Traffic Engineering With Three-Segments Routing

Vítor Pereira, Miguel Rocha, and Pedro Sousa

***Abstract*—Segment Routing (SR) is a new fertile ground for Traffic Engineering (TE). By decomposing forwarding paths into segments, which specify a list of intermediate delivery points that a packet must visit on its way to the final destination, SR improves TE tasks and enables new solutions for the optimization of network resource utilization. This work proposes an Evolutionary Computation approach that enables Path Computation Element (PCE), or Software-defined Network (SDN) controllers, to optimize SR configurations for improved traffic distribution. Furthermore, we present a robust semi-oblivious method to address the variability of traffic requirements as well as alternative approaches to ensure a good network performance after link failures. In all cases, the optimization of network resource utilization is achieved using at the most three segments to configure each SR path. Moreover, all proposed optimization methods are made publicly available in a optimization framework developed by the authors.**

***Index Terms*—Segment routing, network optimization, evolutionary computation.**

I. Introduction

SEGMENT routing (SR) [1] is a Software-Defined Networking (SDN) [2] technology proposed by the IETF to address identified limitations in Multi-Protocol Label Switching (MPLS), with particular emphases on scalability, simplicity, and ease of operation [3]. SR implements the source routing paradigm, where a source node directs incoming packet flows across the network by specifying a list of intermediate delivery points that a packet must visit on its way to the final destination. This is achieved by breaking down routing paths into smaller parts, called segments, enabling better network utilization and improving traffic engineering tasks.

Because it is built over already existing Interior Gateway Protocols (IGP), such as Open Shortest Path First (OSPF) [4] and Intermediate System to Intermediate System (IS-IS) [5], SR inherits some of their advantages such as scalability, the even load balancing of traffic between equal-cost paths, that is, Equal-Cost Multi-Path (ECMP) [6], and the automatic rerouting of traffic after a link or node failure. Contrasting with MPLS, where labels are defined per Forwarding Equivalent Classes (FEC), SR segments are defined per nodes, which reduces the number of entries in routing tables and simultaneously simplifies tunneling management. SR is enabled by a small number of IGP and Border Gateway Protocol (BGP) [7] extensions and can be applied both in MPLS and IPv6 (SRv6) architectures.

The flexibility in defining forwarding paths and the simplified task management of SR provide a fertile ground for Traffic Engineering (TE) and, in particular, to optimize network performance while reducing operational costs. Exploring such advantages, this work extends previous work by the authors and proposes a TE approach for SR able to optimize the utilization of network resources. Most previous work on SR TE aim to minimize the Maximum Link Utilization (MLU) by forcing traffic to traverse some intermediate nodes (one or more) between source and destination. However, the MLU is oblivious to the existence of unused or poorly utilized links and is unable to characterize fully the distribution of traffic in the entire network. Furthermore, the MLU may be trapped to the existence of bottleneck links due to poor network design, changes in the network topology (e.g., link failure), or biased traffic requirements. In such cases, the MLU solely reflects the local link utilization becoming inefficient as an optimization metric. This work aims to optimize the distribution of traffic on the entire network minimizing an alternative congestion metric and with relatively small alterations from shortest path routing. Optimized shortest path routing configurations are already able to achieve a few percents from optimality [8], which can be further enhanced resorting to the flexibility provided by SR while forwarding the majority of traffic along shortest paths from source to sink.

One important feature of our proposal is that the optimization is achieved using at the most three segments to configure edge-to-edge routing paths, reducing paths encoding overhead while meeting routers limitation of label stack depth. We also propose adaptive robust TE methods to respond to network volatile operational conditions. Variations in traffic necessities, as well as fault events, such as link failures, are conditions that undermine network performance. Ergo, the optimization model is extended to enable traffic load balancing corrections between end-to-end SR paths, allowing to better accommodate changes in traffic necessities. Additionally, typical responses to topology link failures only aim to reestablish loop-free connectivity between affected routers, without any consideration on how such failures impact network performance. In such a context, we compare distinct alternatives to improve resource utilization after the network has recovered from the fault, weighing their advantages and limitations.

Manuscript received February 20, 2020; revised April 28, 2020; accepted May 3, 2020. Date of publication May 7, 2020; date of current version September 9, 2020. This work has been supported by FCT—Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020. The associate editor coordinating the review of this article and approving it for publication was K. Xue. *(Corresponding author: Vítor Pereira.)*

Vítor Pereira and Miguel Rocha are with the Centre of Biological Engineering, Department of Informatics, University of Minho, 4704-553 Braga, Portugal (e-mail: vpereira@ceb.uminho.pt; mrocha@di.uminho.pt).

Pedro Sousa is with the Centro Algoritmi, Department of Informatics, University of Minho, 4704-553 Braga, Portugal (e-mail: pns@di.uminho.pt).

Digital Object Identifier 10.1109/TNSM.2020.2993207

The majority of the addressed problems are NP-hard involving the simultaneous optimization of more than one objective and, consequently, there is no unique solution but a set of solutions with distinct trade-offs between the objectives. Multi-Objective Evolutionary Algorithms (MOEAs) are nature-inspired methods that mimic the metaphor of natural biological evolution. They maintain a population of solutions whose interactions, matting and mutations, drive the optimization process across the search space, providing in a single run a set of solutions that populate a Pareto front. Evolutionary Computation techniques have additional advantages that justify their utilization. For example, they are not bounded to linearity constraints, they are conceptually simple and can be applied to virtually any optimization task, notably to the optimization of network configurations [9], [10].

The remainder of the document is organized as follows. Section II presents a brief introduction to Segment Routing TE, and Section III discusses some related work. Next, Section IV introduces and describes the proposed optimization model. In Sections V and VI, the model is extended to acknowledge networks changing conditions, respectively, traffic variations and the possibility of link failures. Section VII presents simulation results for each optimization methods, and Section VIII draws the main conclusions.

II. Traffic Engineering in Segment Routing

A Segment Routing path is composed of a succession of segments which represent instructions, topological or service-based. While a service instruction pinpoints a service or a Network Function Virtualization (NFV) [11] where a packet should be delivered, a topological instruction determines a path along which a packet should be forwarded. A segment is identified with a Segment Identifier (SID)[1] with global or local significance in the network. Similarly to labels in MPLS, global segments are advertised to all network nodes but, to perform such a task, SR does not require a Label Distribution Protocol (LDP) or a Service Reservation Protocol (SRVP). Segments may be advertised by the IGP or BGP extensions that enable the signaling of segments or, alternatively, the distribution of segments may be centrally assured by a controller or a Path Computation Element (PCE). The IETF Segment Packet Routing in Networking (SPRING) working group defines two main types of topological segments, Prefix and Adjacency Segments [12]:

- A **Prefix-SID** is a segment that refers to a specific network prefix. Prefix-SIDs are always global within an IGP domain and refer to the shortest path computed by IGP to the related prefixes. A packet that enters an IGP area with an active Prefix-SID will be forwarded along the ECMP-aware shortest path to the prefix. Since a prefix could represent a node or a group of nodes within an IGP domain, Prefix-SIDs are further divided into Node SIDs and Anycast SIDs.
- An **Adjacency-SID** represents a local segment (interface) to a specific SR node. Each router assigns a locally significant segment ID for each of its IGP adjacencies.

[1] The terms segment and segment identifier will be used interchangeably.

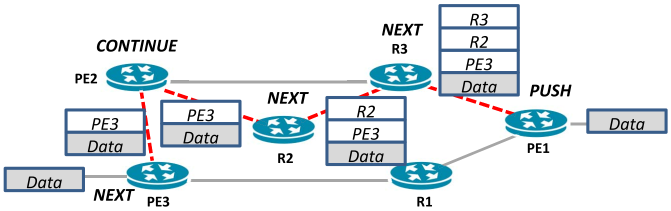


Fig. 1. Example of SR operations.

Any edge to edge forwarding path can be decoupled as a combination of node SIDs and adjacency SIDs. Three basic operations on SIDs and segment lists enable packets forwarding on generic SR data planes: *PUSH*, *NEXT*, and *CONTINUE*. A *PUSH* operation inserts a new SID on the top of the segment list. When a new packet arrives at an SR domain edge router, the routing policy (an ordered list of segments) is inserted into the packet's header by a series of *PUSH* operations. The last inserted segment is the first to be processed and is considered active. In Fig. 1 an example is provided where a data packet is to be routed from a provider's edge router PE1 to a node PE3. When the active segment is completed, a *NEXT* operation makes the following SID active. In the MPLS forwarding plane, *NEXT* corresponds to popping the topmost label, whereas in SRv6, *NEXT* consists of copying the following segment from the SR stack header to the destination address of the IPv6 header. While a segment is not completed, intermediate nodes perform *CONTINUE* operations, stating that the current segment is to remain active. This operation is performed by intermediate nodes on shortest-path segments. As a consequence, in SRv6 implementations, a *CONTINUE* operation corresponds to a plain forwarding action, whereas in SR-MPLS, a *CONTINUE* operation is implemented as a *SWAP* of the top label. When a packet leaves the SR domain, a NEXT operation removes the last SID, and the packet is processed according to conventional IP routing.

A key advantage of SR is that intermediate nodes only need to know globally distributed SIDs contrasting with other SDN implementations, such as OpenFlow [2], which are required to maintain per-flow states.

