Received: 22 January 2019 | Accepted: 24 January 2019 | Published on: 12 March 2019

DOI: 10.1002/net.21877

SPECIAL ISSUE ARTICLE

WILEY

An optimization approach for congestion control in network routing with quality of service requirements

Pasquale Avella[1] | Giacomo Bernardi[2] | Maurizio Boccia[3] | Sara Mattia[4,†]

[1]Dipartimento di Ingegneria, Università del Sannio, Benevento, Italy
[2]Dipartimento di Engineering, EOLO S.p.A, Busto Arsizio, Italy
[3]Dipartimento di Energia Elettrica e Tecnologie dell'Informazione, Università di Napoli Federico II, Napoli, Italy
[4]Istituto di Analisi dei Sistemi ed Informatica, Consiglio Nazionale delle Ricerche, Roma, Italy

Correspondence
Istituto di Analisi dei Sistemi ed Informatica, Consiglio Nazionale delle Ricerche, via dei Taurini 19, 00185 Roma, Italy.
Email: sara.mattia@iasi.cnr.it

Abstract

In this paper we study a network design problem arising in the management of a carrier network. The aim is to route a traffic matrix, minimizing a measure of the network congestion while guaranteeing a prescribed quality of service. We formulate the problem, devise presolve procedures to reduce the size of the corresponding mixed-integer programming formulation and show that the proposed approach can efficiently solve some real-life problems, leading to an improvement with respect to the current practice in a real case study.

KEYWORDS

congestion, mixed-integer model, preprocessing, routing

1 | INTRODUCTION

This study concerns the use of optimization techniques to route a set of traffic demands (commodities) in a real-life widespread wireless network, ensuring that given quality of service (QoS) indicators are met.

The optimization approach is motivated by two major trends in the broadband industry: (*a*) video streaming generates the vast majority of data exchanged and (*b*) data consumption grows faster at peak time than the day average. As a result, Internet Service Providers must over-provision their networks to avoid congestion at peak time, which would result in abrupt decrease in the quality of experience perceived by their customers. The investment efficiency is thus constrained by building capacity which is only used a few hours per day.

The research activity has been conducted in cooperation with EOLO S.p.A. [16], an Italian carrier operating a large fixed wireless access network, with about 260 000 customers in Northern and Central Italy. Its network infrastructure consists of about 2000 radio towers, connected by about 5000 radio links. The structure of the network is depicted in Figure 1, where each node is a tower, each edge is a radio link corresponding to two directed arcs and the red nodes are the main data centers, being the endpoint of most of the commodities. The network is used to accommodate about 8000 traffic demands. Each demand is defined by a source node, a destination node, and a required bandwidth. Demands are classified into three groups according to the degree of sensitivity of traffic to routing delay: *real-time* (very delay-sensitive applications, such as bidirectional voice/video), *standard* (default) and *bulk* (no delay-sensitive traffic, such as file sharing). The protocol is IP (Internet Protocol) over MPLS (MultiProtocol Label Switching) which allows to explicitly select end-to-end routes for the commodities. Each demand must be routed on a single path (unsplittable flows), guaranteeing that it crosses a number of arcs not exceeding the maximum prescribed by the service level required by the corresponding class. Two main performance indicators affect the routing policy: the congestion of the arcs and the length of the routes. The congestion of an arc is the amount of traffic traversing the arc with respect to its reference bandwidth. High levels of congestions on any arc of a path lead to delays and to packet loss. Furthermore,

[†] The author is a member of INdAM
Corrections added on 14-March 2019; after first publication.

