ARTICLE TEMPLATE

An exact approach for finding bicriteria maximally SRLG-disjoint/shortest path pairs in telecommunication networks

José Craveirinha[a], Marta Pascoal[a,b] and João Clímaco[a]

[a]INESC Coimbra, University of Coimbra, Portugal [b]University of Coimbra, CMUC, DMUC, Portugal; Politecnico di Milano, Milano, Italy

ARTICLE HISTORY
Compiled June 16, 2023

ABSTRACT
The paper addresses a bicriteria optimisation problem in telecommunication networks that aims at finding Pareto efficient pairs of paths between two given nodes, seeking to minimise the number of SRLGs (Shared Risk Link Groups) common to both paths and the path pair cost. This problem is of particular importance in telecommunication routing design, namely concerning resilient routing models where both a primary and a backup paths have to be calculated to minimise the risk of failure of a connection between origin and terminal nodes, in case of failure in the primary path. An exact resolution method is applied for solving this problem, enabling the calculation of the whole set of Pareto optimal solutions, which combines a transformation of the network representation with a path ranking algorithm. A comprehensive experimental study on the application of this approach, using reference network topologies, considering random SRLG assignments to the links and random link bandwidth occupations, together with the discussion on typical examples of solution selection and potential advantages of the method, are presented.

KEYWORDS
Telecommunication routing design; Shared Risk Link Groups; Resilient routing models; Bicriteria optimisation

1. Introduction

Multicriteria shortest path problems have important applications in telecommunication networks, specially in routing design – see (Raith and Ehrgott 2009; Clímaco and Pascoal 2012; Clímaco and Craveirinha 2019). For a better understanding of the present work we review some background concepts concerning the underlying problem of telecommunication network routing design addressed by our approach.

In the design of routing models for telecom networks it is assumed, in general, a two-layer network representation: a physical layer (the physical network) that is a representation of the relevant physical and functional elements involved in the communication process (such as cables, optical fibers, switches, routers, control software, or others, depending on the technological features and architecture of the considered network) and the logical layer (the logical network) which is a mathematical representation of the possible interconnections between functional service access points, corresponding to origin-destinations nodes. The routing procedures involve the calculation of one or

CONTACT Marta Pascoal. Email: marta.brazpascoal@polimi.it

more paths between origin-destination nodes in the logical network (the representation level where routing algorithms work). Our bicriteria approach addresses a telecom network routing design problem involving the calculation of two paths: an active path and a protection path (usually with reserved bandwidth for each end-to-end connection) to be used when the active path becomes inoperative due to unavailability of any of its arcs as a result from a failure in some underlying element(s) of the physical network. In this context, (for modelling purposes) it is used the concept of Shared Risk Ling Group (SRLG), such that each arc in the logical network is associated with a list of SRLGs, each of them corresponding to a risk in the physical network, represented by a specific label. Each label may be identified mathematically by an element in a discrete finite set. Each SRLG is defined by the set of logical arcs which may be affected by a specific risk (defined at the physical network level) and, in the applications, its specification is obtained from the mapping of the physical network onto the logical network. Therefore, a one-to-one correspondence between labels-risks-SRLGs can be established. Note that, in many situations, an arc is associated with more than one SRLG since logical arcs tend to share physical network elements thence the associated risks of failure. For example, a cut in an optical fibre or a fault in a wavelength division multiplexer usually affects several arcs of the logical network. Further analysis of this and other types of problems of telecommunication network routing design from an Operations Research perspective and focusing on the application of multicriteria approaches can be seen in (Clímaco and Craveirinha 2019). This type of problems is of paramount importance having in mind that very high levels of service availability should be maintained in the event of failures and the enormous amounts of traffic that can be lost. Usually the network designer seeks to calculate a pair of SRLG-disjoint paths, ensuring that no single fault of the active path affects the backup, ou protection, path. This problem was shown to be NP-complete in (Hu 2003).

There may arise situations for which no SRLG-disjoint path pair exists, a case in which the aim of the routing procedure may consist of finding a maximally SRLG-disjoint path pair, that is, a path pair with the minimal number of common SRLGs, so as to minimise the risk of simultaneous failure of the two paths. Moreover, a key concern is bandwidth usage optimisation, seeking to optimize the use of bandwidth resources throughout the network links, in order to achieve the maximal possible network traffic carrying capability. This is usually represented in terms of a linear objective function, such that the cost of using a link is dependent on its used bandwidth and the cost of a path is the sum of the costs of its arcs. These considerations lead to a typical formulation of the routing problem with path protection involving the calculation of a pair of paths which are maximally SRLG-disjoint and minimal in terms of the total cost.

A multiobjective lexicographic formulation of this type of problem considering four objective functions was addressed in (Gomes et al. 2016), where two effective heuristics were presented. Other algorithms (either exact algorithms or heuristics) for various lexicographic formulations of similar problems were proposed in (Hu 2003; Todimala and Ramamurthy 2004; Rostami et al. 2007; Gomes and Craveirinha 2010; Silva et al. 2011; Gomes et al. 2013a,b; de Sousa et al. 2019; Pascoal et al. 2022).