III. Related Work

A. SR Optimization

The problem of determining optimal configurations using the least number of segments is one of the most important topics in Segment Routing. The authors in [13] propose an integer programming algorithm as well as a heuristic approach to address the traffic engineering of packet networks with SR. By defining a maximum label stack depth, the heuristic begins by distributing traffic on SR paths configured with one SID, which are IGP shortest-paths, and consecutively assigns traffic on SR paths with increasing segment list depth. In [14], the authors propose offline, online, and traffic oblivious optimization algorithms to minimize the MLU for a given IGP link weights configuration. The proposal is to find k intermediate nodes on non-shortest paths to the destination that enable to optimize the network traffic load balancing. This approach, uses a fixed number of nodal segments to configure each SR path, which may result in longer hop-by-hop paths that can contribute to a higher delay. In [15] the author introduce a bounded stretch constraint to limit shortest-path distances between intermediate nodes, avoiding longer paths. In [16], the authors propose a local search algorithm to minimize the MLU by quickly rearranging a few forwarding paths. Such an approach has the benefit of acknowledging traffic variations in the network environment. Alternatively, instead of considering the traffic necessities of a single moment in time, some authors propose oblivious approaches, where a routing configuration needs to be assessed irrespective of the realization of a set of feasible traffic necessities. In [14], the authors derive a set of split variables, independent from any predefined traffic requirements, to split traffic to intermediary nodes, based on Game Theory techniques. In [17], the authors estimate upper and lower bounds of feasible traffic necessities to be used as constraints in an oblivious routing problem. In both cases, a linear programming algorithm delivers solutions where SR paths are configured with a predefined fixed number of node segments, that is, with a segment stack length of 2 and 3 segments, respectively. Although such approaches keep the network configuration stable, they unavoidably suffer from low optimality during most of the operational time. In [18], the authors propose DEFO, a declarative and expressive approach to control forwarding paths in carrier-grade networks, a two-layer architecture composed of an IGP connectivity and an optimization layer. The last translates high-level goals, notably the MLU minimization, into compliant network configurations by mixing a constrained programming solution with a local search technique to find middle point routing solutions quickly. More recently, SR4CG [19] proposes to leverage column generation to obtain SR configurations within a reduced computational time. The obtained solutions, with a configurable maximum segment list depth, make use of node and adjacency segments to forward traffic and obtain near-optimal routing configurations. Extensive surveys on SR-TE, organized by categories, can be found in [20], [21].

B. Link Failure Recovery

Segment Routing is built over already existing IGPs and takes advantage of a multiplicity of their features. One of those features is the automatic rerouting of traffic after a link failure. Upon a link failure, the IGP recomputes all shortest-paths, and nodal segments are automatically repaired without any additional intervention. However, the time required to detect a link failure, propagate the fault, and recompute the shortest-paths can be excessively long and, therefore, recovery paths should preferably be pre-computed and installed in the data plane. The Topology-Independent Loop-Free Alternate (TI-LFA) [22] follows this strategy and provides local protection for IGP SIDs with a sub-50msec loss of connectivity. TI-LFA, proposed by the SPRING working group, pre-computes backup paths along the post-convergence path from the Points of Local Repair (PLR) to all possible destinations. However, TI-LFA demonstrates some limitations. The authors in [23] showed that TI-LFA works only with a limited number of failures. For two or more simultaneous link failures, packets may quickly become stuck in a loop. Also, even though TI-LFA solves the problem of loss of connectivity, it does not offer any guarantee regarding the network performance after a link failure.

The authors in [24] propose to use robust disjoint SR paths, pairs of paths that are constructed to remain disjoint in the eventuality of a set of failures defined by an operator. In [25], a linear programming model is proposed to optimize recovery paths in *k*-segments routing. The idea is to explore the IGP automatic rerouting of traffic by optimizing the initial SR configuration, for known traffic demands, such that there is sufficient link bandwidth to handle any single link failure. In [26] the authors argue that the performance of a routing scheme should be measured not only in terms of fault-tolerance but also with respect to the resulting congestion. As such, the work proposes a Congestion And Stretch Aware static fast rerouting (CASA), a deterministic algorithm which relies on a combination of combinatorial design and arborescences to reroute traffic after a link failure.

IV. SALP-SR Optimization Model

The main idea of the present approach is to attain a near-optimal resource utilization using forwarding paths that deviate at the most one hop from the shortest-path. Such a strategy takes advantage of the underlying IGP features, notably of the ECMP traffic load balancing, to attain a better utilization of available resources. A single hop deviation from the shortest-path between a node s and a node t implies the usage of a non-shortest path link identified as an adjacency segment. As a result, the Single Adjacency Label Path Segment Routing (SALP-SR) forwards traffic flows using only SR paths with the following configuration format alternatives:

- **1-Segment**: The label path is configured with a unique Node SID, the SID of the destination node t, `[t]` (Figure 2(b));
- **2-Segment**: The label path is configured with a Node SID and an Adjacency SID. In this case, there are two possible configurations:
  1) `[(s,u); t]`, the adjacency segment starts at the source node s, and the Node SID identifies the destination node t (Figure 2(c));
  2) `[v; (v, t)]`, the adjacency segment ends at the destination node t, and the Node SID identifies the start node of the adjacency segment, v (Figure 2(d));
- **3-Segment**: The label path is configured with a Node SID, an Adjacency SID, and a Node SID, `[u; (u, v); t]`, where (u, v) is the adjacency segment (Figure 2(e)).

The model relies on optimized IGP configurations and on the computation of how much traffic is to be forwarded along adjacency segments to accomplish a near-optimal network resource utilization. For an easier understanding of the devised TE proposal for SR, we next present an illustrative example.

Figure 3(a) presents part of a network that runs the OSPF or IS-IS routing protocol, the computed shortest paths from a node s to a node t and respective hop-by-hop ECMP traffic splitting ratios. The fraction of traffic from s to t arriving at node u is 2/3 (1/2 + 1/2 × 1/3) of the intended traffic.

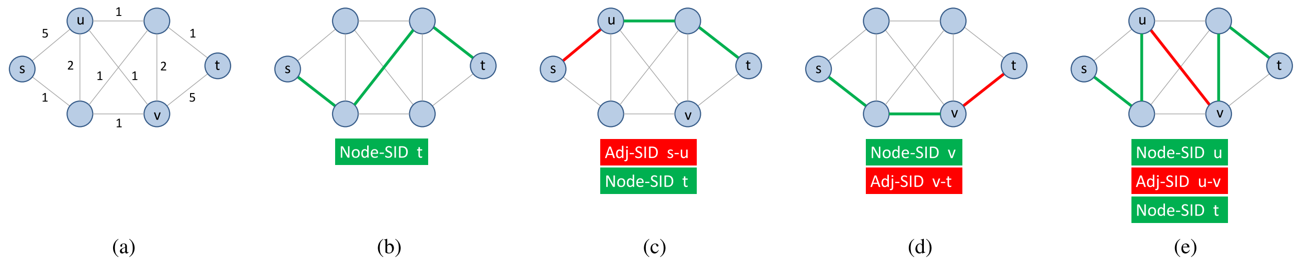


Fig. 2. SR path configuration formats.

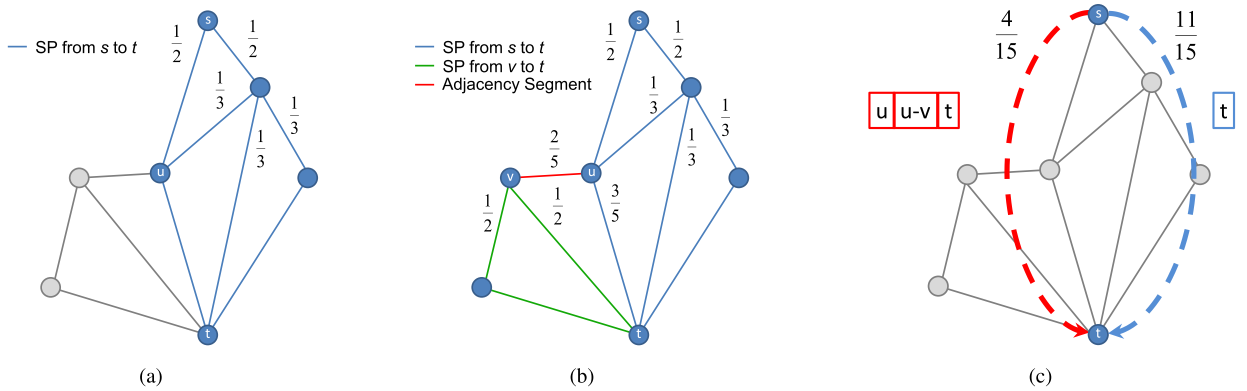


Fig. 3. Three-Segments Routing example.

Link (u, t) is, however, unable to accommodate such volume resulting in packet loss and increased delays. To address such a scenario, a portion of the traffic arriving at u can be forwarded using an adjacency segment that does not belong to any shortest path from u to t. In such a case, and considering already computed optimized hop-by-hop load balancing ratios at u, Fig. 3(b), 2/5 of the traffic from s to t arriving at u, that is 4/15 (=2/5 $*$ 2/3) of the total traffic, is forwarded using the adjacency segment (u, v). An SR path configuration that prevents link (u, t) to become congested is shown in Fig. 3(c).

A trivial but important consequence of SALP-SR is that all label stacks have a maximum depth of three labels or SIDs. In our approach, the problem of computing the amount of traffic to be routed over non-shortest path links, identified by an adjacency segment, is solved by adopting the exponential traffic load balancing function proposed by the Distributed Exponentially-weighted Flow Splitting (DEFT) [27] routing protocol. The load balancing strategy assigns flows to a next-hop with a probability that decreases exponentially with the additional length of the path. The big advantage of DEFT when compared with other unequal load balancing mechanisms is that it only requires the link weights configuration to compute traffic splitting ratios.

A. Hop-by-Hop Traffic Load Balancing

The mathematical model of a data communications network for the general routing problem is defined as follows. Consider the capacitated directed graph $G = (N, A)$, where N and A denote the sets of nodes and arcs. The nodes and arcs represent routers and links, the last with a capacity constraint c_a, for all $a \in A$. Table I lists the mathematical symbols used in this paper. The distance from a node u to a node t, when traffic is routed through a node v adjacent to u ($u, v, t \in N$), is expressed as $\delta_v^t + w_{u,v}$, where δ_v^t is the shortest distance from the next-hop v to t, and $w_{u,v}$ is the weight of the link $(u, v) \in A$. The extra length of the path from u to t using v, when compared to the shortest path δ_u^t, is obtained by Eq. (1), and denoted as $h_{u,v}^t$. The proportion of traffic P from u to t routed using the link (u, v) is computed using Eq. (3), where the function Γ, Eq. (2), exponentially penalizes the utilization of paths with greater extra length $h_{u,i}^t$, $(u, i) \in A$.