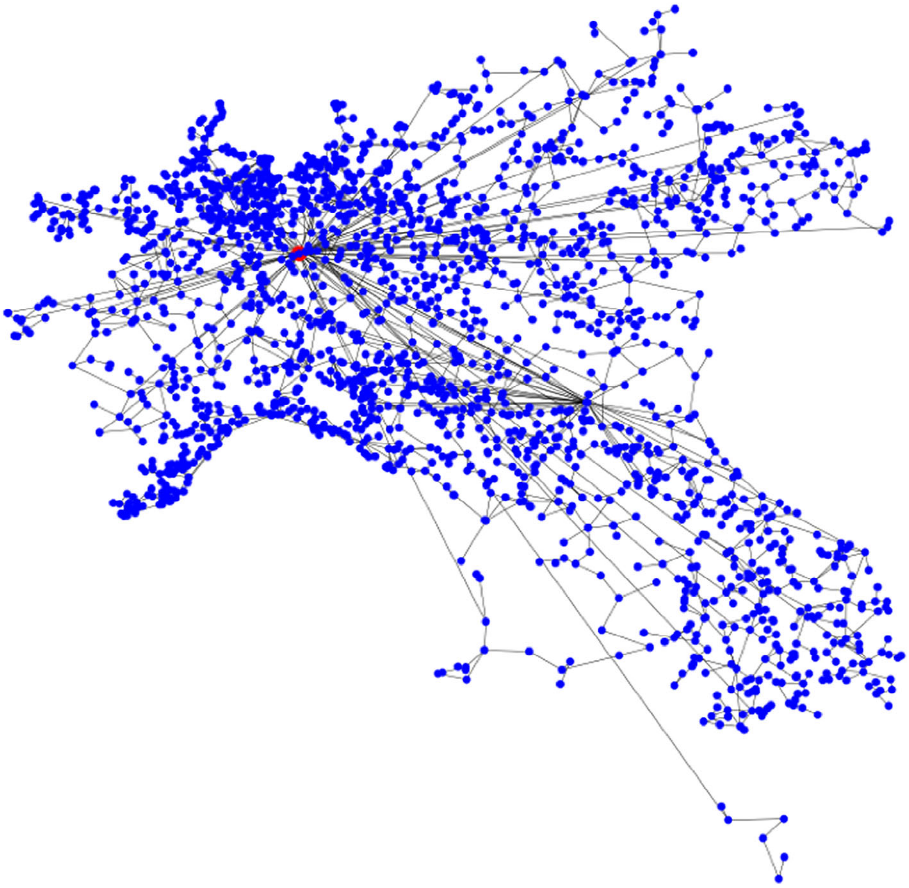

FIGURE 1 The EOLO network as of October 2017 [Color figure can be viewed at wileyonlinelibrary.com]

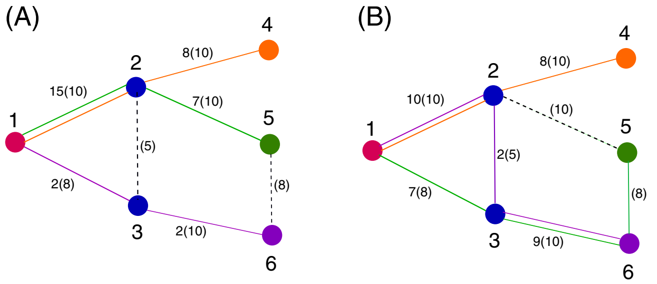

FIGURE 2 SRP and rerouting [Color figure can be viewed at wileyonlinelibrary.com]

saving capacity on the arcs may also help ensuring the survivability of the network in case of failures, as the spare capacity can be used to accommodate backup paths. The length of a route is the number of arcs on the route. Long routes increase the bandwidth occupancy as well as the probability of delays and packet loss. Also, long routes lead to a less reliable service, being more exposed to failures of the arcs.

The routes are periodically recomputed at fixed intervals (e.g., 15 minutes), as both link availability and demands may vary. In addition, routes are recomputed after every change in the network topology (an existing link becoming unavailable or a new link becoming available). The current routing policy adopted in EOLO's network essentially consists of routing each commodity along the path between the origin and the destination having the smallest number of arcs. We call this policy the *shortest route policy* (SRP). The main advantage of SRP is that it is easy to implement, as the routing of a commodity is independent of the others. On the other hand, it might lead to highly unbalanced loads, with few very congested arcs and many others underutilized. In many real-world scenarios, rerouting a small subset of paths would lead to a significant reduction of the overall congestion. Figure 2 shows how rerouting a demand can lead to a more balanced traffic load than the one produced by SRP, on a simple network with six towers and seven links. Two labels are associated with each link, namely the traffic traversing the arc and (in brackets) the reference bandwidth. Denoting by d_{ij} the amount to be routed between source node (or origin) i and destination node j, the demands to be routed on the network are: $d_{12} = 8$; $d_{15} = 7_{,}$ and $d_{16} = 2$. In Figure 2A), each demand is routed on the shortest path between the origin and the destination (SPR policy). With the SPR policy the bandwidth on links (1,2) significantly exceeds its reference bandwidth (or bandwidth capacity). Moreover two links ((5,6) and (2,3)) are unused. Rerouting d_{15} and d_{16} on longer paths ({(1,3),(3,6),(6,5)} for the commodity d_{15} and {(1,2),(2,3),(3,6)} for d_{16}, respectively) all the arcs of the network are not congested.

The main scope of the study is to use optimization in order to get a more balanced traffic load. By segregating the bulk of latency-insensitive traffic and strategically allowing it to be detoured on slightly longer arcs than the shortest path, our optimization approach enables EOLO to perform per-application multipath routing, spreading network demands in real-time across the whole networks, and improving the overall investment efficiency. We refer to the routing policy developed in this paper as the *load balancing policy* (LBP). In the proposed mixed-integer programming (MIP) model, the routing constraints (QoS requirements, unsplittable routing and routing of all the demands) are considered hard-constraints, whereas obtaining low congestion values on the arcs is a soft constraint. That is, we model the routing requirements as constraints of the model, while we increasingly penalize high levels of congestion in the objective function. These kinds of approaches often result in models with a large number of constraints and variables, making it difficult to solve them to optimality. To reduce the problem size, so that a good solution can be obtained in the required time limit, we develop dedicated preprocessing techniques based on both the network topology [5, 15, 40] and the estimation of an upper bound on the amount of traffic that can traverse an arc.

In the rest of the paper, after reviewing the literature in Section 2 we: (1) discuss how to model this real-life problem by a mixed-integer programming formulation with a piecewise-linear objective function (Section 3); (2) present preprocessing techniques to make the problem tractable by a commercial solvers on standard workstations (Section 4); (3) show results on a real case study provided by EOLO, illustrating that the proposed LBP leads to quite significant improvements in the load balancing with respect to SRP (Section 5). Conclusions are given in Section 6.