The major contributions of this work are: i. the proposal of a bicriteria optimisation approach for the telecom network routing design problem with protection, hereafter designated as NRDPP; ii. the implementation of an adequate exact resolution method for the formulated bicriteria problem, based on the algorithm in (Pascoal and Clímaco 2020), enabling to obtain a set of exact Pareto optimal solutions; iii. the development of an application study using reference telecom network topologies; iv. the description of a procedure for selecting a solution in the obtained Pareto optimal (or efficient) solutions set, suitable for the application in telecom network routing design. These contributions can be specified as follows. Firstly, we formalize the NRDPP in terms of a bicriteria optimisation problem with a linear objective function representing the additive cost of using the two paths and a non-linear objective function representing the number of risks (thence SRLGs) common to the two paths. Note that the non-linear objective function is particularly difficult of handling since it is the cardinal of the intersection of two discrete sets associated with the two paths. Secondly, by identifying each SRLG with a specific label, we describe a resolution method based on the algorithm in (Pascoal and Clímaco 2020), enabling the obtainment of the set of exact Pareto optimal solutions. This base algorithm was customized to take into account relevant features of the application and improve computational performance. We also include an outline review of the auxiliary path ranking algorithm in (Martins et al. 1999). Thirdly, we put forward an extensive experimental study on the application of our bicriteria approach to the NRDPP by using reference test logical network topologies in (Orlowski et al. 2010) and considering random costs (obtained from typical bandwidth availability distributions) and random risk/SRLG assignments to the arcs. This enabled an evaluation of the computational performance, potentialities and possible advantages of the application of our approach. This application study shows that in a large percentage of instances the algorithm obtains, as a subproduct, the exact lexicographic optimum (in term of number of common risks) so that this type of solution may also be considered as a possible final solution, that is, the pair of paths to be chosen for each origin-destination for given input on the network structure, including SRLGs and bandwidths. Note that this exact lexicographic optimal solution (that is, the pair of paths which are maximally SRLG disjoint and, as a secondary objective, of minimal cost) is a common approach in resilient routing design. Furthermore, the resolution method enables the calculation of a set of exact Pareto optimal solutions close to the lexicographic optimum, which eventually may be useful as possible final solutions to the network routing design problem. Moreover, as a contribution to the practical application of our approach, we also discuss and present adequate selection procedures for the final solution to be used in the context of our approach to the NRDPP. In the computational experiments the run times for different number of SRLGs and different mean number of SRLGs per link are analysed, showing the applicability and efficiency of the approach in most types of networks. Potential advantages of this type of approach besides its exactness are discussed, namely regarding the fact that it enables the analysis of trade-offs between risk-disjointness and cost. Furthermore, we show that there are situations (namely when the lexicographic optimum pair of paths has one or more common SRLGs and the risk occurrence probabilities vary significantly), for which it may arise that some of the trade-off solution(s) of this model can be optimally resilient (in probabilistic terms), as hown in the application study. Finally, a typical example of selection of an efficient solution in the Pareto optimal set is shown, highlighting potential advantages of the approach. In Appendix, a review of the auxiliary path ranking algorithm in (Martins et al. 1999), used in the implementation of the resolution algorithm, is shown.

The remainder of the text is organized as follows. In Section 2 the notation and preliminary definitions are introduced. The bicriteria optimisation problem is formalised and the method for solving it is shown by presenting a review of the maximal label disjoint-minimal cost algorithm on which it is based and by referring to auxiliary sub-algorithms. Also, the fundamental underlying mathematical properties of the resolution algorithm are shown and its application to a small instance of the problem is exemplified. The application study of the approach to five sets of logical networks based on reference logical network topologies, considering various distributions of random SRLGs and arc costs, is developed in Section 3, including an analysis of the results and method's performance for the different sets of networks. Moreover, the selection of an efficient solution in the Pareto optimal set, is discussed by recurring to an illustrative example for one of the networks, thence highlighting potential advantages of the approach. Finally, conclusions of this work are outlined in Section 4.

2. The maximally SRLG-disjoint/minimal cost path pair problem

In this section we formalize elementary concepts used in this work, define the addressed problem, describe the algorithm to solve it, and show its basic mathematical properties.

2.1. Preliminary definitions

Let $G = (N, A)$ be a directed network, where N is the set of n nodes and $A \subseteq N \times N$ is the set of m arcs, $n, m \in \mathbb{N}$. Given two nodes $v_1, v_r \in N$, a path from v_1 to v_r in the network G is a sequence $p = \langle v_1, v_2, \ldots, v_r \rangle$, where $(v_i, v_{i+1}) \in A$, for any $i = 1, \ldots, r-1$. Let $s, t \in N$ be called the source and the terminal nodes of G, respectively, and P denote the set of paths in G from s to t without repeated nodes, that is, loopless paths. Note that we consider only loopless paths because all the arc costs are non-negative and the cost function is linear so that the consideration of paths with loops would be unnecessary, as formally shown in Proposition 2.1 in the next subsection.

Let L be the finite discrete set of network labels (such that each label corresponds to exactly one failure risk), ensuring a one-to-one correspondence between labels and SRLGs. Hereafter, we will denote the arc (v_i, v_j) simply by (i, j). Then, the following parameters and sets are associated with any arc $(i, j) \in A$:

- $L_{ij} = \{l_{ij}^1, \ldots, l_{ij}^k\} \subseteq L$, the set of labels corresponding to risks/SRLGs assigned to arc (i, j) that is, which may affect (i, j),
- $A_l = \{(i, j) \in A : l \in L_{ij}\} \subset A$, which defines the SRLG A_l corresponding to label $l \in L$, and
- $c_{ij} \in \mathbb{R}_0^+$, the cost for using the arc (i, j).

The set of arc labels and the linear (additive) cost for any path $p \in P$, are defined by:

$$l(p) = \bigcup_{(i,j) \in p} L_{ij} \tag{1}$$

and

$$c(p) = \sum_{(i,j) \in p} c_{ij}, \tag{2}$$

respectively. Given a pair of paths $(p, q) \in P \times P$, the following objective functions are defined:

- the number of labels/SRLGs common to both paths, given by $z_1(p, q) = |l(p) \cap l(q)|$,

- and the pair's cost, given by $z_2(p,q) = c(p) + c(q)$.

As an example, let us consider the network in Figure 1 with unitary arc costs, $c_{ij} = 1$, and the sets L_{ij} defined by the different arc colours (or the different letters appended to the arc) shown in the image, for any $(i,j) \in A$. Assuming that the initial node is $s = 1$ and the terminal node is $t = 4$, the pair of paths (p,q) with $p = \langle 1,3,4 \rangle$ and $q = \langle 1,3,2,4 \rangle$ has $z_1(p,q) = 1$ shared risk (represented by g, the green colour, given that $l(p) \cap l(q) = \{g\}$), and cost $z_2(p,q) = 5$.

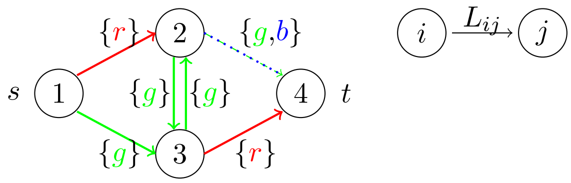


Figure 1. Example network (N,A)

2.2. The problem

Herein we formulate the bicriteria maximally SRLG-disjoint shortest path pair problem (or SRDSP) problem which aims at finding pairs of paths from s to t in (N,A) which minimise the objective functions z_1 and z_2:

$$\begin{array}{ll} \min & z_1(p,q) \\ \min & z_2(p,q) \\ \text{s. t.} & p,q \in P \end{array} \tag{3}$$

When the two objective functions are conflicting, no pair of paths optimizes z_1 and z_2 simultaneously. So, a compromise solution has to be chosen by the network designer in the set of efficient (or Pareto optimal) solutions. A pair of paths $(p',q') \in P \times P$ is said to dominate another pair $(p,q) \in P \times P$ when

$$\begin{cases} (z_1(p',q'), z_2(p',q')) \leq (z_1(p,q), z_2(p,q)) \\ (z_1(p',q'), z_2(p',q')) \neq (z_1(p,q), z_2(p,q)) \end{cases} \tag{4}$$

where the first inequality should be read component-wise. The pair of paths $(p,q) \in P \times P$ is said to be efficient (or Pareto optimal) if and only if there is no other pair $(p',q') \in P \times P$ that dominates it. If (p,q) is an efficient solution, then its objective function vector $(z_1(p,q), z_2(p,q))$ is said to be a non-dominated point.