TABLE I
TABLE OF SYMBOLS

Symbols	Description
N, A	Set of nodes and arcs.
c_a, l_a, u_a	Capacity, load, and utilization of an arc $a \epsilon A$.
$w_{u,v}$	Weight of the link from u to v.
δ_u^t	Shortest-path distance from u to t.
$h_{u,v}^t$	Extra length of the path from u to t using next hop v.
$\Gamma\left(h_{u,v}^t\right)$	Extra length penalizing function.
$P\left(h_{u,v}^t\right)$	Hop-by-hop traffic splitting.
p	Node-p value used to correct traffic load balancing.
$\mathcal{A}^{s,t}$	Set of adjacency segments in paths from s to t.
$\mathcal{P}^{s,t}$	Set of shortest-paths from s to t.
Φ_a	Congestion cost of link $a \in A$.
$\Phi*$	Normalized network congestion value.
$\mathcal{SR}^{s,t}$	Set of SR parallel paths from s to t.
$\alpha_{u,v}^{s,t}$, $\alpha_{sp}^{s,t}$	SR parallel paths traffic splitting ratios.
D_i	The i-th traffic demands matrix.
$d^{s,t}$	Traffic demands from s to t.
$d_k^{s,t}$	Portion of $d^{s,t}$ forwarded using the SR Path $k \in \mathcal{SR}^{s,t}$.

$$h_{u,v}^t = \delta_v^t + w_{u,v} - \delta_u^t. \tag{1}$$

$$\Gamma\left(h_{u,v}^t\right) = \begin{cases} e^{-h_{u,v}^t}, & \text{if } d_v^t < d_u^t, \\ 0, & \text{otherwise}. \end{cases} \tag{2}$$

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

B. Load Balancing Between Parallel Paths

SALP-SR enables to define more than a path between the same source/destination pairs (*s*, *t*), each forwarding a fraction of the intended traffic between the two nodes. To compute the traffic splitting ratios between parallel paths, we start by defining, for each pair of nodes (*s*, *t*), the set $\mathcal{A}^{s,t}$ of non-shortest path links (*u*, *v*) adjacent to a shortest-path from *s* to *t*, that is, *u* is a shortest path node and the extra length $h^t_{u,v}$ is positive. Each adjacency link $(u, v) \in \mathcal{A}^{s,t}$ originates a distinct SR path from *s* to *t*. The fraction of traffic from *s* to *t* to be routed using the adjacency segment (*u*, *v*), $\alpha^{s,t}_{u,v}$, is computed using Eq. (4), which mirrors the procedure described in Fig. 3. For each path belonging to the set of all shortest paths from *s* to *u*, $\mathcal{P}^{s,u}$, the computation of the hop-by-hop splitting ratios $P(h^t_{i,j})$ of each outgoing link (*i*, *j*), allows to obtain the fraction of traffic from *s* to *t* arriving at *u* and, subsequently, the load balancing ratio $\alpha^{s,t}_{u,v}$ of the SR path configured with the adjacency segment (*u*, *v*). The remaining traffic, $\alpha^{s,t}_{sp}$ of the intended traffic from *s* to *t* (Eq. (5)), is forwarded using shortest-path routing, and the SID of destination node *t* encodes the forwarding path.

$$\alpha^{s,t}_{u,v} = \begin{cases} P\left(h^t_{u,v}\right), & u = s, \\ P\left(h^t_{u,v}\right) \sum_{p \in \mathcal{P}^{s,u}} \left(\prod_{(i,j) \in p} P\left(h^t_{i,j}\right)\right), & u \neq s. \end{cases} \tag{4}$$

$$\alpha^{s,t}_{sp} = 1 - \sum_{(u,v) \in \mathcal{A}^{s,t}} \alpha^{s,t}_{u,v}. \tag{5}$$

By applying both equations in the illustrative example, Fig. 3, the traffic splitting ratios are 4/15 for the label path with the adjacency SID, and the remaining traffic, 11/15, to the shortest-path, that is, the path with only the SID of the node *t*, Fig. 3(c).

C. Congestion Metric

Contrary to most approaches that consider the minimization of the MLU, we chose to use as main objective the minimization of the network congestion measure Φ proposed by Fortz and Thorup [8]. The congestion measure Φ assigns to each link $a \in A$, a congestion cost Φ_a as a function of the utilization $u_a = l_a/c_a$, i.e., how close the load l_a is to the link capacity c_a.

Given a Traffic Matrix (TM) *D*, a model of the volume $d^{s,t}$ of traffic flowing between each origin-destination pairs (*s*, *t*), the goal is to find a routing configuration *S* that enables to minimize the sum of all link costs Φ (Eq. (6)) while meeting the traffic forwarding requirements.

$$\Phi(S) = \sum_{a \in A} \Phi_a(l_a). \tag{6}$$

The link cost metric Φ_a, whose plot is shown in Fig. 4, is a piecewise convex continuous function that forces the optimal utilization of resources by exponentially penalizing over-utilized links and favoring the use of less congested ones. The evaluation of a routing configuration $S =< a_1, \ldots, a_m >$ is obtained by computing the sum of all links utilization Φ_a, Eq. (6), when the routing model described in Section IV-D is applied to the network configured with *S*. To enable results comparison, the sum of all links congestion costs Φ is normalized as Φ*, Eq. (7), where $dist_1(s, t)$ is the minimum number of hops between nodes *s* and *t* and $d^{s,t}$ the traffic requirements between the two nodes. The lower bound of the normalized congestion metric Φ*, for not null traffic requirements, is equal to 1, and is obtained when all link loads are under 1/3 of their capacity. Additionally, when all links have a load equal to the limit of their capacities, Φ* is equal to 10.67, a value we here consider as being the threshold of congestion for acceptable network operational conditions [8].

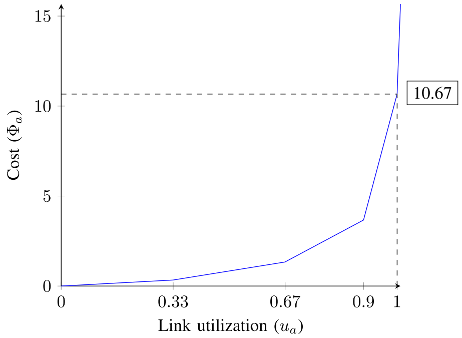


Fig. 4. Link utilization cost function Φ_a [8].

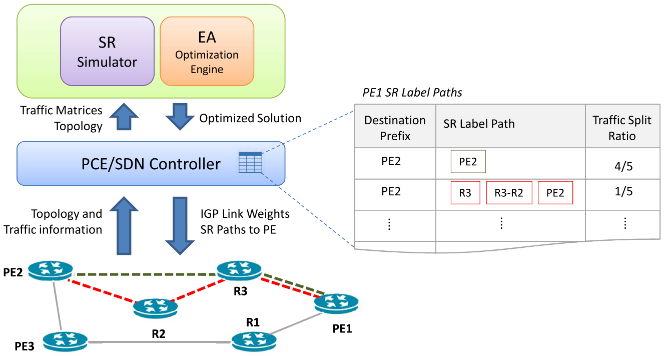
$$\Phi^*(S) = \frac{\Phi(S)}{\sum_{(s,t) \in N \times N} (dist_1(s, t) \times d^{s,t})}. \tag{7}$$

There are several reasons that justify the use of Φ* instead of the MLU for the main congestion metric. Firstly, Φ* reflexes the utilization of all network links as a single real value and not only the utilization of the link with the highest utilization. The MLU is oblivious to the existence of unused or poorly utilized links and is unable to fully characterize the distribution of traffic in the entire network. Secondly, any change to a configuration alters the congestion value Φ*, which does not always happen with the MLU. Furthermore, the MLU may be trapped to the existence of bottleneck links, and, in such cases, it solely reflects the local link utilization becoming inefficient as an optimization metric.

The cost function Φ*, initially proposed to optimize link weights configurations for OSPF and IS-IS routing protocols, has been extensively used for a vast number of networking optimization problems [27], [28]. However, when routing constraints are applied, such as determining optimized link weights configurations, the optimization of the Φ* metric often translates into NP-hard problems that require meta-heuristics, such as Evolutionary Algorithms (EAs) [29], [30], [31], to be solved.

D. Optimization Model

The first objective of SALP-SR is to find a configuration of integer values $S = \langle a_1, a_2, \ldots, a_m \rangle$, where *m* is the number of links, which optimizes a network performance objective. The objective can be the minimization of the congestion metric Φ*, but might also include other objectives such as, for example, to minimize the MLU. The values a_i are nothing more than a set of IGP's integer link weights $w_{u,v}$, $u, v \in N$, but from which the optimization algorithm also derives the other additional



Destination Prefix	SR Label Path	Traffic Split Ratio
PE2	PE2	4/5
PE2	R3 R3-R2 PE2	1/5
⋮	⋮	⋮

Fig. 5. Conceptual architecture of the optimization model.

settings, SR paths and load balancing ratios between parallel paths. For all nodes u, if the extra length of a path to t using the adjacent node v is a positive integer, $h^t_{u,v} > 0$, then the link (u, v) is used to forward a portion of traffic arriving at u with destination t. Based on the previous formulation, SR paths accept the following configuration formats:

- **3-Segments**: For each non-shortest path link (u, v), in a path from s to t, such that $h^t_{u,v} > 0$ the algorithm produces an SR path configured with two nodal segments and one adjacency segment ([u; (u,v); t]).
- **2-Segments**: When u or v are the source or destination nodes, s and t respectively, the path becomes a 2-segment SR path ([(s,u); t] or [u; (u,t)]).
- **1-Segment**: A shortest-path between s and t is converted into a 1-segment SR path, [t].

E. General Architecture

In this proposal, a configuration that optimizes network resource utilization encompasses: 1) a link weight configuration for the link-state IGP that underlies the SR network, 2) edge-to-edge SR label paths configuration, and 3) load balancing ratios between parallel label paths for traffic with the same destination. A conceptual representation of the SALP-SR optimization model elements is presented in Fig. 5. The optimization model assumes that traffic necessities are known and modeled as a traffic matrix.