2 | LITERATURE REVIEW

Optimization models of problems arising from telecommunication networks are quite popular in the literature. Many aspects of telecommunications networks design, planning and management have been considered. In [6, 8, 10, 26, 38], among the others, the problem of choosing minimum cost integer capacities for the arcs of a network to ensure the routing of a set of commodities has been studied and different solution approaches and formulations have been proposed. In [1, 9, 14, 22, 32, 33], instead, arc capacities have already been set and the decision is to choose the arcs to be activated to ensure the routing of some traffic demands. Two-layer networks are investigated in [17, 27, 28] and references therein. For papers dealing with the survivability of a network under some failure scenarios see [1, 11, 23, 27, 35], where several protection and restoration techniques are analyzed. Models for problems with uncertain demands are considered in [4, 25, 29–31, 34]. For a more complete overview and additional references on optimization models for telecommunications problems we address the reader to [39].

For objective functions commonly used in telecommunications network problems, [20] point out that the most common optimization criterion adopted to minimize the congestion is the Kleinrock delay function (e.g., [24])

$$\sum_{a \in A} \frac{x}{c - x}$$

where x the traffic on an edge $a \in A$ and c is its capacity. The Kleinrock function penalizes congestion with increasing costs. It is convex, so it could be minimized by convex programming algorithms, but it is often approximated by a convex piecewise linear function to benefit of the effective mixed-integer programming solvers nowadays available [7, 18–21, 36].

Problems where the length of the paths is limited, known as hop-constrained routing problems, are also studied in the literature. Those problems are hard so solve, as they include as a special case the resource constrained shortest path problem, which is known to be *NP*-hard. In [37] a heuristic method is proposed: it consists of solving the linear relaxation and then using the obtained solution as a starting point for a local search. In [12, 13] a Benders-like formulation is investigated. The problem consists of selecting a minimum cost set of arcs so that the corresponding induced subgraph contains at least a given number arc-disjoint paths of at most a given length between a set of source-destination pairs. The commodities are not classified according to the QoS and the arcs are uncapacitated, that is, once an arc has been selected, it can accommodate any amount of traffic and no congestion-driven objective is considered. For additional details on traffic engineering we refer to [3, 41] and references therein.

As far as we know, there is no paper in the literature presenting an exact approach for a problem with a step-increasing objective function, QoS restrictions and unsplittable routing, as the one we consider here. We observe that even problems with a piecewise-linear and unsplittable flows only (without QoS constraints) are challenging from a computational viewpoint and suffer from poor lower bounds (e.g., [20]).

3 | PROBLEM FORMULATION

The network is described by a directed graph $G(V, A)$ with node set V, representing the transmission towers to be interconnected, and arc set A, representing the virtual links between the towers.

A reference capacity C_{ij} (bandwidth) is associated with each arc $(i,j) \in A$. Unlike standard network design problems, this value is not the maximum amount of capacity available on the arc and it can be exceeded. Its main purpose is to be used as reference value to compute the level of congestion of the arcs. High level of congestions are penalized in the objective function.

A traffic matrix has to be routed on G.

The traffic between two nodes is represented by a commodity k, flowing from a source node $s_k \in V$ to a sink node $t_k \in V$. Let $d_k = d_{s_k t_k}$ be the required bandwidth of commodity k. Each commodity has to be routed through a non-bifurcated (unsplittable) path consisting at most of L_k arcs. Let K denote the set of commodities to be routed on G. Let f_{ij}^k be a binary variable which takes value one if commodity $k \in K$ traverses arc $(i,j) \in A$ and zero otherwise. Let x_{ij} be the total flow on arc $(i,j) \in A$. The goal is to route the commodities without exceeding path lengths L_k for $k \in K$ and minimizing a measure of the network congestion.

Let $\phi(x_{ij})$ be the cost function providing a measure for the congestion of arc (i, j). The problem can be formulated as follows.

$$\min \sum_{(i,j)\in A} \phi(x_{ij}) \tag{1}$$

$$\sum_{(i,j)\in\delta_k^+(i)} f_{ij}^k - \sum_{(i,j)\in\delta_k^-(i)} f_{ji}^k = \begin{cases} 1 & i = s_k \\ -1 & i = t_k \\ 0 & \text{otherwise} \end{cases} \qquad \begin{array}{l} k \in K, \\ i \in V \end{array} \tag{2}$$

$$\sum_{(i,j)\in A} f_{ij}^k \le L_k \qquad k \in K \tag{3}$$

$$\sum_{k\in K} d_k f_{ij}^k = x_{ij} \qquad (i,j) \in A \tag{4}$$

$$\mathbf{x} \in \mathbb{R}_+^{|A|}, \mathbf{f} \in \{0,1\}^{|K|\times|A|}$$

Let $\delta_k^-(i)$ and $\delta_k^+(i)$ be the sets of the incoming and outgoing arcs of node $i \in V$, respectively. Constraints (2), together with the integrality requirements on the flow variables $\mathbf{f}$, impose that, for each commodity $k \in K$, nodes s_k and t_k are connected by a single path. Constraints (3) impose that, for each commodity $k \in K$, the length of the path (the number of crossed arcs) does not exceed L_k. Constraints (4) define the variable x_{ij} as the sum of the flows traversing the arc (i, j).