The set of efficient solutions is denoted by X_E, whereas the set of non-dominated vectors is denoted by Y_N and is also known as the non-dominated frontier or the Pareto frontier. The exact resolution of the SRDSP problem involves the calculation of the set of efficient pairs of paths in P and the corresponding Y_N. If $z(R)$ denotes the set of images of the pairs if paths in $R \subseteq P \times P$, that is, $z(R) = \{(z_1(p,q), z_2(p,q)) : (p,q) \in R\}$, then $Y_N = z(X_E)$. As shown in Proposition 2.1, this set Y_N may be obtained by considering only loopless paths from s to t. As an illustrative example of sets X_E and Y_N the list of pairs of paths linking $s = 1$ to $t = 4$ in the network in Figure 1 is shown in Table 1, where pairs of the same paths in reverse order are omitted. In this case the set of efficient pairs of paths is

$$X_E = \{(p_1, p_2), (p_2, p_1), (p_2, p_2), (p_2, p_4), (p_4, p_2)\},$$

corresponding to the set of non-dominated points $Y_N = \{(2, 4), (1, 5)\}$.

Table 1. Pairs of paths from node 1 to node 4 in the network (N, A)

Path p_k	Path q	$l(p_k) \cap l(q)$	$z_1(p_k, q)$	$z_2(p_k, q)$
$p_1 = \langle 1, 2, 4\rangle$	$\langle 1, 2, 4\rangle$	$\{r, g, b\}$	3	4
	$\langle 1, 3, 4\rangle$	$\{r, g\}$	2	4
	$\langle 1, 2, 3, 4\rangle$	$\{r, g\}$	2	5
	$\langle 1, 3, 2, 4\rangle$	$\{g, b\}$	2	5
$p_2 = \langle 1, 3, 4\rangle$	$\langle 1, 3, 4\rangle$	$\{r, g\}$	2	4
	$\langle 1, 2, 3, 4\rangle$	$\{r, g\}$	2	5
	$\langle 1, 3, 2, 4\rangle$	$\{g\}$	1	5
$p_3 = \langle 1, 2, 3, 4\rangle$	$\langle 1, 2, 3, 4\rangle$	$\{r, g\}$	2	6
	$\langle 1, 3, 2, 4\rangle$	$\{g\}$	1	6
$p_4 = \langle 1, 3, 2, 4\rangle$	$\langle 1, 3, 2, 4\rangle$	$\{g, b\}$	2	6

Proposition 2.1. *For the SRDSP, there is a set $R \subseteq P \times P$ composed of pairs of loopless paths such that $Y_N = z(R)$.*

Proof. Let $(\bar{z}_1, \bar{z}_2) \in Y_N$ be a non-dominated point for the SRDSP, such that $\bar{z}_i = z_i(p, q)$, with $(p, q) \in X_E$, $i = 1, 2$. Let us assume, with no loss of generality, that p is a loopless path, contrary to q which has the form $q = q_1 \diamond C \diamond q_2$, where C is any of its loops. (The same reasoning can be applied in case p, or both p and q, have loops.) Therefore, the path $q^* = q_1 \diamond q_2$ has less loops than q. If q^* is not loopless, then the reasoning can be repeated as many times as necessary until a loopless path is obtained.

Because all the arcs in q^* are also in q, then $l(q^*) \subseteq l(q)$, and because there are no negative costs in the network, then $c(q^*) \leq c(q)$ also holds. Therefore,

$$z_1(p, q^*) \leq z_1(p, q) \quad \text{and} \quad z_2(p, q^*) \leq z_2(p, q),$$

and we must have

$$z_1(p, q^*) = z_1(p, q) = \bar{z}_1 \quad \text{and} \quad z_2(p, q^*) = z_2(p, q) = \bar{z}_2,$$

otherwise (p, q^*) would dominate (p, q) and this could not be an efficient solution, thence contradicting the assumption. □

The SRDSP problem differs from classical bicriteria path problems for two main reasons. The first is the fact that it aims at finding pairs of paths. The second has to do with the non-linear nature of the objective function z_1 that counts the number of risks shared by the two paths. Most research on path problems and K-shortest path problems has been focused on linear objective functions, but some works are devoted to non-linear versions of the problem, like in (Gabriel and Bernstein 2000; Reinhardt and Pisinger 2011; Gualandi and Malucelli 2012). Reviews on multicriteria path problems can be found in (Raith and Ehrgott 2009; Clímaco and Pascoal 2012).

2.3. Overview of the resolution algorithm

The exact algorithm in (Pascoal and Clímaco 2020) addresses two points: finding pairs of paths from s to t and handling the functions z_1 and z_2 from a bi-objective perspective. The first aspect involves a modification of the graph G where each path corresponds to a pair of paths in the original graph. The second involves the adaptation to the SRDSP problem of a bi-objective method based on ranking paths, presented in (Clímaco and Martins 1982). Both procedures are now briefly outlined.

The algorithm starts by transforming the network $G = (N, A)$ into a new network, $G' = (N', A')$, such that:

- each node $i \in N$ is duplicated as the node i';
- each arc (i, j) is duplicated as the arc (i', j'); and
- a new arc is added which links node t to node s', the arc (t, s').

The new sets of nodes and arcs are

$$\begin{aligned} N' &= N \cup \{i' : i \in N\}, \\ A' &= A \cup \{(i', j') : (i, j) \in A\} \cup \{(t, s')\}. \end{aligned} \tag{5}$$

The initial node of network G' is the node s, and the new terminal node is node t'. The arc costs and labels are also duplications of those in the network G, such that:

$$c_{i'j'} = c_{ij} \quad \text{and} \quad l_{i'j'} = l_{ij}, \quad \text{for any } (i, j) \in A. \tag{6}$$

Finally, $c_{ts'} = 0$ and $l_{ts'} = \{x\}$, with x an extra label, $x \notin L$ (details of Algorithm 1 can be found in (Pascoal and Clímaco 2020)).

Every path from s to t' in G' corresponds to a pair of paths from s to t in G and denoting by $\diamond$ the concatenation of two paths, the following result holds.