The data gathered from direct measurements is sufficient, in most cases, to populate a traffic matrix using estimation techniques [32], [33]. Two main strategies are commonly used to infer TM: (1) indirectly from link loads [34]; (2) directly from sampled flow statistics [35]. Some other approaches take advantage of both strategies [36]. In this work, we are particularly interested in flow-based methods to measure traffic statistics in SDN. OpenFlow switches [37], unlike commodity switches, provide a permissive query API that allows traffic measurements with low overhead. Upon the arrival of a new flow or upon the expiration of a flow entry, *PacketIn* and *FlowRemoved* messages, respectively, are sent by OpenFlow switches to the controller, and thereby enable to compute the link utilization between switches. Exploration works of these features resulted in new proposals for TM estimation in networks with OpenFlow capabilities [38], [39].

The optimization of network resource utilization is performed by a northbound application that integrates an SR simulator and an Evolutionary Algorithm (EA) optimization engine. As depicted in Fig. 5, the northbound application interacts with a PCE/SDN controller to collect topology and traffic-related information required by the optimization process. During the optimization process, new solutions iteratively provided by the EA are evaluated by an SR routing simulator, which implements the SALP-SR routing model, enabling not only to compute the Φ^* congestion measure, but also other measures such as the MLU. At the end of the optimization process, the best solution is saved by the PCE/SDN controller into a local database and installed in the network's data plane. This task involves installing the links weights configuration for the IGP operation, the SR label paths and traffic splitting fractions for provider edges operations. In Fig. 5 traffic originating in PE1 with destination to PE2 is forwarded along two parallel SR paths, a shortest-path and a path that makes use of an adjacency segment. Traffic is divided between both paths on a 4/1 ratio.

V. Semi-Oblivious Congestion Optimization

Estimating edge-to-edge TM in a network is an essential part of many network design and operation tasks, such as capacity planning, routing protocol configuration, network provisioning, load balancing, and anomaly detection. However, a direct and precise measurement of TMs in large IP networks is extremely hard, if not unattainable, due to a large number of source-destination pairs, the high volume of traffic at each link, and the lack of a measurement infrastructure. Moreover, the diversity of services available on contemporary networks, as well as human behaviors and habits, cause traffic variations, both in volume and flow patterns, which may not be accommodated by installed routing solutions. We propose to optimize the initial SR routing configuration considering a set of representative TMs and adjust traffic load balancing to improve network resource utilization. In this approach, the initial SALP-SR configurations are obtained by resorting to multi-objective optimization.

A. Multi-Objective Optimization for Traffic Variation

The notion of optimization for single-objective problems is well-understood and consists of finding the extremum of the objective function. However, the same cannot be said of multi-objective problems. Objectives on a multi-objective problem are frequently conflicting, a solution that improves one of the objectives will eventually degrade at least one of the others. Consequently, there is no single global solution, rather a set of optimal points that populate a Pareto Front. When optimizing an SR configuration for more than one TM, the obtained solutions are not optimal regarding any of the TMs, they instead are a compromise, a trade-off between the objectives.

To acknowledge traffic variations, for example between two periods such as night and day, and find a configuration that enables the network to sustain good functional performance in both periods, we devise a multi-objective optimization defined as follows. For a given network topology and a set of demand matrices D_i, the aim is to find an SR configuration S that simultaneously minimizes the functions $\Phi_i^*(S)$, where $\Phi_i^*(S)$ represents the function $\Phi^*(S)$ considering the traffic demands of matrix D_i.

The set of TMs D_i may represent foreseen traffic necessities or derived as representative of a larger set of traffic requirements. For example, the TMs may be obtained using a clustering algorithm [40], or chosen to bound the set of routable traffic matrices [14], [17]. In this work, we use the first alternative, dividing a set M of TMs into two clusters using K-means [41] clustering algorithm ($k = 2$), and the traffic matrices D_i, $i = 1, 2$, are the cluster's centroids. However, K-means is highly sensitive to the initial placement of centroids upon which the algorithm iterates until finding the "best" centroids. To avoid a poor clustering, in the experiments the initial centroids C_1 and C_2 are two opposite vertices of the higher-dimensional polyhedral which bounds the TMs space M. We first define two auxiliary TM, D_{max} and D_{min}, which, for each pair of nodes (i, j), contain the maximum and minimum traffic requirements between nodes i and j in M. The two traffic matrices C_1 and C_2 are then obtained by swapping with a probability $p = 1/2$ all values from both matrices D_{max} and D_{min}.

B. Correcting Load Balancing

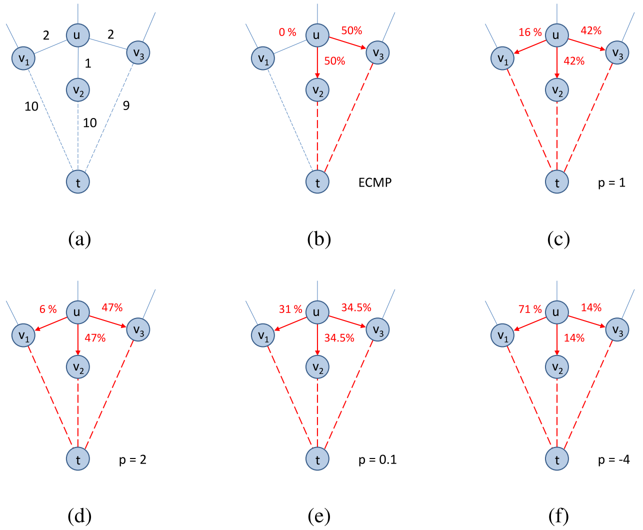
An SR configuration solution that tries to accommodate distinct TMs is not tied to any particular TM, and consequently can always be improved by adjusting load balancing ratios. Furthermore, corrections on load balancing between SR parallel paths can also be exploited to address changes on traffic necessities as well as to respond to congestion states originated by topology faults, like link failures. Next, we extend the proposed routing optimization model with a mechanism to tune traffic load balancing and improve the distribution of traffic for networks on changing conditions.

The extra length $h_{u,v}^t$ is an integer that takes values in range $[0, w_{\max} - 1]$ and defines, at a node u, the fraction of traffic with destination t to be forwarded using the link (u, v). The extra length $h_{u,v}^t$ is, by definition, tied to the installed configuration $S =< a_1, \ldots, a_m >$, Eq. (1), and can not be changed without altering the IGP configuration. However, from a TE perspective, it is desirable to be able to change traffic load balancing while keeping the remaining configurations, IGP link weights, and SR paths, unchanged. In this context, we introduce a new parameter p in Eq. (8) and (9), which enables to leverage the extra length effect in the splitting function. Also, the splitting function Γ is extended to enable non-shortest-path links to forward more traffic than links on shortest-paths.

$$\Gamma\left(h_{u,v}^{t}, p\right) = \begin{cases} f\left(h_{u,v}^{t}, p\right), & \text{if } d_v^t < d_u^t, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

$$f\left(h_{u,v}^{t}, p\right) = \begin{cases} e^{-h_{u,v}^{t} \times p}, & \text{if } p > 0, \\ 1, & \text{if } p = 0, \\ 1 - \frac{p}{h_{u,v}^{t}}, & \text{if } p < 0. \end{cases} \tag{9}$$

The manipulation of the p variable permits to change the amount of traffic forwarded on non-shortest-paths without any change to the installed IGP link weights and forwarding paths. Instead of assigning the same p value to all nodes on the topology, a different value is assigned to each of the $|N|$ nodes. These variables, denoted as *node-p* variables, become a new set of parameters that can be optimized to improve hop by hop traffic load balancing and consequent SR parallel paths traffic load balancing under the scheme presented in Section IV-B.



(a) (b) (c)

(d) (e) (f)

Fig. 6. Node-p value effect on traffic load balancing.

An illustrative example of the effect of distinct node-p values is presented in Fig. 6. Considering the installed link weights shown in Fig. 6(a), the next-hops on a shortest path from u to t are nodes v_2 and v_3. Therefore, using ECMP load balancing, traffic is equally split between these two nodes (Fig. 6(b)). As the extra length of the path from u to t using node v_1 is $h_{u,v_1}^t = 1$, link (u, v_1) is also used to forward a portion of traffic. Different node-p values allow managing the amount of such traffic, which may be lesser (Fig. 6(d)), equivalent (Fig. 6(e)) or greater (Fig. 6(f)) than the amount carried by shortest-path links. In initial configurations, node-p values are set to 1 (Fig. 6(c)).

C. Minimizing the MLU

During SALP optimization procedures for a single TM, we consider as a second objective the minimization of the MLU obtained directly from the solutions link loads. Although the main goal of SALP is to optimize traffic distribution, that is, minimize Φ^*, by defining as second objective the minimization of the MLU the algorithm convergence time is improved.

SALP-SR also enables the minimization of the MLU by optimizing the traffic load balancing between parallel SR paths, that is, by optimizing the traffic splitting ratios $\alpha_{sp}^{s,t}$ and $\alpha_{u,v}^{s,t}$ for s, t, u, $v \in N$. Considering $\mathcal{SR}^{s,t}$ as the set of all parallel SR paths from a node s to a node t, obtained from a SALP configuration solution, we define the linear programming optimization problem as:

$$\text{minimize} \quad MLU \tag{10}$$

$$\text{subject to} \quad \sum_{k \in \mathcal{SR}^{s,t}} \alpha_k^{s,t} = 1, \tag{11}$$

$$d_k^{s,t} = \alpha_k^{s,t} \times d^{s,t}, \tag{12}$$

$$l_a = \sum_{s,t \in N} \sum_k f_a(d_k^{s,t}), \quad k \in \mathcal{SR}^{s,t}; s,t \in N, \tag{13}$$

$$l_a/c_a \leq MLU, \quad a \in A. \tag{14}$$

where $d_k^{s,t}$ is the amount of traffic intended from s to t to be routed using the SR path $k \in \mathcal{SR}^{s,t}$, and $f_a(d_k^{s,t})$ is the amount of such traffic traveling over the link $a \in A$. The load l_a of a link is the sum of all traffic traveling over it, and l_a/c_a is the link utilization ratio.