Since in our problem the capacity C_{ij} associated with each arc does not define an upper bound for the total flow traversing the arc (i, j), we could not use the Kleinrock function as it is. In addition, the need of tackling large-scale instances by taking advantage of the powerful functionalities of mixed-integer linear solvers, motivated modeling the cost function $\phi(x_{ij})$ as a convex piecewise linear function, with objective increasing with arc congestion (see also [20]). Let $Q = \{\alpha_1, \alpha_2, \dots, \alpha_n\}$ be a set of congestion levels, with $\alpha^p \ge \alpha^{p-1} \ge 0$, measuring the percentage of bandwidth occupancy on the arcs. We associate to each range $I^p = [\alpha^p, \alpha^{p+1}]$, $p = 1, \dots, n-1$ a cost h^p. We denote by I the set of the considered I^p. Assuming that $h^p < < h^{p+1}$, in any optimal solution an interval is used if and only if the previous one has been already saturated. It follows that we can disaggregate x_{ij} into x_{ij}^p, $p \in I$ and the formulation becomes:

$$\min \sum_{(i,j)\in A} \sum_{p\in I} \frac{h^p}{C_{ij}} x_{ij}^p \sum_{(i,j)\in\delta_k^+(i)} f_{ij}^k -$$

$$\sum_{(i,j)\in\delta_k^-(i)} f_{ji}^k = \begin{cases} 1 & i = s_k \\ -1 & i = t_k \\ 0 & \text{otherwise} \end{cases} \qquad \begin{array}{l} k \in K, \\ i \in V \end{array}$$

$$\sum_{(i,j)\in A} f_{ij}^k \le L_k \qquad k \in K$$

$$\sum_{k\in K} d_k f_{ij}^k = \sum_{p\in I} x_{ij}^p \qquad (i,j) \in A$$

$$x_{ij}^p \le C_{ij}\alpha^{p+1} \qquad p \in I, \quad (i,j) \in A$$

$$\mathbf{x} \in \mathbb{R}_+^{|A|\times|I|}, \mathbf{f} \in \{0,1\}^{|K|\times|A|} \tag{5}$$

Objective function (5) is a piecewise-linear function where the larger the congestion of an arc, the larger the cost. The first model already included many variables, the second one even more. In order to reduce the size of the model, some preprocessing procedures are described in the next section.

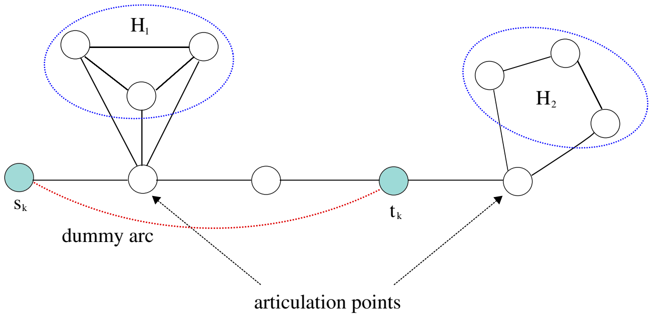

FIGURE 3 Articulation points [Color figure can be viewed at wileyonlinelibrary.com]

4 | PRESOLVE OPERATIONS

We adopt three logical presolve operations. The first two (L_k and *articulation point* presolve) are based on the network topology and, given a commodity $k \in K$, aim at identifying nodes and arcs that cannot be used to route k, either because the length may exceed L_k (L_k presolve) or because the corresponding path is not simple (articulation point presolve). In this way, we can associate to each commodity $k \in K$ a subgraph of G, denoted by $G^k(V^k, A^k)$, including the nodes and the arcs which can be used by commodity k. After that, we can remove to the model all the variables f_{ij}^k for $(i,j) \notin A^k$, thus reducing the number of the **f** variables, and we can write flow conservation constraints (2) only for the nodes $i \in V^k$, thus reducing the number of constraints. Instead, the third presolve operation (*bounds tightening*) is based on the estimation of an upper bound on the amount of traffic that can traverse an arc. The scope is to eliminate **x** variables representing values of congestions that cannot be reached, thus reducing the number of the **x** variables and of the corresponding constraints.

4.1 | L_k presolve

Given a commodity $k \in K$, the L_k presolve excludes arcs of A that cannot be used to route commodity k because there is no path of length less than or equal to L_k which uses them. This presolve has originally been used for the Resource Constrained Shortest Path problem [5, 15]. For any two nodes $u,\ v \in V$, let $\sigma(u,v)$ denote the length of the shortest path from u to v. Given a commodity k and a node $i \in V$, i belongs to V^k if and only if $\sigma(s_k, i) + \sigma(i, t_k) \leq L_k$. If this condition is not satisfied, that is, if $\sigma(s_k, i) + \sigma(i, t_k) \geq L_k + 1$, then $i \not\in V^k$ and all the arcs in $\delta^+(i) \cup \delta^-(i)$ do not belong to A^k. Given a commodity k and an arc $(i,j) \in A$, $(i,\ j)$ belongs to A^k if and only if $\sigma(s_k, i) + \sigma(j, t_k) \leq L_k - 1$. If this condition is not satisfied, that is, if $\sigma(s_k, i) + \sigma(j, t_k) \geq L_k$, then $(i,j) \not\in A^k$. For algorithms on all-pairs shortest paths, we refer the reader to [2]. We denote by $G_1^k(V_1^k, A_1^k)$ the graph for commodity k after the L_k presolve.