Lemma 2.2. *(Pascoal and Clímaco 2020) Any path p from s to t' in the network G' has the form: $p = q \diamond \langle t, s' \rangle \diamond r'$, with q a path from s to t and r' a path from s' to t'. Moreover,*

$$z_1(p) = |l(q) \cap l(r')| \quad \textit{and} \quad z_2(p) = c(q) + c(r'). \tag{7}$$

For solving the SRDSP problem, an adaptation of the algorithm in (Clímaco and Martins 1982) was used. This is based on ranking paths by non-decreasing order of the cost, z_2 which originates a sequence of solutions from which the efficient ones can be selected, $\{p_i\}_{i=1,\ldots,k}$. Contrary to the standard bi-objective shortest path problem, while the objective function z_2 in the SRDSP problem is linear, the function z_1 is non-linear, and so it is more difficult of handling, namely as far as finding its optimum is concerned. Still, a ranking algorithm can be applied for finding paths in G', according to z_2. Moreover, a subsequence of efficient solutions non-decreasing in z_1 can be obtained according to the following result.

Lemma 2.3. *(Pascoal and Clímaco 2020) Let $\{p_i\}_{i \geq 1}$ be the sequence of efficient paths from s to t' in G' with respect to (z_2, z_1); then, these paths can be arranged in a way that satisfies:*

$$z_2(p_i) < z_2(p_{i+1}) \textit{ and } z_1(p_i) > z_1(p_{i+1})$$

or

$$z_2(p_i) = z_2(p_{i+1}) \textit{ and } z_1(p_i) = z_1(p_{i+1}).$$

As a consequence, paths in G' can be ranked by order of z_2, and the dominance test proposed by (Clímaco and Martins 1982) prunes those which are dominated. This test consists of comparing the objective function values of each current solution with those of the latest non-dominated solution candidate. For ranking paths we used, in our implementation, the MPS algorithm, the deviation algorithm in (Martins et al. 1999), which is reviewed in Appendix A. Other ranking algorithms can be used instead, for instance (Yen 1971; Martins and Pascoal 2003; Carlyle and Wood 2005). At a given step of the method, say step k, let m_2 denote the cost of the latest path computed in G' until step k (which is the greatest cost found so far, given the paths ranking), and let M_1 denote the lowest number of common labels of the pairs of paths computed until step k. Because paths are ranked according to z_2, their costs never decrease. Then, given a new path p in G', the dominance test included in the algorithm is as follows:

- If $z_2(p) = m_2$ and $z_1(p) = M_1$, then p is added to the set of candidates to non-dominated solutions.
- If $z_2(p) = m_2$ and $z_1(p) < M_1$, then the candidate solutions are dominated. The path p is a new candidate to nondominated solution.
- If $z_2(p) = m_2$ and $z_1(p) > M_1$, then the path p is dominated.
- If $z_2(p) > m_2$ and $z_1(p) = M_1$, then the path p is dominated and the current solutions, in the set of candidates, are non-dominated.
- If $z_2(p) > m_2$ and $z_1(p) < M_1$, then the current solutions, in the set of candidates, are non-dominated and the path p is a new candidate to nondominated solution.
- If $z_2(p) > m_2$ and $z_1(p) > M_1$, then the current solutions, in the set of candidates, are non-dominated and the path p is dominated.

In addition, m_2 and M_1 must be updated during the process, whenever $z_2(p) < m_2$ or $z_1(p) < M_1$. The algorithm starts by computing the shortest path in G' with respect to z_2, say p. This path is then used to initialize m_2 and M_1 ($m_2 = z_2(p)$, $M_1 = z_1(p)$). The paths are ranked until all solutions have been computed or an acceptable pair of paths with respect to the number of shared labels has been found. The pseudo-code of the algorithm used for the SRDSP problem, designated as `BRRA` (Bicriteria Resilient Routing Algorithm) is outlined in Algorithm 1.

Note that this base algorithm was customized for the envisaged application. In particular, we could easily introduce an upper bound on the total cost of the two paths, remembering that these costs are calculated by non-decreasing order. Such bound should only be considered for solutions with previously fixed acceptable number of common risks between the active and the protection paths. This feature enables, in many practical situations, the saving of run time as compared to the complete running of the base algorithm. Furthermore, in the application, we could take into account failure probabilities associated with the arcs if such information was available to the network designer so that trade-offs between risk-disjointness and total cost of the two paths may be analysed for a limited number of calculated Pareto optimal solutions. An illustrative example of this type of procedures is shown in Section 3.5. As an illustrative example of the working of the algorithm Table 2 lists the pairs of paths from $s = 1$ to $t = 4$ in the network in Figure 1 obtained by a ranking algorithm with respect to the paths cost. Applying the dominance test described above allows the set X_E to be obtained, according to the remarks in the last column of the table. The efficient pairs of paths are marked with a star.

Algorithm 1: Finding the non-dominated SRDSPs (`BRRA`)

$G' \leftarrow$ Duplicate the network G `// Network modification`
$p_2^* \leftarrow$ Shortest pair of paths from s to t' with respect to z_2 in G'
$m_2 \leftarrow z_2(p_2^*)$
$M_1 \leftarrow z_1(p_2^*)$
$k \leftarrow 0$
$P_N \leftarrow \emptyset$
$P_X \leftarrow \{p_2^*\}$
while *there are paths left to rank and a stopping condition is not met* **do**
    $k \leftarrow k + 1$
    $p_k \leftarrow k$-th shortest path from s to t' with respect to z_2 in G'
    if $z_2(p_k) = m_2$ **then**
        if $z_1(p_k) = M_1$ **then** $P_X \leftarrow P_X \cup \{p_k\}$
        if $z_1(p_k) < M_1$ **then**
            $M_1 \leftarrow z_1(p_k)$
            $P_X \leftarrow \{p_k\}$
    else
        if $z_1(p_k) < M_1$ **then**
            $P_N \leftarrow P_N \cup P_X$
            $m_2 \leftarrow z_2(p_k)$
            $M_1 \leftarrow z_1(p_k)$
            $P_X \leftarrow \{p_k\}$

3. Application study

The code `BRRA` was applied to different benchmark telecommunication network topologies described in (Orlowski et al. 2010), listed in Table 3, with n, m being the number of nodes, arcs, respectively. The implementation used C language and the MPS algorithm (Martins et al. 1999) to rank the loopless paths. The tests ran on an Intel® i7-6700 Quad core, with 8Mb of cache, a 3.4 GHz processor and 16 Gb of RAM, over openSUSE Leap 42.2.