The SALP routing model can, therefore, be used to optimize both the network traffic distribution and MLU. The last can be achieved: 1) solely resorting to EAs, 2) by solving the LP for a SALP configuration (SALP+LP), or 3) by running a hybrid optimization strategy where all solutions are optimized for the MLU during the evolutionary process (SALP hybrid).

VI. Addressing Single Link failures

The main idea of the IETF proposal for link failure recovery, TI-LFA, is to provide loop-free recovery paths, between the Point of Local Recovery (PLR) and provider's edge destinations, which remain unchanged after the recomputation of shortest-paths due to a link failure. PLRs maintain a table of recovery paths that are deployed as soon as the fault is detected, enabling a sub 50ms response and loss of connectivity. This approach, whose only concern is to provide recovery paths from the PLR on, neglects the impact of the fault on the overall network congestion level.

IGP shortest-paths recomputation affects nodal segments paths and, consequently, the traffic distribution over the available resources. To prevent an eventual congestion problem, operators are forced to over-provision bandwidth to absorb unexpected traffic fluctuations. Moreover, micro-loops may also occur, where traffic at the PLR is forwarded back to an already traversed node. With TI-LFA, micro-loops are only solved after the IGP has converged. As a consequence, and for those reasons, additional measures need to be implemented.

Any additional or alternative approach needs to consider: 1) the definition of where SR segment ID stack should preferably be updated, e.g., at the PLR or at network ingress nodes; 2) which portion of the recovery path should be updated, e.g., to the destination or only to the next segment not affected by the failure. These questions define three end-to-end possibilities: PLR to the destination (TI-LFA), PLR to the next segment, or edge-to-edge SR recovery paths. It might also be conceivable to implement corrections on traffic load balancing to improve traffic distribution, or even alter the SID stack of SR paths not affected by the failure. In this context, we devised distinct approaches to analyze single link failure impacts on SR networks resources utilization and, simultaneously, evaluate their responses.

A. Edge-to-Edge Shortest Path (E2E SP)

The most straightforward response to a single link failure is to reroute traffic using IGP shortest paths. In practice, all SR paths that included the failing link (u, v) are reconfigured with a single Node Segment [node(t)], where t is the provider's edge destination. This approach has the disadvantage of not being responsive enough. As edge routers need to become aware of the fault, it can only be implemented after the fault is announced to all routers, and the IGP has converged. On the positive side, it does not require any centralized control, and only edge nodes need to recompute SR paths.

B. Single Objective SALP-SR (SO SALP)

A SALP-SR configuration encompasses IGP link weights configuration, edge-to-edge SR path definitions, and load balancing splitting ratios between parallel paths. All configurations are derived from a set of integers, the IGP link weights, and from a set of real values assigned to each node (node-p values). When the network topology changes due to a link failure, the change is announced to all network nodes and to the controller or PCE. To reflect the changes, the IGP recomputes the shortest-paths, while the controller redefines the SR paths and, most importantly, the traffic load balancing between parallel paths. This is equivalent to applying the path computation process described in Section IV-E to the altered network topology while preserving the already installed IGP weights.

A disadvantage of this procedure is that, before it can be applied, the fault needs to propagate through the network. Furthermore, it might imply to change SR paths already assigned to some flows in order to comply with the new configuration. On the positive side, this procedure does not require any preemptive computation. SR paths and load balancing ratios are computed in a few milliseconds, and the new configuration is installed at the edge nodes by the controller.

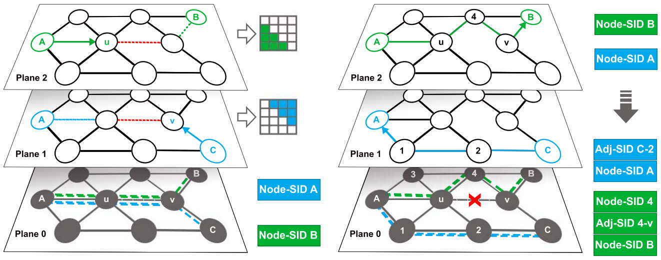
The initially optimized IGP link weights, SR paths, and load balancing ratios between parallel paths depend on the installed node-p values. Consequently, after a link failure the node-p values and IGP weights cease to be coupled optimally, which impacts the quality of SR parallel paths load balancing ratios. Therefore, additionally to the SALP-SR paths recomputation, new and improved traffic splitting ratios between parallel SR paths may be installed by optimizing the node-p values, and thus improving network operational conditions.

The two possible approaches, without and with node-p values optimization, lead to the following two cases: 1) SALP-SR rerouting with default node-p values (*SO SALP*) and 2) SALP-SR rerouting with optimized node-p values (*SO SALP+P*).

C. Multi-Objective SALP-SR (MO SALP)

In this approach, the initial network optimization is performed considering simultaneously: First Objective - minimize the network congestion Φ^* on a fully functional state; and Second Objective - minimize the maximum congestion after a single link failure, that is, every single link is set to fail, and the worst measured congestion is minimized. The formulation of this second objective is $Min(Max(\Phi^*_{(n-1,a)}))$, where ($n$ − 1, a) denotes the failure of each individual link a.

As defined for the SALP-SR approach to a single link failure, the Multi-Objective SALP-SR approach contemplates



(a) Traffic necessities identification (b) Recovery paths computation

Fig. 7. Multiplane recovery path optimization.

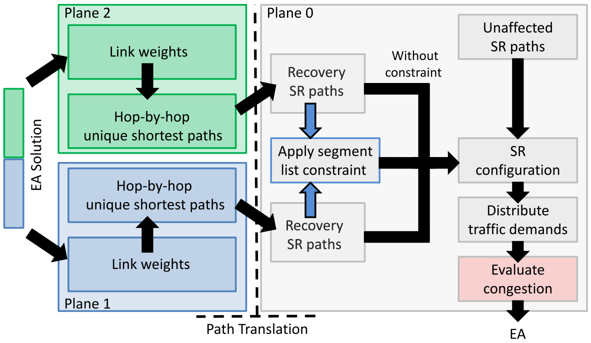


Fig. 8. Evaluation of link failure recovery solutions without or with constrained segment list depth (MP and CMP).

two cases: 1) SALP-SR rerouting with default node-p values (*MO SALP*) and 2) SALP-SR rerouting with optimized node-p values (*MO SALP+P*) In both cases, after a link failure and considering the already existing IGP link weights configuration, the SR paths are recomputed.

D. Multiplane Optimization (MP)

One of the attributes of SALP-SR is that traffic always moves towards the destination when considering distances as the sum of shortest-path link weights. Although this characteristic is a positive SALP-SR property, it nonetheless narrows the number of possible recovery path solutions. This approach forsakes this restriction, i.e., recovery paths may include segments that locally drive traffic away from the destination. This goal is achieved using additional network planes during the optimization process, where each additional plane is used to obtain SR recovery paths for the traffic affected by the failure. A conceptual representation of this approach is presented in Fig. 7.

The computation of recovery paths for the failure of each link (u, v) is divided into two main steps:

1 - Identification of traffic that needs to be rerouted: The optimization procedure identities traffic, which, before the failure, travels over (u, v). From this analysis, the algorithm produces two traffic demand matrices, D_1 and D_2, one for each of the failing link entry ports, representing traffic necessities that need to be rerouted after the link failure, Fig. 7(a).

2 - Recovery paths computation: The optimization of recovery paths is performed using a single objective EA whose objective is to minimize the network congestion measure Φ^*. While the link weights configuration 0 remains unchanged, the additional planes, planes 1 and 2, have a set of independent link weights, as illustrated in Fig. 7(b). During the optimization process, the link weights of plane 1 and 2 are concatenated into a single solution vector which evolves, enabling to obtain hop-by-hop unique shortest-paths that are used to steer the affected traffic affected. At each iteration, hop-by-hop paths are translated into SR recovery paths that reflect the existing IGP configuration (plane 0), Fig. 8. The obtained SR recovery paths are then added to the unaffected SR paths in plane 0, configuring a new solution which is evaluated by the EA.

This optimization is an extremely time-consuming process. Therefore it must be performed preemptively for each possible single link failure. SR recovery path configurations, resulting from the optimization process, are stored in a database to be deployed if and when necessary. Planes 1 and 2 only compute unique recovery paths between each source/destination pair to ensure a unique translation mapping of hop-by-hop paths to SR recovery paths in plane 0. If more than an SR path for the same source/destination pair is affected by the failure, the affected traffic is aggregated into the same recovery path, and the traffic load balancing between parallel paths is adjusted accordingly. Traffic splitting ratios assigned to unaffected parallel paths are kept unaltered.

E. Constrained Multiplane Optimization (CMP)

The formerly presented multiplane optimization approach has a drawback: SR paths may require more than 3 SIDs to be configured. To overcome such a result and impose a maximum length of 3 SIDs in the segment header list, SR paths in the final solution that exceed that restriction can be replaced with shortest path SR configurations, i.e., the node-SID of the destination.

The multiplane approach can also be extended so that the segment list depth constraint is included in the evaluation of all solutions during the optimization process, as depicted in Fig. 8. After the translation of hop-by-hop paths to SR paths, SR path configurations with segment list depth greater than 3 are converted to SR paths with a single SID, the destination nodal SID, forcing traffic to be forwarded along shortest paths from source to destination.

VII. Simulation Results

A. Experiments Setup

The validation of the SALP optimization model was carried out for a single and multi TMs, the last simulating changing traffic conditions, and in the context of a link failure. In all cases, the aim is to find an SR configuration (IGP Link weights, SR paths, and parallel paths load balancing ratios) that optimizes the distribution of traffic on the available resource by minimizing the network congestion metric Φ^*.