4.2 | Articulation point presolve

Here we assume to ignore the orientation of the arcs and, hence, we refer to them as edges. The scope is to remove from G_1^k node and edges not belonging to simple (i.e., acyclic) paths from s_k to t_k. Given $G_1^k(V_1^k, A_1^k)$, we define node $i \in V_1^k$ as an articulation point of $G_1^k(V_1^k, A_1^k)$ if its removal disconnects G_1^k into two or more components. After the removal of i, Let $H \subset V_1^k$ define a connected component of G_1^k not containing s_k and t_k. Any $(s_k,\ t_k)$-path traversing a node in H forms a cycle since it visits node i at least twice. It follows that the nodes in H cannot belong to V^k and the arcs in $\cup_{i \in H} \delta^+(i) \cup \delta^-(i)$ do not belong to A^k. Let $E_1^k = A_1^k \cup \{(s_k, t_k)\}$, the articulation points can be efficiently detected by running the Tarjan's algorithm [40] on $G_1^k(V_1^k, E_1^k)$. Figure 3 reports an example of subgraph G_1^k with two articulation points, where H_1 and H_2 are the two sets of nodes that can be removed by the subgraph. We denote by $G_2^k(V_2^k, A_2^k)$ the graph for commodity $k \in K$ after the L_k and the articulation point presolve.

4.3 | Bounds tightening

We can remove for the graph nodes and arcs not belonging to any G_2^k for $k \in K$. We denote the corresponding graph by $G_2(V_2, A_2)$. For the edges in A_2, a trivial upper bound u_{ij} on the required bandwidth can be computed as below.

$$u_{ij} = \sum_{k \in K:(i,j) \in A_2^k} d_k \quad (i,j) \in A_2$$

Recalling that the congestion levels have the property that $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$, we have that $u_{ij} < \alpha_q C_{ij}$ implies that all the congestion levels α_h for $h \geq q$ are redundant and the variables $x_q, x_{q+1}, \ldots, x_n$ and the related constraints can be deleted from the formulation.

TABLE 1 Traffic demand patterns of the benchmark instances

	#real	dtot	#stan	dtot	#bulk	dtot
D	3814	8453	1907	18 404	1907	22 320
A	3814	7356	1907	12 545	1907	40 940

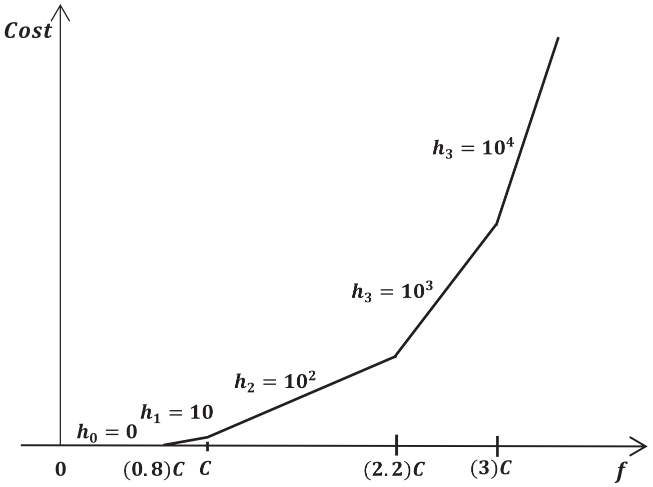

FIGURE 4 The piecewise linear objective function

5 | COMPUTATIONAL EXPERIMENTS

The approach consists in modeling the problem as described in Section 3, applying the presolve operations described in Section 4 and then solving the resulting reduced formulation by a commercial solver. The algorithm has been implemented in C and the computation has been carried out on an i7-2620 processor @ 2.7 GHz with 8GB of RAM using FICO Xpress 7.9 as solver. The solution process is stopped as soon as the integrality gap falls below 1%. In the rest of the section we present the instances, show the reduction provided by the presolve and then discuss the final results.

5.1 | The instances

The approach has been tested on two real-life instances, called D (day) and A (afternoon), provided by EOLO. The instances are defined on the same network topology—the current EOLO network, consisting of 1908 nodes and 4426 arcs—and differ because of the traffic pattern, as summarized in Table 1, where *dtot* is the total demand for the respective commodity type. We cannot provide the full traffic matrices, but in the table we report, for each instance, the number of real-time (#real), standard (#stan), and bulk (#bulk) commodities, along with the total demand for each priority class.