The used reference networks are undirected. To use them in the application study and having in mind that the corresponding logical networks are directed, each edge $\{i, j\}$ is duplicated as two directed arcs in opposite directions, (i, j) and (j, i). The results presented hereafter are average values found for 10 seeds of the random number generator and 45 origin-destination pairs. For each arc $(i, j) \in A$, $c_{ij} = 1/b_{ij}$ represents the cost of the link occupation, b_{ij} being the available bandwidth, the values of which are randomly generated according to the distributions in Table 4, in the sets:

$I_i = \{2 + 2k : k = 20i, \ldots, 20i + 19\}$, $i = 0, 1, 2$;
$I_3 = \{2 + 2k : k = 60, \ldots, 78\}$.

The distributions D1, D2 and D3 represent uniformly, highly and lightly loaded networks, respectively. The sets of SRLGs/labels, L_{ij}, assigned to the arcs are uniformly generated between 1 and $|L| = 15, 20, 25$, with mean number of SRLGs per arc $\alpha = 1, 2, 4$.

Table 2. Ranked pairs of paths from node 1 to node 4 in the network (N, A)

Path pairs $(p_k, p_{k'})$	$(z_1(p_k, p_{k'}), z_2(p_k, p_{k'}))$	Remark
(p_1, p_1)	(3,4)	
$(p_1, p_2)^*$	(2,4)	Dominates the previous pair
$(p_2, p_1)^*$	(2,4)	
$(p_2, p_2)^*$	(2,4)	
(p_1, p_3)	(2,5)	Dominated
(p_1, p_4)	(2,5)	Dominated
(p_2, p_3)	(2,5)	Dominated
$(p_2, p_4)^*$	(1,5)	
(p_3, p_1)	(2,5)	Dominated
(p_3, p_2)	(2,5)	Dominated
(p_4, p_1)	(2,5)	Dominated
$(p_4, p_2)^*$	(1,5)	
(p_3, p_3)	(2,6)	Dominated
(p_3, p_4)	(1,6)	Dominated
(p_4, p_3)	(1,6)	Dominated
(p_4, p_4)	(2,6)	Dominated

Table 3. Test parameters

Network	n	m	$\delta = m/n$	$\|L\|$	α
`NSFFixedLabels`	11	52	4.7	21	1.5
`NSFRandomLabels`	14	42	3.0	15, 20, 25	1, 2, 4
`France`	25	90	3.6	15, 20, 25	1, 2, 4
`Cost266`	37	114	3.1	15, 20, 25	1, 2, 4
`Germany50`	50	176	3.5	15, 20, 25	1, 2, 4

3.1. `NSFFixedLabels` *and* `NSFRandomLabels` *networks*

The code `BRRA` was capable of solving until the end all the instances for fixed (this corresponds to a practical instance of the network structure) and for randomly generated SRLGs. The average run times shown in Figures 2 and 3, of the order of ms, do not vary much with $|L|$ and tend to increase with α for each given distribution, as might be expected. From the structure of the underlying combinatorial problem one may not conclude the existence of a congruence between the increase (or decrease) in $|L|$ and the average run times, contrary to what occurs with α. In fact, from these results and from further experimentation we observed different types of situations (concerning average run time variations with respect to $|L|$) in different network topologies and, in some cases, even for the same network with different available bandwidth distributions, as shown for example in Figure 8.

3.2. `France` *network*

The results for the instances concerning the `France` network are shown in Table 5 and in Figures 4, 5. The code `BRRA` was capable of finding all the Pareto optimal pairs of paths in almost all the cases, that is, in at least 94% of the problems (this worst case was for distribution D2), although the algorithm halted due to memory constraints, in a significant number of cases, specially for $\alpha = 4$. If the algorithm is halted due to

Table 4. Available bandwidth distributions

	I_0	I_1	I_2	I_3
D1	25%	25%	25%	25%
D2	70%	15%	10%	5%
D3	18%	18%	18%	46%

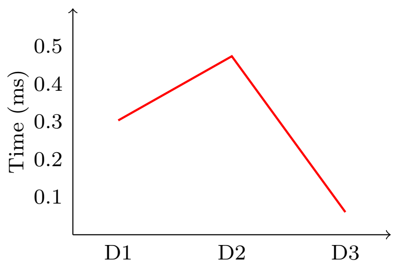


Figure 2. `NSFFixedLabels`: Run times for problems solved until the end

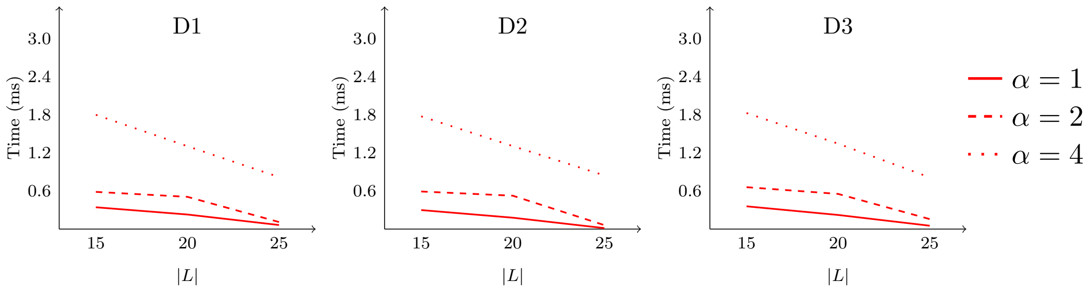


Figure 3. `NSFRandomLabels`: Run times for problems solved until the end

memory constraints, the set of efficient path pairs obtained until that moment, P_X, is output and two situations may occur:

(1) all the non-dominated points have been calculated, that is, $z(P_X) = Y_N$;
(2) not all the non-dominated points have been calculated, that is, $z(P_X) \subsetneq Y_N$.

A related problem aiming at the calculation of a lexicographic optimal pair of paths with respect to (z_1, z_2) was addressed in (Pascoal et al. 2022), where an exact combinatorial algorithm and an integer linear programming (ILP) formulation were presented. The solution for the lexicographic problem corresponds to a point of the Pareto frontier of our bicriteria problem and its objective function values coincide with those obtained by the code `BRRA` if running with no limitations. Therefore, the solutions obtained by `BRRA` were compared with the lexicographic optimal solution of the ILP formulation given in (Pascoal et al. 2022) to distinguish situations (1) and (2), and calculate the percentage of cases for which all the efficient solutions were obtained. These values are summarized in Table 5. In the following, when `BRRA` runs normally we say that the problem is solved until the end (shortly designated as STE). Whenever `BRRA` runs normally or it is halted due to memory limitations but situation (1) occurs, then we say that all efficient solutions are found (shortly designated as AES). As for the run times, the trend is still that they do not change much with the bandwidth distribution but tend to increase with α as expected. The problems that were not halted due to memory storage ran in up to 1.5s, whereas if one considers all the instances for which it was possible to find all the non-dominated solutions the code required up to 4.5s. It is worth noting that the samples used to obtain the average values shown in Figures 4 and 5 are different, given that different instances may have been solved until the end or such that all non-dominated points were obtained.

