAs (MOEA) are very appealing, as they are able to deliver, in a single run, a set of possible solutions. This solution set of non-dominated solutions, when plotted in the objective space, is known as a Pareto front (Figure 2), being the main goal in multi-objective optimization. Two distinct non-dominated solutions in the Pareto front, are such that, if one is better than the other on the accomplishment of one objective, then it is certainly worse on at least one of the others. Furthermore, MOEA maximizes the diversity of the generated solutions in terms of trade-off between the objectives. In the context of the addressed TE optimization problems, with such a set of solutions, a network administrator would be presented with several configuration settings instead of a single one, which would be the case for most other approaches that are not population based optimization heuristics.

4. Experimental Framework and Configuration Settings

The experiments where performed on a publicly available optimization framework, NetOpt, developed by the authors, and whose validity has been demonstrated in other studies (e.g. [13,14,15]). A preliminary version of the optimization framework is available at `http://darwin.di.uminho.pt/netopt`.

4.1. Experimental 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. The framework internal core includes a routing simulation module that distributes the traffic along the network links, considering ECMP or DEFT load balancing schemes, and thus provides an estimation of the foreseeable congestion levels. The optimization module is the core of the framework and resorts to several Evolutionary Algorithms, namely two multi-objective algorithms, the Non-dominated Sorting Genetic Algorithm (NSGA-II) [16], and the Strength Pareto Evolutionary Algorithm (SPEA2) [17] and a Single-Objective Evolutionary Algorithm (SOEA) that uses weighted-sum aggregation. In the scope of the present work, only NSGA-II was used for the MOOP as it has been experimentally shown in previous work by the authors that it delivers better solutions.

The produced solutions are integer vectors of weights configuration that take values in a user configurable range $[w_{\min}; w_{\max}]$. In real implementations, OSPF link weights are valued from 1 to 65535, but here only values in range [1; 20] were considered, in order to reduce the search space and, simultaneously, to increase the probability of finding equal cost multi-paths. The parameter p, in Eq. 2, is correlated with the weights range maximum value $w_{\max}$. For proof of concept, in the experiments a p value of 1.0 was considered. Future work will analyze the use of other values for p values.

4.2. Configuration Settings

The experiments were run on three synthetic networks, varying the number of nodes and the average degree of each node. The link capacities are uniformly distributed in the interval [1; 10] Gbits. For each topology, a set of random demand matrices $D\alpha$ was generated, where α identifies the expected mean of congestion in each link. In more detail, for each pair of nodes (s,t), $s \neq t$, the amount of traffic from s to t is modeled by Eqs. 6 and 7, where R is a random number in range $[0,1]$, $d_{s,t}$ is the euclidean distance between both nodes, $\overline{c_a}$ is the average capacity of all links, $|E|$ is the number of links in the topology and $H_{s,t}$ the minimum number of hops between s and t. The use of the euclidean distance in the formulation $D(s,t)$ implies that close pairs of nodes have relatively more demand.

$$D(s,t) = \frac{R \times \delta}{d_{s,t}} \quad (6) \qquad\qquad \delta = 2 \times \alpha \times \overline{c_a} \times |E| \times \sum\nolimits_{(s,t) \in N^2} \frac{H_{s,t}}{d_{s,t}} \quad (7)$$

The single and multi-objective EAs have similar configurations. The individuals that populate the initial populations are randomly generated, with link weights taken from a uniform distribution within the mentioned range. To generate new individuals, the EAs use several reproduction operators, making possible to obtain new offspring by means of parents' recombination:

- 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 by the previous integer value, with equal probabilities, within the allowed range;
- Uniform crossover: 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.
- Two point crossover: two crossover points are chosen and the contents between these points are exchanged between two mated parents.

Table 1. Single objective optimization and traditional weighting schemes congestion comparison.