The experiments consider a set of distinct synthetic network topologies generated with the Brite topology generator [42] using a Barabasi-Albert model. The topologies vary in size and average node in/out-degree with links' capacity uniformly distributed in the interval [1; 10] Gbits. Random traffic

TABLE II
COMPARISON OF SALP-SR VARIANTS WITH CG4SR, DEFO AND 2SR OPTIMIZED CONFIGURATIONS

Topology			SALP		SALP + LP		SALP Hybrid		CG4SR		DEFO		2SR		MCF Φ^*		MCF MLU	
Name	Nodes	Links	Φ^*	MLU	Φ^*	MLU	Φ^*	MLU	Φ^*	MLU	Φ^*	MLU	Φ^*	MLU	Φ^*	MLU	Φ^*	MLU
$Rand30_2$	30	110	1.38	0.68	1.45	0.63	1.57	0.51	1.41	0.70	1.84	0.57	2.14	0.49	1.30	0.67	1.80	0.49
$Rand30_4$	30	220	1.59	0.73	1.61	0.65	1.63	0.47	1.64	0.88	2.02	0.68	2.31	0.62	1.22	0.67	1.75	0.47
$Rand50_2$	50	194	1.61	0.93	1.66	0.92	1.54	0.92	1.60	0.91	2.03	0.92	2.30	0.91	1.39	0.94	2.79	0.91
$Rand50_4$	50	380	2.49	0.96	2.27	0.83	1.83	0.60	2.21	0.90	2.45	0.79	2.18	0.92	1.21	0.67	2.12	0.55
$rf3967$	79	294	1.59	0.90	1.65	0.72	1.63	0.70	2.03	0.88	2.00	1.23	40.60	0.73	1.38	0.87	2.06	0.68
$rf1221$	104	302	1.45	0.87	1.56	0.85	1.55	0.86	2.12	0.88	1.78	0.86	55.55	1.30	1.40	0.87	2.26	0.85

matrices *D* were generated considering each topology characteristics. We also included in the first set of experiments two Rocketfuel ISP topologies, the Exodus-USA (AS3967), and the Telstra-Australia (AS1221), with traffic necessities and IGP configurations obtained from REPETITA [43].

B. MOEAs Configuration

The OSPF routing protocol accepts for link weight integer values between 1 and 65535. However, to reduce the size of the optimization search space we only considered link weights taken from the range [1; 20]. The node-p values, used to improve traffic load balancing, take real values in the range [−10; 10], incremented by 0.01, and encoded as integers. We also defined a minimum threshold for parallel traffic load balancing, with a negligible impact on the network congestion measure Φ^*. Any parallel path between a source *s* and a destination *t* forwards at least 5% of the aggregated traffic from *s* to *t*.

The evolution of solutions is assured by an implementation of the Non-dominated Sorting Genetic Algorithm (NSGA-II) [44]. At each generation, the EA creates new solutions using mutation and recombination operators with equal probability of being applied: one-point and two-points crossover (selects, respectively, one and two points in both parents and swaps the genetic information delimited by the point(s)); incremental/decremental mutation (replaces randomly selected genes by the next or previous value within the allowed range); and random mutation (replaces randomly selected genes by a random value within the allowed range). Parents for matting are selected running a tournament selection mechanism, i.e., the solution with the best fitness is selected from a random subset of the population. The individuals that best solve the problem are selected, using a ranking selection over their fitness, to integrate the next generation. In the experiments, the populations are initialized with 100 random individuals and kept with constant size throughout all iterations. All experiments use the same configuration, varying in the number of iterations, which defines the algorithm stopping criterion.

C. Single Traffic Demand Matrix

The first set of experiments aims to evaluate the SALP-SR optimization variants under single traffic demands, that is, without any regard to traffic variations. We compare our proposal against three other approaches for SR-TE optimization: CG4SR [19], 2SR [14], and DEFO [18], whose results were obtained using REPETITA and with IGP's configurations optimized to minimize the congestion metric Φ^*. The maximum number of segments for each approach was CG4SR: 4 segments, DEFO: 3 segments, and 2SR: 2 segments. In respective to SALP, the results are representative of 30 experiments, with a stopping criterion of 1500 iterations. Also included in Table II are optimization results of the Multi-commodity Flow (MCF) relaxation problem respective to both, Φ^* and MLU, minimization objectives.

SAPL's initial configurations offer globally better results than CG4SR, DEFO, and 2SR in regard to its optimization objective (Φ^*). The vice-versa is also true, that is, results for the minimization of the MLU are globally better with CG4SR, 2SR and DEFO than with SALP. In the random topologies, CG4SR results are very similar to those obtained with SALP. The provided IGP configurations, optimized for the Φ^* metric, enabled CG4SR to obtain equivalent results. When using the IGP configurations obtain from REPETITA (*rf*3967 and *rf*1221), CG4SR performs significantly worst than SALP for the same metric.

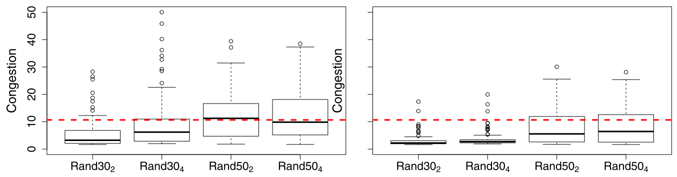
When applying a load balancing correction for the minimization of the MLU over a SALP solution (SALP+LP), that are obtained in just a few seconds, we observe that no algorithm is better than the others. The SALP hybrid optimization, on the other hand, is able to achieve a good compromise between the two objectives, and although each evaluation requires a longer computational time, due to solving the MLU LP, the algorithm converges faster and requires far fewer iterations. The quality of the CG4SR, DEFO, and 2SR solutions depend on pre-optimized IGP configurations. Contrariwise, SALP optimizes both the IGP and SR configurations simultaneously. Results also show that with SALP, most traffic is forwarded resorting to SR paths configured with a single SID, i.e., shortest paths. The percentage of SALP SR paths configured with 1, 2 and 3 segments are, in average and respectively, $Rand30_2$: 79%, 10%, 11%; $Rand30_4$: 67%, 28%, 5%; $Rand50_2$: 82%, 11%, 7%; $Rand50_4$: 65%, 27%, 8% ; *rf*3967: 79%, 6%, 15%; and *rf*1221: 76%, 5%, 19%, showcasing one of SALP main goals.

D. Traffic Variation

The evaluation of SALP-SR under traffic variations was conducted considering, for each network, a set *M* of 100 random TMs. All TMs have an expected average link utilization in the range [0.27, 0.35] when optimizing the MCF for the metric Φ^*. Two traffic demand matrices, D_1 and D_2, are derived from each set *M* using k-means clustering under the scheme described in Section V-A. The initial SR configurations are obtained running a multi-objective EA which simultaneously

TABLE III
AVERAGE CONGESTION VALUES BEFORE AND AFTER TRAFFIC LOAD BALANCING CORRECTION

Topology	$Rand30_2$	$Rand30_4$	$Rand50_2$	$Rand50_4$
MO SALP	5.47	9.73	12.47	12.55
MO SALP+P	3.03	3.62	8.06	8.34



(a) MO SALP

(b) MO SALP+P

Fig. 9. Congestion distributions under traffic variations.

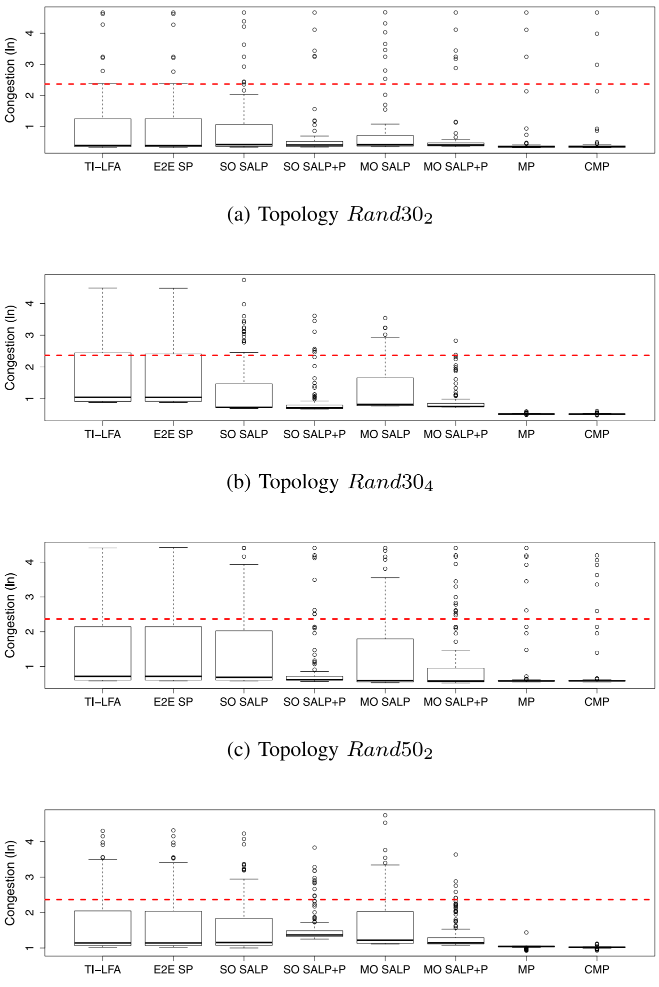
minimizes Φ_1^* and Φ_2^*, the congestion measure Φ^* when considering, respectively, the TMs D_1 and D_2. The solution with the smallest trade-off value $\alpha \times \Phi_1^* + (1-\alpha) \times \Phi_2^*$ is selected from the Pareto optimal set and installed. In the previous selection procedure, α is the real value in range [0;1] that reflects the proportion of elements in each of the two clusters. The network congestion Φ^* is than evaluated for all TMs in *M* (MO SALP), and the congestion with optimized node-p values (MO SALP+P) is also appraised to quantify the improvement in the network performance.

The average congestion values of 10 experiments, for each network, are presented in Table III. Also, Fig. 9 presents the distributions of the Φ^* congestion values obtained for each TM in *M*. The network's operational threshold, a congestion value of 10.67, is identified as a dashed red line.

Results show that the initial configurations were able to ensure acceptable performance for roughly 50% or more of the TMs. In the particular case of the $Rand30_2$ topology, and despite the randomness in traffic variations, this percentage rises to almost 90%. When the new configurations fail to accommodate traffic variations, node-p values are optimized, providing better network resource utilization. Although insufficient in some cases, this approach always introduces improvements in congestion. In the particular case of the $Rand50_2$ topology, the number of TMs that the network is able to accommodate increases by 30%, while in the $Rand50_4$ topology, the number of such TMs increases by 20%.