Table 5. Results for `France`

		$\alpha=1$			$\alpha=2$			$\alpha=4$		
Dist.\\$|L|$		15	20	25	15	20	25	15	20	25
D1	STE (%)	92	86	92	59	68	83	23	33	43
	AES (%)	97	98	98	97	98	97	98	98	98
D2	STE (%)	89	85	89	59	68	85	23	33	43
	AES (%)	94	95	94	97	98	100	99	98	98
D3	STE (%)	94	87	94	62	69	85	25	34	44
	AES (%)	100	100	100	98	98	98	100	100	99

STE: Problems solved until the end; AES: Problems for which all efficient solutions were found

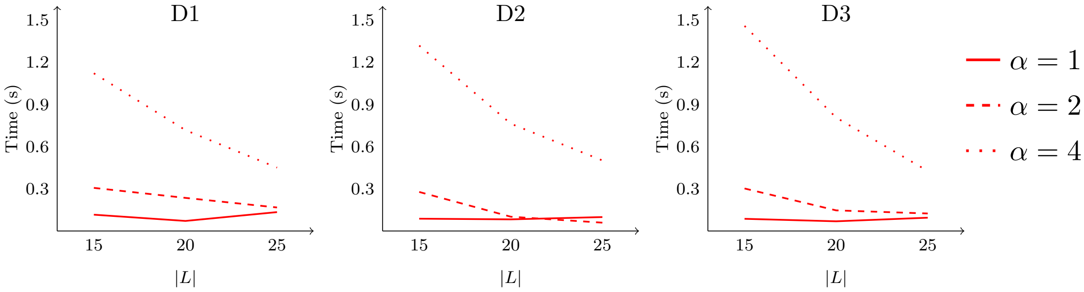


Figure 4. `France`: Run times for problems solved until the end

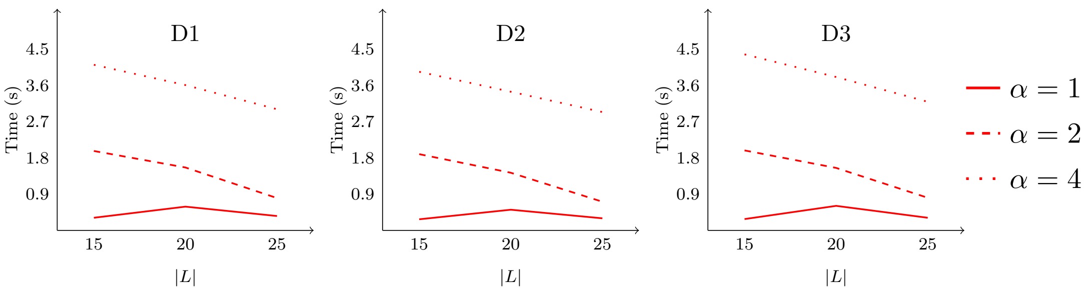


Figure 5. `France`: Run times for problems for which the whole Pareto front was found

Table 6. Results for `Cost266`

		$\alpha = 1$			$\alpha = 2$			$\alpha = 4$		
Dist.\\$|L|$		15	20	25	15	20	25	15	20	25
D1	STE (%)	83	93	95	21	31	57	3	7	11
	AES (%)	92	98	98	89	84	92	88	87	77
D2	STE (%)	86	95	96	22	33	60	3	7	12
	AES (%)	95	100	98	98	96	99	96	95	97
D3	STE (%)	78	87	88	20	29	53	3	7	11
	AES (%)	87	91	90	87	82	85	93	88	87

STE: Problems solved until the end; AES: Problems for which all efficient solutions were found

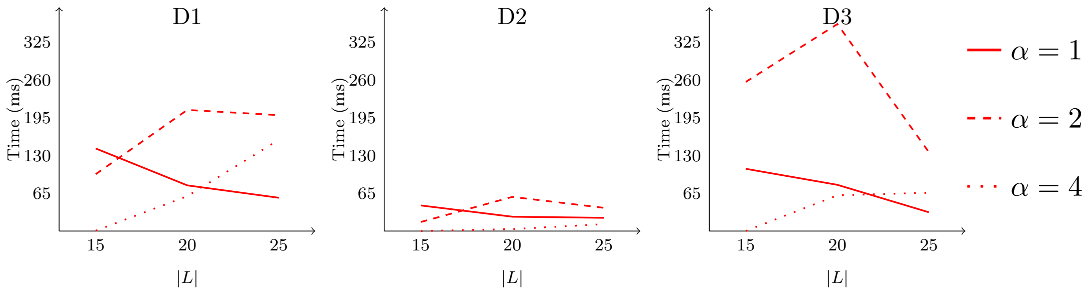


Figure 6. `Cost266`: Run times for problems solved until the end

3.3. Cost266 network

The percentage of problems solved until the end decreased very significantly for $\alpha = 2, 4$ but the percentage of problems for which all efficient solutions were found is still very high, as shown in Table 6. As for the average run times, they were lower (always less than 350ms), in the former set of instances (Figure 6), than for the France network and of the same order of magnitude for the latter set (up to 4.9s), see Figure 7. The variations with the distributions and their defining parameters $\alpha, |L|$ follow the same trends as for `France` network.