Topology		Demands	Unit		InvCap		Optimized Weights	
Nodes	Links		ECMP	DEFT	ECMP	DEFT	ECMP	DEFT
6	9	0.40	229.699	229. 699	731.399	674.896	2.000	1.891
30	55	0.50	317.725	317.725	494.918	393.095	3.433	2.953
30	110	0.33	260.541	260.541	498.361	267.707	2.702	2.789
30	110	0.40	426.759	426.759	717.950	557.718	20.678	6.138

5. Experimental Results

Single objective and multi-objective problems were used to evaluate the performance of the two alternative traffic load balancing strategies, ECMP and DEFT, against largely used weights configuration schemes. The following sections describe such experiments.

5.1. Single Objective Optimization

A set of single objective optimization problems were used to compare the ECMP and DEFT congestion optimization. For comparison purposes, the congestion levels obtained by two commonly used traditional weighting schemes are also presented, namely, *InvCap*, where each link weight is set to a value inversely proportional to its capacity, and *Unit*, where every link weight is set to one (i.e. the best paths are the ones with the minimum number of hops). Each configuration was run 10 times, with a stopping criterion of 1000 iterations. Some representative results taken from the experiments set are presented in Table 1. As MOEAs are non-deterministic, the network congestion values are the mean of all 10 runs.

The illustrative results show that both ECMP and DEFT optimization outperform the congestion levels attained by the traditional weighting schemes Unit and InvCap. Although this result is not surprising in itself, it displays the enormous gap between them, and clearly states the advantages of using weight optimization TE. In fact, as mentioned before, congestion values higher than 10 2/3 imply that the network topology is unable to support the considered demands. The values observed in Table 1 mean that the heuristics provided extremely poor results, which would translate into severe congestion and packet loss affecting the network operating conditions.

A direct comparison of the two optimization schemes allow to recognize that, in almost all case scenarios, DEFT can reduce the network congestion to lower values than those attained by the ECMP load balancing optimization. It is important to note that some results, when compared, may be misleading due to the heavy penalization applied to each link by the congestion measure Φ. In the case scenario with a 30 nodes topology and 110 links, for a D0.4 demand matrix, the mean optimized congestion values are 20.678 and 6.138 for ECMP and DEFT respectively. The difference between the obtained values do not linearly translate into link utilization. In fact, the difference between the congestion values is mostly due to three links being over congested, and whose over-usage is heavily penalized by the convex function Φ. The NetOpt framework provides a set of tools that enable the network administrator to perform such analysis and to make sustained choices of link weights configurations.

There are however some cases, such as the optimization for the topology with 30 nodes and 110 links, for a traffic matrix D0.33, where DEFT is unable to provide a better congestion level than the one achieved by the ECMP optimization. This does not mean that no better solutions exist, it rather expresses a need to enlarge the search space of the EA, by increasing the w_{max} value, and adequately adjust the p value of the Γ function. There are advantages and disadvantages in the use of relatively low w_{max} values. As stated before, a low w_{max} value, or a low amplitude range of weights if w_{min} is not 1, reduces the search space and potentiates equal cost multi-path. However, a reduced amplitude range of weights diminishes the DEFT solution diversity as the extra length of the path $h^t_{u,v}$ varies in range 1 to $(w_{max} - 1)$.

5.2. Multi-Objective Optimization

The traditional routing problem deals with the selection of paths to route given amounts of demands between origin and destination routers, and most previous studies assume 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. Traffic demands may follow periodic and foreseeable changes resulting in matrices with distinct levels of demands for distinct time periods or demand matrices that, despite inducing similar overall levels of traffic, may have quite distinct source-destination individual entries. It is although possible to seek for weight configurations that permit an acceptable level of congestion for a given set of considered traffic matrices, by resorting to a multi-objective optimization as introduced in Section 2.3.

A Trade-off Analysis (TOA) measure, presented in Table 2 and Table 3 is used to compare multi-objective results. For a solution in a Pareto front, and given a value of α, the TOA measure is obtained with the aggregated sum $\alpha \times \Phi_1 + (1-\alpha) \times \Phi_2$. Parameter α, which takes values in the range $[0,1]$, enables to define different trade-offs between the objectives, and to compare results on a partially ordered space. Only α values of $0.25; 0.5; 0.75$ are here presented.