Piecewise linear objective function has for each edge four breakpoints, corresponding to the congestion levels:

$$\alpha^1 = 0.8C_{ij}$$

$$\alpha^2 = C_{ij}$$

$$\alpha^3 = 2.2C_{ij}$$

$$\alpha^4 = 3.0C_{ij}$$

The unit costs associated with each line segment are (see Figure 4)

- $h_0 = 0$ if $x_{ij} \in [0, \alpha^1]$
- $h_1 = 10$ if $x_{ij} \in (\alpha^1, \alpha^2]$
- $h_2 = 100$ if $x_{ij} \in (\alpha^2, \alpha^3]$
- $h_3 = 1000$ if $x_{ij} \in (\alpha^3, \alpha^4]$
- $h_4 = 10000$ if $x_{ij} \in (\alpha^4, +\infty)$

A different L_k value is assigned to each commodity k, computed as the sum of the length of the shortest path between s_k and t_k and of a coefficient associated with the priority class.

TABLE 2 Presolve reduction

	No presolve		L_k		$L_k + AP$		$L_k + AP + BT$	
	# rows	# cols	# rows	# cols	# rows	# cols	# rows	# cols
D	8 008 182	17 977 666	211 247	335 447	111 535	172 235	111 535	156 714
A	8 008 182	17 960 096	211 196	353 035	120 740	172 233	120 740	157 482

TABLE 3 Presolve computation times

	$\mathrm{L_k}$	AP	BT	Total
Inst D	9.45	6.14	6.19	21.78
Inst A	9.42	5.67	5.70	20.79

TABLE 4 MIP solution

	LP	XLP	BLB	BUB	% gap	# nodes	Time
D	530.08	553.56	553.56	559.01	0.97	1	34.91
A	1375.99	1485.62	1492.74	1507.05	0.95	10	28.87

5.2 | Presolve reduction

In Table 2 we show the cumulative effect of presolve on the size of the formulation. Columns L_k, $L_k + AP$ and $L_k + AP + BT$ show the number of rows and columns of the formulation in case of no presolve, for L_k, for L_k and articulation point and for L_k, articulation point and bounds tightening presolve. Each row of the table corresponds to an instance. The three presolve operations achieve a drastic reduction, as they delete about 99% of the columns and 98% of the rows. Table 3 reports on the computation time spent in each of the presolve operations. For both the instances presolve operations totally required about 20 seconds. Most of the presolve time was spent to compute all-pairs shortest paths in the L_k presolve. It is worth noting that, as long as the network remains the same, L_k and articulation point presolve do not need to be repeated when the traffic matrix changes, thus potentially leading to a significant further reductions in computing times.

5.3 | Results

The results are reported in Table 4. Columns *LP*, *XLP*, *BLB*, *BUB* show the value of the LP relaxation, the value of the LP relaxation after Xpress cuts, the best lower bound and the best upper bound at the end of the algorithm, respectively. We also report the final gap (% gap), the number of nodes (# nodes) and the branch-and-bound time (wall clock seconds), which does not include the presolve time reported in Table 3. Both the instances are solved in less than 40 seconds, that is, less than 60 seconds including presolve times.

We now discuss here the effect of applying the routing policy produced by the optimization algorithm (LBP) and compare it with the policy currently used by the company (SRP). Figure 5 shows, for each policy, the number of arcs for which the amount of traffic exceeds 80% of the reference value (*almost congested* arcs). Figure 6 reports the number of congested arcs,

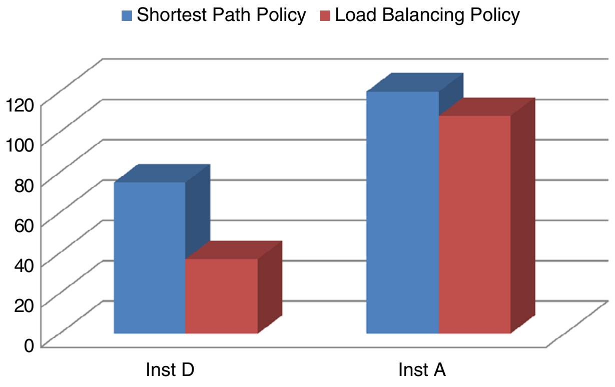

FIGURE 5 Number of arcs near to the congestion [Color figure can be viewed at wileyonlinelibrary.com]

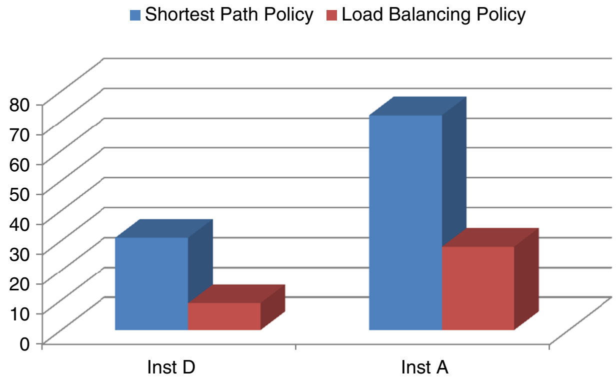

FIGURE 6 Number of congested arcs [Color figure can be viewed at wileyonlinelibrary.com]