3.4. Germany50 network

This network has significantly more nodes, arcs and connectivity than the previous ones justifying why less percentage of instances were solved until the end by `BRRA`, as can be seen in Table 7, especially for $\alpha = 4$ (worst-case scenario). Nevertheless, `BRRA`

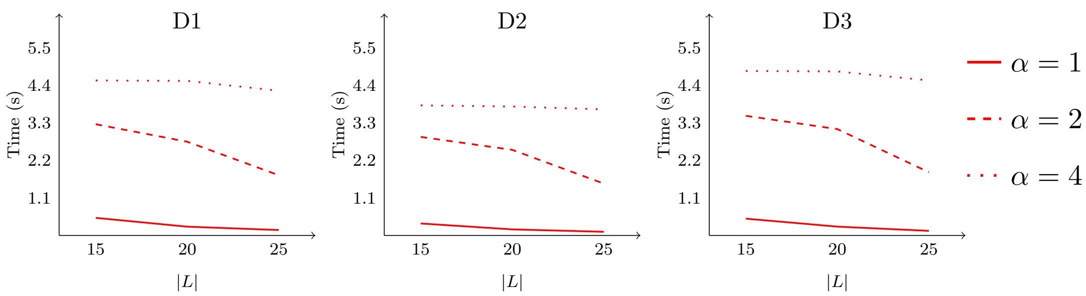


Figure 7. `Cost266`: Run times for problems for which the whole Pareto front was found

Table 7. Results for `Germany50`

		$\alpha = 1$			$\alpha = 2$			$\alpha = 4$		
Dist.\\$|L|$		15	20	25	15	20	25	15	20	25
D1	STE (%)	79	84	90	29	45	64	3	8	11
	AES (%)	80	90	92	66	70	77	68	71	67
D2	STE (%)	93	94	98	38	54	78	3	9	14
	AES (%)	95	100	100	87	87	94	89	91	86
D3	STE (%)	71	77	84	26	38	57	3	7	10
	AES (%)	72	82	84	57	59	68	61	63	56

STE: Problems solved until the end; AES: Problems for which all efficient solutions were found

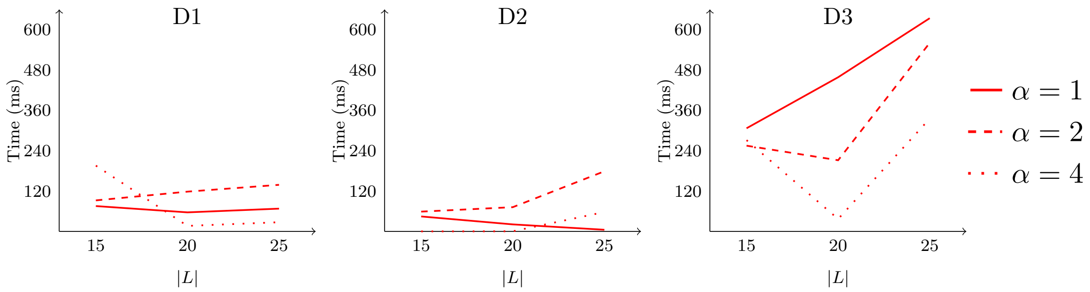


Figure 8. `Germany50`: Run times for problems solved until the end

was still capable of obtaining the whole Pareto solutions set for between 56% and 100% of all the instances. As for the average run times these are of the same order of magnitude as in `Cost266` network for the former set of instances (Figure 8) excepting for D3 where they tend to be higher, and of the same order of magnitude for the latter set (Figure 9), excepting for D3, $\alpha = 2$ where they tend to be lower.

Overall, we conclude that the algorithm was capable of calculating exactly the whole Pareto set in most cases or, at least, a major part of this set, in the reference test networks, still in times compatible with a wide range of applications to telecommunication network routing design problems.

3.5. Solution selection procedures

Concerning the selection of an efficient solution in the obtained Pareto set, network designers tend to choose the lexicographic optimal solution (corresponding to z_1^*, the

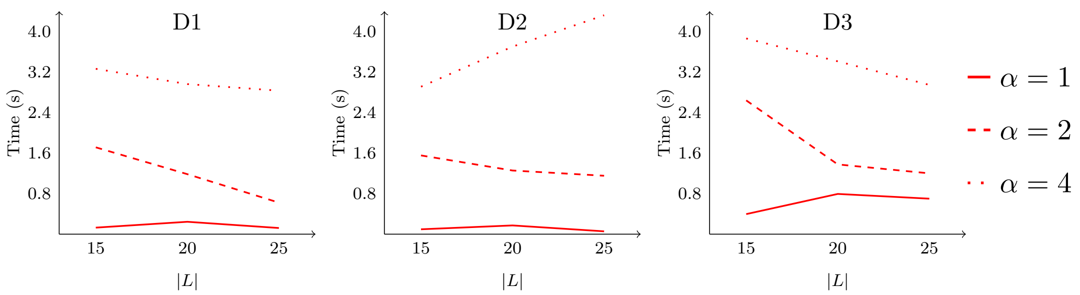
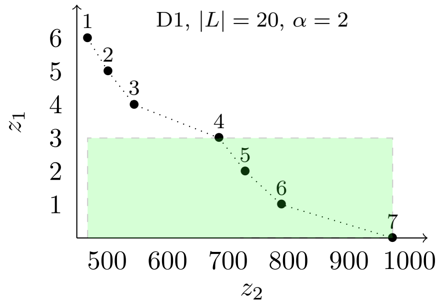


Figure 9. `Germany50`: Run times for problems for which the whole Pareto front was found

minimal z_1), seeking to minimise the probability of end-to-end connection failure, in order to guarantee maximal service availability. Hence, we admit that, in practice, one might consider the analysis (in a bicriteria context) of non-dominated solutions with a number of common labels of $z_1 = z_1^* + i$, $i = 0, 1, 2, 3$, and thus analyse possible trade-offs between SRLG disjointness and cost.

An illustrative example for an instance in the `France` network is shown in Figure 10. In this case only the 4 solutions in the green area of the Pareto set would be analysed. An obvious choice would the lexicographic optimum (solution 7), which ensures full SRLG disjointness, or, alternatively, solution 6 if the designer (or possibly a corporate customer exploring the services associated with this end-to-end connection) accepts a minimal worsening in optimal resilience for obtaining a gain of approximately 19% in total route cost. Similar examples could be shown for other instances/networks.



Sol.	z_2	z_1
1	467.814	6
2	501.616	5
3	544.921	4
4	685.217	3
5	728.522	2
6	788.572	1
7	972.173	0

$z_1^{req} = l^* = 0$

$z_1^{acc} = 3$

$z_2^{acc} = 467.814$

Figure 10. Non-dominated solutions for a selected instance in `France` network

As noted in the introduction, some situations may arise for which the lexicographic solution has one or more SRLGs common to both paths and the risk occurrence probabilities vary significantly. Now we will show that, in some circumstances, the consideration of alternative efficient solutions is not only advisable in terms of trade-off analysis but also may be useful for selecting a solution with maximal survivability value for the end-to-end connection supported by the pair of paths.