Table 2 presents the congestion values for Φ_1 and Φ_2 with minimal trade-off between the objectives. The results, for the multi-objective optimization problems, illustrate that, similarly to the observation made for the single objective optimization problems, DEFT optimization delivers network configuration solutions that are at least as good as OSPF with ECMP optimized solutions. In a sense, DEFT can be considered as an extension of ECMP, as in addition to directing traffic through shortest paths, it makes use of other available non-shortest paths. In some cases, with the applied optimization configurations, the two schemes produce equally good solutions for both states of the network, with distinct traffic demand matrices. In the first scenario, DEFT and ECMP deliver solutions that have similar quality. Yet, in the 30 nodes with 55 links, for two D0.5 matrices, the DEFT optimization scheme offers a solution with a $\alpha = 0.5$ trade-off congestion value of 3.997 against a ECMP solution with a 7.002 overall congestion value.

The performance gap between the two compared load balancing schemes is more evident in demanding conditions where more traffic needs to be routed on the network, as can be observed in Table 3. As an illustrative example, for a topology with 30 nodes and 55 edges, in which two distinct traffic demand matrices D0.5 need to be accommodated in two distinct moments in time, DEFT presents a mean congestion value of 9.906, considering a trade-off $\alpha = 0.75$, while ECMP solutions present a mean congestion value of 14.523.

Table 2. Congestion cost with minimal trade-off values for $\alpha = 0.5$.

Nodes	Links	Demands	ECMP			DEFT		
		D_1 & D_2	Φ_1	Φ_2	Trade-off	Φ_1	Φ_2	Trade-off
30	55	0.3	1.452	1.343	1.397	1.406	1.334	1.369
30	55	0.4	2.142	2.206	2.374	2.090	1.374	2.232
30	55	0.5	6.656	7.347	7.002	3.470	4.524	3.997
30	110	0.3	2.528	2.570	2.549	2.197	2.260	2.228

Table 3. Mean trade-off congestion values for $\alpha = 0.25; 0.5; 0.75$.

Nodes	Links	Demands	0.25		0.5		0.75	
		D_1 & D_2	ECMP	DEFT	ECMP	DEFT	ECMP	DEFT
30	55	0.3	2.329	1.523	2.058	1.514	2.329	1.505
30	55	0.4	5.127	3.809	4.732	3.433	4.336	3.057
30	55	0.5	17.959	12.171	16.241	11.039	14.523	9.906
30	110	0.3	7.623	4.395	8.516	4.258	9.408	4.120

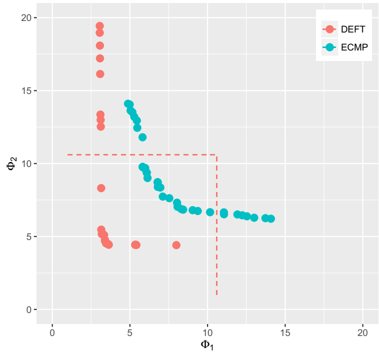

Figure 2. Comparison of ECMP and DEFT Pareto fronts.

To enable a better interpretation of the obtained results, two Pareto fronts from ECMP and DEFT experiments are shown in Figure 2. Both sets of non-dominated solutions are from the same MOOP, with 30 nodes and 55 edges, and two D0.5 traffic demand matrices. The figure clearly shows that DEFT load balancing optimization was able to attain configurations which translate into reduced congestion levels, when compared with ECMP optimized congestion, even when traffic patterns and requirements shift from one traffic matrix to the other. The dotted red line in Figure 2 identifies the threshold beyond which the network start to experience a large percentage of package loss due to a high congestion level.

6. Conclusions

This work has highlighted the benefits of using the routing configurations provided by Evolutionary Algorithms optimization engines with ECMP and DEFT load balancing alternatives. Resorting to single objective evolutionary algorithms, it was possible to observe the benefits of using optimized configuration weights, for both ECMP and DEFT load balancing mechanisms, when compared with the configurations provided by commonly used weight setting heuristics.