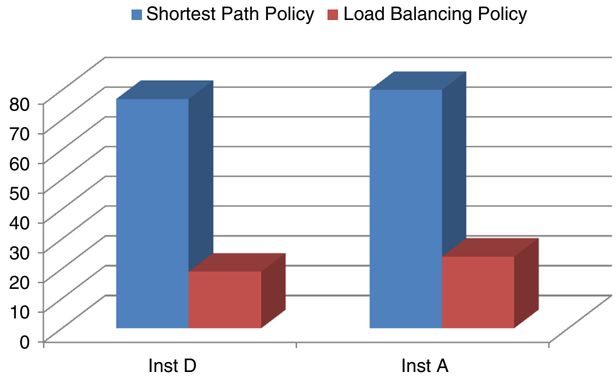

FIGURE 7 Average overcongestion of the arcs [Color figure can be viewed at wileyonlinelibrary.com]

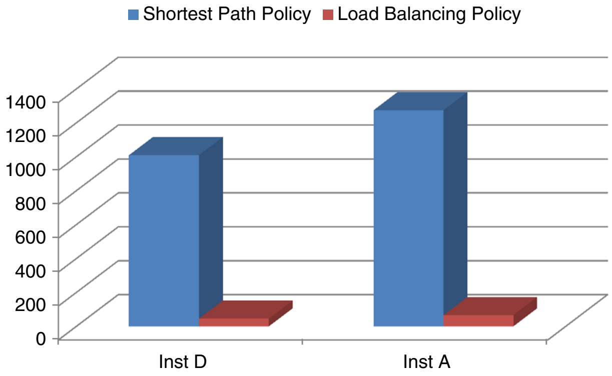

FIGURE 8 Maximum value of overcongestion [Color figure can be viewed at wileyonlinelibrary.com]

that is, the arcs for which the amount of traffic exceeds its reference bandwidth. Figure 7 illustrates the average overload on the saturated arcs, while Figure 8 shows the maximum value of the overload in the LBP and SRP solutions. As clearly depicted in the figures, LBP dramatically reduces the number of congested arcs: they are about 40% (instance D) and 70% (instance A) with the SPR, whereas only 10% and 20% of the arcs are congested using LBP. The number of almost congested arcs is also decreased: the effect is more evident on the D instance, where LBP is able to obtain half of the almost congested arc of SRP. The average congestion is reduced from almost 80% to almost 20% and the maximum congestion is largely smaller for LBP than for SRP.

6 | CONCLUSIONS

We studied a network design problem arising in the management of a carrier network. The study was conducted in cooperation with EOLO. The main scope was to devise an optimization approach able to produce a routing policy with a lower level of arc congestion with respect to the policy currently implemented by the company. We modeled the problem using a piecewise-linear objective function to penalize the arc congestion. We applied to the resulting model some presolve procedures, which turned out to be very effective to reduce the size of the formulation. This allowed us to efficiently solve real-life problems using a commercial solver with default settings. The resulting routing policy presents levels of average and maximum congestion far below the ones of the current policy on two real-life instances provided by the company.

ACKNOWLEDGMENTS

We thank the network team at EOLO, in particular (in alphabetical order): F. Alberti, P. Biasioli, S. Ceccato, M. Citterio and A. Milani.