E. Single Link Failure

As in the previous experiments, traffic demand matrices were randomly generated with an expected utilization of 30% of the network resources. The initial single and multi-objective configurations were obtained using a stopping criterion of 1000 iterations. Node-p values optimization have a stopping criterion of 100 iterations, and multiplane optimization were run with a stopping criterion of 150 iterations for each individual link failure. Results presented in Table IV are average post-convergence congestion values per link from 10 runs of each experiment. Also included are the confidence intervals (95%) for the congestion of each link when optimization is required. Results are divided into two main groups, before (normal state) and after a single link failure. In the last, organized by each approach previously defined, values are the mean of network congestion after the failure of each link, one at a time.



(a) Topology $Rand30_2$

(b) Topology $Rand30_4$

(c) Topology $Rand50_2$

(d) Topology $Rand50_4$

Fig. 10. Individual link failure congestion values distribution.

The minimum node in/out-degree of a network topology significantly influences the quality of recovery paths. Topologies with higher minimum node degree have more available edge-to-edge recovery paths after a single link failure. In this context, it is understandable that topologies with a higher minimum node in/out-degree present globally better results. Next, we discuss each approach's results.

1) Edge-to-Edge Shortest Path Approach (E2E SP): Results for the first group of link failure experiments reveal that there are no significant differences in the congestion values between recovery methods that solely rely on post-convergence shortest paths, that is, from the PLR to the destination (TI-LFA) and edge-to-edge shortest-path. They both globally display values below (but near) the operational threshold of the network (10.67). In particular, TI-LFA simulations, although capable of shortening connectivity lost to under 50 msec, present some of the highest congestion values on all experiments.

2) SALP-SR Approach (SO SALP): Approaches that use SALP-SR enhance shortest-path recovery approaches by taking advantage of non-shortest path links, and as expected,

TABLE IV
AVERAGE CONGESTION BEFORE AND AFTER SINGLE LINK FAILURE

Topology			Normal State		Single Link Failure State							
Name	Nodes	Links	SO	MO	TI-LFA	E2E SP	SO SALP	SO SALP+P	MO SALP	MO SALP+P	MP	CMP
$Rand30_2$	30	110	1.40	1.42	8.63	8.62	8.49	6.09 (± 4.900)	8.19	6.26 (± 4.720)	5.01 (± 4.416)	4.78 (± 4.764)
$Rand30_4$	30	220	1.96	2.35	10.68	10.80	6.75	3.46 (± 0.964)	5.07	2.84 (± 1.048)	1.68 (± 0.006)	1.68 (± 0.005)
$Rand50_2$	50	194	1.79	2.07	8.73	8.75	8.31	5.63 (± 2.697)	8.36	6.39 (± 2.692)	4.86 (± 2.669)	4.46 (± 2.175)
$Rand50_4$	50	380	2.43	4.91	7.75	7.78	6.76	5.31 (± 0.674)	7.50	4.18 (± 0.475)	2.82 (± 0.018)	2.77 (± 0.009)

TABLE V
HEADER STACK DEPTH OF MULTIPLANE OPTIMIZATIONS

Topology	$Rand30_2$	$Rand30_4$	$Rand50_2$	$Rand50_4$
Paths with more than 3 SIDs	4.7%	10.8%	14.6%	24.8%
Maximum header stack length	5	6	7	7
Congestion when replacing long SR paths with SR SP	6.21	2.19	6.28	3.84

enabling network congestion to diminish. In the first approach, SO SALP, forwarding paths are reconfigured by the SALP-SR algorithm considering the newly computed IGP shortest-paths. It is important to emphasize that none of the approaches alter IGP weights, and SR paths not affected by the link failure remain unchanged. If additionally, load balancing ratios are adjusted, SO SALP+P column in Table IV, network congestion values drop significantly when compared to those obtained by shortest path rerouting. In fact, after correcting load balancing ratios, only a small percentage of link failures result in a congested network state, that is, above the 10.67 threshold in Fig. 10.

3) Multi-Objective SALP-SR (MO SALP): Initial configurations obtained with the multi-objective SALP optimization (MO SALP) are more robust to single link failures than the single objective configurations (SO SALP), as can be observed in Table IV and Fig. 10. However, the differences in congestion values fade with the adjustment of load balancing ratios (SO SALP+P vs. MO SALP+P). A multi-objective optimization establishes a compromise between the optimization goals, a trade-off, by relaxing configuration fitness on both objectives. However, an increase in SR configurations flexibility, with a penalization on fully functional network congestion, is insufficient to improve on all results obtained with single-objective optimization. Additional measures need to be installed to improve on the already obtained results.

4) Multiplane Approach (MP): The multiplane optimization provides a broader set of recovery path alternatives which are not bounded to shortest path or SALP routing models. Consequently, the multiplane approach permits to significantly reduce the post-convergence congestion values of the previous approaches and attain results close to those observed before the link failure. However, SR paths provided by this approach may not comply with the constraint of a maximum header stack depth of 3 segments. The percentage of SR paths that use more than 3 SIDs and the maximum number of SIDs for each network topology, are presented in Table V.

It comes with no surprise that as topologies grow in the number of nodes and links, the percentage of multiplane recovery SR paths that require more than 3 SIDs and the maximum SID header stack length escalates rapidly. The excessive label stacking length may cause scalability issues as the maximum SID header stack length varies currently from 3 to 5 depending on the equipment manufacturer [45].

To overcome such restriction and impose a maximum length of 3 SIDs, SR paths that exceed that restriction may be substituted with shortest path SR configurations. Congestion values obtained by performing this alteration are also presented in Table V. In all cases the congestion increases to values equivalent to those obtained with SAPL load balancing corrections, SO SALP+P and MO SALP+P in Table IV.

5) Constrained Multiplane Approach (CMP): The results for the constrained multiplane optimization with a maximum segment list depth of 3 are also included in Table IV. At first glance, it might be surprising that the constraint multiplane congestion values are slightly better than those provided by the unconstrained version. The multiplane approach uses unique shortest paths to achieve an entire mapping of hop-by-hop paths to SR paths in distinct IGP configuration spaces. The constrained approach, on the other hand, replaces some unique shortest paths by equal shortest paths of the working IGP space at each solution evaluation. This difference allows the EA to obtain solutions that make better utilization of network resources and provide slightly better congestion values. The drawback of this approach is the time required to obtain recovery shortest-paths that need to be pre-computed and stored to be deployed when needed. Also, it only allows protecting the network against single link failures, which nonetheless are the most common type of topology failure.

Single link failures have two main impacts on the network's operations. They undermine both connectivity and overall network congestion. Segment Routing enables the deployment of more complete and effective responses to the problem of preserving the network's post-failure congestion levels. We derive two main conclusions from the explored approaches: 1) TI-LFA is an excellent solution to quickly reestablish networks connectivity, but is insufficient to provide operational levels of congestion after a link failure; 2) other approaches such as SALP-SR with node-p optimization, or the constrained multiplane approach, deliver better post-convergence congestion levels, but require more time to be deployed. A combination of both approaches presents itself as a good compromise to achieve both goals, shortening the reaction time, and decreasing network congestion. After a link failure, connectivity can be reestablished using TI-LFA, and as soon as the IGP converges, optimized SR paths can be installed at edge nodes, achieving this way both desired goals.

SALP uses multiprocessing and all optimizations were run inside a High-Performance Computing cluster. As such, the time required by each optimization depends greatly on the available computational resources. Node-p values optimization run under a minute with 4 core CPUs, MLU optimization requires only a few seconds. Initial SALP optimizations, however, are very time consuming, but on the other hand, they are rarely required as the correction on load balancing should be enough to maintain the network on operational conditions. This is comparable to IGP's weights optimization when running MPLS/SR.

VIII. Conclusion

By decomposing forwarding paths into segments, identified by labels or SIDs, SR improves Traffic Engineering and enables new solutions for the optimization of network resource utilization. This work proposed an optimization technique, the Single Adjacency Label Path (SALP-SR), to improve resource utilization in SR networks. SALP-SR possesses several advantages: it provides near-optimal resource utilization while using at most three segments to configure forwarding paths; most traffic is routed using a single SID; after each hop traffic is always closer to its destination; SALP-SR enables to improve load balancing between parallel paths by optimizing the MLU or network resource utilization. This last feature enables to address congestion problems that may result from changes in traffic demands or link failures while preserving SR paths and IGP link weights. We also compared distinct approaches for link failures recovery. We identified some limitations of TI-LFA and proposed alternatives that may be coupled with TI-LFA to provide solutions that are both responsive and congestion aware. Future work will also envisage protection against node failures and additional constraints such as Network Function Virtualization points where traffic needs to be delivered.

The Network Optimization framework, NetOpt, that implements all proposed methods is publicly available at http://www.bio.di.uminho.pt/netopt, allowing to reproduce all experiments.

References

[1] C. Filsfils, N. K. Nainar, C. Pignataro, J. C. Cardona, and P. Francois, "The segment routing architecture," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[2] N. Feamster, J. Rexford, and E. Zegura, "The road to SDN: An intellectual history of programmable networks," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 2, pp. 87–98, Apr. 2014.

[3] A. Farrel, O. Komolafe, and S. Yasukawa, "An analysis of scaling issues in MPLS-TE core networks," Internet Eng. Task Force, Fremont, CA, USA, RFC 5439, Feb. 2009. [Online]. Available: https://rfc-editor.org/rfc/rfc5439.txt

[4] J. Moy, "OSPF version 2," Internet Soc., Reston, VA, USA, RFC 2328, Apr. 1998. [Online]. Available: https://rfc-editor.org/rfc/rfc2328.txt

[5] H. Gredler and W. Goralski, *The Complete IS-IS Routing Protocol*. London, U.K.: Springer, 2005.