A direct comparison between the congestion levels provided by DEFT and ECMP in this context showed that, for all case scenarios, DEFT allows to achieve a congestion level at least as good as the one provided by ECMP optimization solutions and, in many cases, outperforms it. In situations where two distinct traffic demands need to be considered, for example night and day traffic variations, an optimized DEFT configuration is also able to provide a better usage of the network infrastructure, enabling lower congestion levels in both time periods. Future work will address improvements to the DEFT proportional flow splitting mechanism, by fine tuning the configuration parameters, as well as the implementation of a DEFT enabled SDN controller. New multi-objective goals will also be explored, considering, for example, the minimization of end-to-end delays.

It is also important to highlight that this study was made resorting to an optimization framework developed by the authors, which was recently complemented with the DEFT load balancing mechanism. This EA based optimization framework can be a valuable tool for network administrators, allowing to attain optimized routing configurations for the network infrastructures.

Acknowledgments

This work has been supported by COMPETE: POCI-01-0145-FEDER-007043 and FCT - Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2013.

References

[1] N. Feamster, J. Rexford, and E. Zegura. The road to SDN. *Queue*, 11(12):20:20–20:40, December 2013.
[2] S. Jain at al. B4: Experience with a globally deployed software defined WAN. *In ACM SIGCOMM, Aug. 2013*.
[3] Cariden Technologies. Building Traffic Matrices: Introduction to MATE Flow Collection. White Paper - Version 2. (October 2012)
[4] J. Moy. OSPF Version 2. RFC 2328 (Standard), April 1998. Updated by RFC 5709.
[5] C. Hopps. Analysis of an Equal-Cost Multi-Path Algorithm. IETF RFC 2992, DOI 10.17487/RFC2992, November 2000,https://tools.ietf.org/html/rfc2992.
[6] D. Xu, M. Chiang, and J. Rexford. DEFT: Distributed exponentially-weighted flow splitting. *Proc. IEEE Conf. Comput. Commun.*, pp. 71-79, 2007.
[7] N. McKeown at al. Openflow: Enabling innovation in campus networks. *SIGCOMM Computer Communication Review*, 38(2):69–74, March 2008.
[8] C. Coello. A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques. Knowledge and Information Systems 1(3), 129-156, 1999.
[9] S. Srivastava, G. Agrawal, M. Pioro, and Medhi, Determining link weight system under various objectives for OSPF networks using a Lagrangian relaxation-based approach, IEEE Trans. Netw. Serv. Manage., vol. 2, no. 1, pp. 9-18, Nov. 2005.
[10] A. Sridharan, R. Guerin, and C. Diot, Achieving near-optimal traffic engineering solutions for current OSPF/IS-IS networks, IEEE/ACM Trans. Netw., vol. 13, no. 2, pp. 234-247, Apr. 2005.
[11] B. Movsichoff, C. Lagoa, and H. Che. Decentralized optimal traffic engineering in connectionless networks, IEEE J. Sel. Areas Commun., vol. 23, no. 2, pp. 293-303, Feb. 2005.
[12] B. Fortz. Internet Traffic Engineering by Optimizing OSPF Weights. In Proceedings of IEEE INFOCOM, pages 519-528, 2000.
[13] M. Rocha, P. Sousa, P. Cortez, and M. Rio. Quality of Service Constrained Routing Optimization Using Evolutionary Computation. Applied Soft Computing, 11(1):356-364, 2011.
[14] V. Pereira, M. Rocha, P. Cortez, M. Rio, and P. Sousa, A Framework for Robust Traffic Engineering using Evolutionary Computation, 7th International Conference on Autonomous Infrastructure, Management and Security (AIMS 2013), Barcelona, Spain, Springer, LNCS 7943, pp. 2-13, 2013.
[15] P. Sousa, M. Rocha, M. Rio, and P. Cortez, Automatic Provisioning of QoS Aware OSPF configurations, Journal of Networks (JNW), 2(2):1-10 , Academy Publisher, April 2007. ISSN: 1796-2056.
[16] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans. Evolutionary Computation, 6(2):182-197, 2002.
[17] E. Zitzler, M. Laumanns, and L. Thiele. Spea2: Improving the strength pareto evolutionary algorithm. Technical report, 2001.