ORCID

Sara Mattia https://orcid.org/0000-0001-5054-731X

REFERENCES

[1] B. Addis, G. Carello, and S. Mattia, *Survivable green traffic engineering with shared protection*, Networks **69** (2017), 6–22.
[2] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms, and Applications*, Prentice-Hall, Upper Saddle River, NJ, 1993.
[3] I. Akyildiz, A. Lee, P. Wang, M. Luo, and W. Chou, *A roadmap for traffic engineering in SDN-openflow networks*, Comput. Netw. **71** (2014), 1–30.
[4] A. Altin, H. Yaman, and M. Pinar, *The robust network loading problem under hose demand uncertainty: Formulation, polyhedral analysis, and computations*, INFORMS J. Comput. **23**(1) (2010), 75–89.
[5] Y. Aneja and K. Nair, *The constrained shortest path problem*, Naval Res. Logistic Q. **25**(3) (1978), 549–555.
[6] P. Avella, S. Mattia, and A. Sassano, *Metric inequalities and the network loading problem*, Discrete Optim. **4** (2007), 103–114.
[7] S. Balon, F. Skivée, and G. Leduc, *How well do traffic engineering objective functions meet the requirements?* International Conference on Research in Networking, Springer, 2006, pp. 75–86.
[8] F. Barahona, *Network design using cut inequalities*, SIAM J. Optim. **6** (1996), 823–834.
[9] A. Benhamichea, R. Mahjoub, N. Perrot, and E. Uchoa, *Unsplittable non-additive capacitated network design using set functions polyhedra*, Comput. Oper. Res. **66** (2016), 105–115.
[10] D. Bienstock, S. Chopra, O. Günlük, and C.-Y. Tsai, *Minimum cost capacity installation for multicommodity network flows*, Math. Program. B **81** (1998), 177–199.
[11] S. Borne, E. Gourdin, B. Liau, and A. Mahjoub, *Design of survivable IP-over-optical networks*, Ann. Oper. Res. **146** (2006), 41–73.
[12] Q. Botton, B. Fortz, L. Gouveia, and M. Poss, *Benders decomposition for the hop-constrained survivable network design problem*, INFORMS J. Comput. **25**(1) (2013), 13–26.
[13] Q. Botton, B. Fortz, and L. Gouveia, *On the hop-constrained survivable network design problem with reliable edges*, Comput. Oper. Res. **64** (2015), 159–167.
[14] T. Crainic, A. Frangioni, and B. Gendron, *Bundle-based relaxation methods for multicommodity capacitated fixed charge network design*, Discrete Appl. Math. **112** (2001), 73–99.
[15] I. Dumitrescu and N. Boland, *Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem*, Networks **42**(3) (2003), 135–153.
[16] EOLO, http://www.eolo.it (Accessed February 18, 2019).
[17] B. Fortz and M. Poss, *An improved Benders decomposition applied to a multi-layer network design problem*, Oper. Res. Lett. **37**(5) (2009), 777–795.
[18] B. Fortz, and M. Thorup, *Internet traffic engineering by optimizing OSPF weights*. Proceedings of INFOCOM 2000, IEEE, 2000, pp. 519–528.
[19] B. Fortz and M. Thorup, *Increasing internet capacity using local search*, Comput. Optim. Appl. **29**(1) (2004), 13–48.
[20] B. Fortz, L. Gouveia, and M. Joyce-Moniz, *Models for the piecewise linear unsplittable multicommodity flow problems*, Eur. J. Oper. Res. **261**(1) (2017), 30–42.
[21] E. Gourdin, and O. Klopfenstein, *Comparison of different QOS-oriented objectives for multicommodity flow routing optimization*. Proceedings of the International Conference on Telecommunications (ICT 2006), 2006.
[22] K. Holmberg and D. Yuan, *A lagrangian heuristic based branch-and-bound approach for the capacitated network design problem*, Oper. Res. **48** (2000), 461–481.
[23] J. Kennington and M. Lewis, *The path restoration version of the spare capacity allocation problem with modularity restrictions: Models, algorithms, and an empirical analysis*, INFORMS J. Comput. **13**(3) (2001), 181–190.
[24] L. Kleinrock, *Communication Nets: Stochastic Message Flow and Delay*, McGraw-Hill, New York, 1964.
[25] C. Lee, K. Lee, and S. Park, *Benders decomposition approach for the robust network design problem with flow bifurcations*, Networks **62**(1) (2013), 1–16.
[26] S. Mattia, *Separating tight metric inequalities by bilevel programming*, Oper. Res. Lett. **40**(6) (2012a), 568–572.
[27] S. Mattia, *Solving survivable two-layer network design problems by metric inequalities*, Comput. Optim. Appl. **51**(2) (2012b), 809–834.
[28] S. Mattia, *A polyhedral study of the capacity formulation of the multilayer network design problem*, Networks **62**(1) (2013a), 17–26.
[29] S. Mattia, *The robust network loading problem with dynamic routing*, Comput. Optim. Appl. **54**(3) (2013b), 619–643.
[30] S. Mattia, *The cut property under demand uncertainty*, Networks **66**(2) (2015), 159–168.
[31] S. Mattia, *A polyhedral study of the robust capacitated edge activation problem*. Proceedings of ODS 2017, volume 217 of PROMS, 2017, pp. 413–419.
[32] S. Mattia, *The capacity formulation of the capacitated edge activation problem*, Networks **71**(4) (2018), 381–402.
[33] S. Mattia, *MIP-based heuristic approaches for the capacitated edge activation problem: The effect of non-compactness*, Soft. Comp. https://doi.org/10.1007/s00500-018-3443-z.
[34] S. Mattia and M. Poss, *A comparison of different routing schemes for the robust network loading problem: Polyhedral results and computation*, Comput. Optim. Appl. **69**(3) (2018), 753–800.

[35] S. Orlowski and M. Pióro, *Complexity of column generation in network design with path-based survivability mechanisms*, Networks **59**(1) (2012), 132–147.

[36] M. Pióro and D. Medhi, *Routing, Flow, and Capacity Design in Communication and Computer Networks*, Elsevier, Amsterdam, The Netherlands, 2004.

[37] H. Pirkul and S. Soni, *New formulations and solution procedures for the hop constrained network design problem*, Eur. J. Oper. Res. **148**(1) (2003), 126–140.

[38] C. Raack, A. Koster, S. Orlowski, and R. Wessäly, *On cut-based inequalities for capacitated network design polyhedra*, Networks **57**(2) (2011), 141–156.

[39] M. Resende and P. Pardalos, *Handbook of Optimization in Telecommunications*, Springer Science & Business Media, New York, 2008.

[40] R. Tarjan, *Depth-first search and linear graph algorithms*, SIAM J. Comput. **1**(2) (1972), 146–160.

[41] N. Wang, K.H. Ho, G. Pavlou, and M. Howath, *An overwiew of routing optimization for internet traffic engineering*, IEEE Commun. Surv. Tutor. **10**(1) (2008), 3–20.

How to cite this article: Avella P, Bernardi G, Boccia M, Mattia S. An optimization approach for congestion control in network routing with quality of service requirements. *Networks.* 2019;74:124–133. https://doi.org/10.1002/net.21877