[6] C. Hopps, "Analysis of an equal-cost multi-path algorithm," Internet Eng. Task Force, Fremont, CA, USA, RFC 2992 (Informational), 2000.

[7] Y. Rekhter, S. Hares, and T. Li, "A border gateway protocol 4 (BGP-4)," Internet Soc., Reston, VA, USA, RFC 4271, Jan. 2006. [Online]. Available: https://rfc-editor.org/rfc/rfc4271.txt

[8] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in *Proc. IEEE INFOCOM Conf. Comput. Commun. 19th Annu. Joint Conf. IEEE Comput. Commun. Soc.*, 2000, pp. 519–528.

[9] J. F. Gonçalves and M. G. C. Resende, "Biased random-key genetic algorithms for combinatorial optimization," *J. Heuristics*, vol. 17, no. 5, pp. 487–525, Oct. 2011. [Online]. Available: http://dx.doi.org/10.1007/s10732-010-9143-1

[10] M. Rocha, P. Sousa, P. Cortez, and M. Rio, "Quality of service constrained routing optimization using evolutionary computation," *Appl. Soft Comput.*, vol. 11, no. 1, pp. 356–364, Jan. 2011.

[11] R. Mijumbi, J. Serrat, J.-L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 236–262, 1st Quart., 2016.

[12] C. Filsfils, D. Dukes, S. Previdi, J. Leddy, S. Matsushima, and D. Voyer, "IPv6 segment routing header (SRH)," Internet Eng. Task Force, Fremont, CA, USA, RFC 8754, Mar. 2020. [Online]. Available: https://rfc-editor.org/rfc/rfc8754.txt

[13] E. Moreno, A. Beghelli, and F. Cugini, "Traffic engineering in segment routing networks," *Comput. Netw.*, vol. 114, pp. 23–31, Feb. 2017.

[14] R. Bhatia, F. Hao, M. Kodialam, and T. V. Lakshman, "Optimized network traffic engineering using segment routing," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 657–665.

[15] T. Settawatcharawanit, V. Suppakitpaisarn, S. Yamada, and Y. Ji, "Segment routed traffic engineering with bounded stretch in software-defined networks," in *Proc. IEEE 43rd Conf. Local Comput. Netw. (LCN)*, Oct. 2018, pp. 477–480.

[16] S. Gay, R. Hartert, and S. Vissicchio, "Expect the unexpected: Sub-second optimization for segment routing," in *Proc. Conf. Comput. Commun. (INFOCOM)*, 2017, pp. 1–9.

[17] H. Roomi and S. Khorsandi, "Semi-oblivious segment routing with bounded traffic fluctuations," in *Proc. Iran. Conf. Elect. Eng. (ICEE)*, 2018, pp. 1670–1675.

[18] R. Hartert *et al.*, "A declarative and expressive approach to control forwarding paths in carrier-grade networks," in *Proc. ACM Conf. Special Interest Group Data Commun.*, New York, NY, USA, 2015, pp. 15–28. [Online]. Available: https://doi.org/10.1145/2785956.2787495

[19] M. Jadin, F. Aubry, P. Schaus, and O. Bonaventure, "CG4SR: Near optimal traffic engineering for segment routing with column generation," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, Apr. 2019, pp. 1333–1341.

[20] Z. N. Abdullah, I. Ahmad, and I. Hussain, "Segment routing in software defined networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 464–486, 1st Quart., 2019.

[21] P. L. Ventre *et al.*, "Segment routing: A comprehensive survey of research activities, standardization efforts and implementation results," 2019. [Online]. Available: arXiv:1904.03471.

[22] A. Bashandy *et al.*, "Topology independent fast reroute using segment routing," Internet Eng. Task Force, Fremont, CA, USA, Internet-Draft draft-ietf-rtgwg-segment-routing-ti-lfa-05, Oct. 2018. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-rtgwg-segment-routing-ti-lfa-05

[23] K.-T. Foerster, M. Parham, M. Chiesa, and S. Schmid, "TI-MFA: Keep calm and reroute segments fast," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2018, pp. 415–420.

[24] F. Aubry, S. Vissicchio, O. Bonaventure, and Y. Deville, "Robustly disjoint paths with segment routing," in *Proc. 14th Int. Conf. Emerg. Netw. Exp. Technol.*, New York, NY, USA, 2018, pp. 204–216.

[25] F. Hao, M. Kodialam, and T. V. Lakshman, "Optimizing restoration with segment routing," in *Proc. IEEE INFOCOM 35th Annu. IEEE Int. Conf. Comput. Commun.*, Apr. 2016, pp. 1–9.

[26] K.-T. Foerster, Y.-A. Pignolet, S. Schmid, and G. Trédan, "CASA: Congestion and stretch aware static fast rerouting," in *Proc. IEEE INFOCOM IEEE Conf. Comput. Commun.*, Paris, France, Apr./May 2019, pp. 469–477. [Online]. Available: https://doi.org/10.1109/INFOCOM.2019.8737438

[27] D. Xu, M. Chiang, and J. Rexford, "Deft: Distributed exponentially-weighted flow splitting," in *Proc. IEEE INFOCOM 26th IEEE Int. Conf. Comput. Commun.*, May 2007, pp. 71–79.

[28] A. Altin, B. Fortz, M. Thorup, and H. Umit, "Intra-domain traffic engineering with shortest path routing protocols," *Ann. Oper. Res.*, vol. 204, no. 1, pp. 65–95, 2013.

[29] C. C. Coello, G. B. Lamont, and D. A. V. Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems* (Genetic and Evolutionary Computation). Heidelberg, Germany: Springer-Verlag, 2006.

[30] V. Pereira, P. Sousa, P. Cortez, M. Rio, and M. Rocha, "Comparison of single and multi-objective evolutionary algorithms for robust link-state routing," in *Evolutionary Multi-Criterion Optimization* (Lecture Notes in Computer Science), vol. 9019. Cham, Switzerland: Springer, 2015, pp. 573–587.

[31] M. Ericsson, M. G. C. Resende, and P. M. Pardalos, "A genetic algorithm for the weight setting problem in OSPF routing," *J. Comb. Optim.*, vol. 6, no. 3, pp. 299–333, Sep. 2002. [Online]. Available: https://doi.org/10.1023/A:1014852026591

[32] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot, "Traffic matrix estimation: Existing techniques and new directions," *SIGCOMM Comput. Commun. Rev.*, vol. 32, no. 4, pp. 161–174, Aug. 2002.

[33] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale ip traffic matrices from link loads," in *Proc. ACM SIGMETRICS Int. Conf. Meas. Model. Comput. Syst.*, New York, NY, USA, 2003, pp. 206–217.

[34] A. Nucci, R. Cruz, N. Taft, and C. Diot, "Design of IGP link weight changes for estimation of traffic matrices," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004, pp. 2341–2351.

[35] K. Papagiannaki, N. Taft, and A. Lakhina, "A distributed approach to measure IP traffic matrices," in *Proc. 4th ACM SIGCOMM Conf. Internet Meas.*, New York, NY, USA, 2004, pp. 161–174.

[36] Q. Zhao, Z. Ge, J. Wang, and J. Xu, "Robust traffic matrix estimation with imperfect information: Making use of multiple data sources," *SIGMETRICS Perform. Eval. Rev.*, vol. 34, no. 1, pp. 133–144, Jun. 2006.

[37] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1355734.1355746

[38] A. Tootoonchian, M. Ghobadi, and Y. Ganjali, "OpenTM: Traffic matrix estimator for openflow networks," in *Proc. 11th Int. Conf. Passive Active Meas.*, 2010, pp. 201–210.

[39] C. Yu, C. Lumezanu, Y. Zhang, V. Singh, G. Jiang, and H. V. Madhyastha, "FlowSense: Monitoring network utilization with zero measurement cost," in *Proc. 14th Int. Conf. Passive Active Meas.*, 2013, pp. 31–41.

[40] D. Sanvito, I. Filippini, A. Capone, S. Paris, and J. Leguay, "Adaptive robust traffic engineering in software defined networks," in *Proc. IFIP Netw. Conf. Workshops (IFIP Netw.)*, May 2018, pp. 145–153.

[41] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002.

[42] A. Medina, A. Lakhina, I. Matta, and J. Byers, "BRITE: Universal topology generation from a user's perspective," Comput. Sci. Dept., Boston Univ., Boston, MA, USA, Rep. BUCS-TR-2001-003, 2001.

[43] S. Gay, P. Schaus, and S. Vissicchio, "REPETITA: Repeatable experiments for performance evaluation of traffic-engineering algorithms," 2017. [Online]. Available: arXiv:1710.08665.

[44] K. Deb, *Multi-Objective Optimization Using Evolutionary Algorithms* (Wiley Interscience Series in Systems and Optimization). Chichester, U.K.: Wiley, 2001.

[45] R. Guedrez, O. Dugeon, S. Lahoud, and G. Texier, "Label encoding algorithm for MPLS segment routing," in *Proc. 15th Int. Symp. Netw. Comput. Appl. (NCA)*, Oct. 2016, pp. 113–117.

Vítor Pereira received the first Licentiate degree in mathematics, the second Licentiate degree in computer sciences, the M.Sc. degree in networks and communication services, and the Ph.D. degree in informatics from the University of Minho, Portugal, in 1998, 2008, 2012, and 2019, respectively, where he is currently a Researcher with the Centre of Biological Engineering.

Miguel Rocha is currently an Associate Professor with the Informatics Department, University of Minho, Portugal, the Director of the Master in Bioinformatics, and a Senior Researcher within the Centre of Biological Engineering, where he co-leads a research team in Bioinformatics and Systems Biology. He has authored around 200 publications in international journals and conferences. He has been the PI and has collaborated in several funded research projects by the Portuguese FCT, European Commission and private companies.

Pedro Sousa received the Licentiate degree in systems and informatics engineering, and the M.Sc. and Ph.D. degrees in computer science from the University of Minho, Portugal, in 1995, 1997, and 2005, respectively. In 1996, he joined the Computer Communications Group, Department of Informatics, University of Minho, where he is an Assistant Professor and performs research activities within Centro Algoritmi, mainly focusing on computer networks related topics.