Let us consider a routing situation in a network such that the lexicographic optimal solution of the problem, $x_1^* = (p_1, q_1)$, has $z_1(x_1^*) = l^* = 1$ and the failure probability assigned to the risk common to p_1 and q_1 is known, P_1. Let us also assume that the algorithm calculated another efficient solution, $x_2 = (p_2, q_2)$, with $z_1(x_2) = l^* + 1$, the failure probabilities for the two associated common risks being P_2 and P_3, with $P_2, P_3 < P_1$. This may clearly arise in physical networks with significant asymmetries in failure probabilities of its components. Thence the end-to-end connection failure probability (corresponding to the service unavailability) is simply given by the total probability theorem, assuming statistical independency among failures of the physical network elements:

$$U(x_2) = P_2 + P_3 - P_2P_3 \approx P_2 + P_3 \quad (P_2, P_3 \ll 1).$$

Therefore, for all cases such that $U(x_2) < P_1 = U(x_1^*)$ the efficient solution x_2 could be optimally resilient, unlike the lexicographic optimum, x_1^*. A realistic numerical example follows, considering values of the same magnitude as in the examples addressed in (de Sousa et al. 2019) in the context of geodiverse routing methods for disaster resilience scenarios. If

$$R_1 = 0.9940 = 1 - P_1,$$

$$R_2 = 0.9995 = 1 - P_2,$$

$$R_3 = 0.9994 = 1 - P_3,$$

then,

$$U(x_1^*) = 0.60\% > U(x_2) = 0.11\%.$$

Moreover, from the definition of non-dominance it results that the cost of solution x_2 is less than the cost of x_1^*, so that, in this type of case, the efficient solution x_2 would be clearly preferable as compared to the lexicographic optimum. This type of argument could be naturally applied to other efficient solutions, by performing an *a posteriori* analysis, based on information on failure probabilities. This illustrates the capabilities of the proposed approach.

4. Concluding remarks

We presented a bicriteria optimisation approach for the telecommunication network routing design problem with path protection and the implementation of an adequate exact resolution method for the formulated bicriteria problem, based on the algorithm in (Pascoal and Clímaco 2020), enabling to obtain a set of exact Pareto optimal solutions. Also, an application study using reference telecommunication network topologies considering random arc bandwidth occupations and random SRLGs/risks assignments to the arcs of the associated logical networks, was developed. The description of a procedure for selecting a solution in the obtained Pareto optimal solution set, suitable for the application in telecom network routing design, was put forward. This may be useful in telecommunication network resilient routing design for analysis of possible trade-offs between SRLG disjointness and total cost. Experiments for assessing the performance of the method on reference test networks have shown that the method is capable of calculating the whole Pareto set in the vast majority of cases or, at least, a major part of this set. In general, the run times increased with the size/connectivity of the networks and, in most cases, with the increase in the mean number of SRLGs per arc. Another advantage of the method is that it calculates, as a subproduct, the lexicographic optimal solution, a common approach to the routing design problem. Furthermore, we have shown that, in some circumstances, which may arise in telecommunication network routing design, including geodiverse routing models, the proposed approach may be useful for analysing possible optimally resilient solutions in the Pareto optimal set. These features make the proposed approach and resolution method suitable for a wide range of applications on resilient routing design of telecommunication networks (namely optical fiber based transport networks and SDNs – Software Defined Networks), including off-line dynamic routing.

Funding

This work was partially supported by the Portuguese Foundation for Science and Technology (FCT) under project grants UID/MAT/00324/2020 and UIDB/00308/2020.

References

Carlyle, W. M. and Wood, R. K. (2005). Near-shortest and K-shortest simple paths. *Networks: An International Journal*, 46:98–109.

Clímaco, J. and Craveirinha, J. (2019). *MCDA/M in Telecommunication Networks: Challenges and Trends*, pages 11–55. Chapman and Hall/CRC, New York, NY.

Clímaco, J. and Martins, E. (1982). A bicriterion shortest path algorithm. *European Journal of Operational Research*, 11:399–404.

Clímaco, J. and Pascoal, M. (2012). Multicriteria path and tree problems: discussion on exact algorithms and applications. *International Transactions in Operational Research*, 19:63–98.

de Sousa, A., Gomes, T., Girão-Silva, R., and Martins, L. (2019). Minimization of the network availability upgrade cost with geodiverse routing for disaster resilience. *Optical Switching and Networking*, 31:127–143.

Gabriel, S. and Bernstein, D. (2000). Nonadditive shortest paths: subproblems in multi-agent competitive network models. *Computational & Mathematical Organization Theory*, 6:29–45.

Gomes, T. and Craveirinha, J. (2010). An algorithm for enumerating SRLG diverse path pairs. *Journal of Telecommunications and Information Technology*, pages 5–12.

Gomes, T., Jorge, L., Melo, P., and Girão-Silva, R. (2016). Maximally node and SRLG-disjoint path pair of min-sum cost in GMPLS networks: a lexicographic approach. *Photonic Network Communications*, 31:11–22.

Gomes, T., Jorge, L., Melo, P., Girão-Silva, R., and Mendes, S. (2013a). Calculating a maximally node and SRLG-disjoint path pair of min-sum cost in GMPLS networks. In *9th International Conference on the Design of Reliable Communication Networks (DRCN 2013)*, pages 314–321.

Gomes, T., Simões, C., and Fernandes, L. (2013b). Resilient routing in optical networks using SRLG-disjoint path pairs of min-sum cost. *Telecommunication Systems Journal*, 52:737–749.

Gualandi, S. and Malucelli, F. (2012). Resource constrained shortest paths with a super additive objective function. In *International Conference on Principles and Practice of Constraint Programming*, pages 299–315. Springer.

Hu, J. Q. (2003). Diverse routing in optical mesh networks. *IEEE Transactions on Communications*, 51:489–494.

Martins, E. and Pascoal, M. (2003). A new implementation of Yen's ranking loopless paths algorithm. *4OR – Quarterly Journal of the Belgian, French and Italian Operations Research Societies*, 1:121–134.

Martins, E., Pascoal, M., and Santos, J. (1999). Deviation algorithms for ranking shortest paths. *The International Journal of Foundations of Computer Science*, 10:247–263.

Orlowski, S., M., R. W., Pióro, and Tomaszewski, A. (2010). SNDlib 1.0 – survivable network design library. *Networks*, 55:276–286.

Pascoal, M. and Clímaco, J. (2020). On a relaxed maximally disjoint path pair problem: A bicriteria approach. *International Transactions in Operational Research*, 27:2045–2063.

Pascoal, M., Craveirinha, J., and Clímaco, J. (2022). An exact lexicographic approach for the maximally risk-disjoint/minimal cost path pair problem in telecommunication networks. *TOP*, 30:405–425.

Raith, A. and Ehrgott, M. (2009). A comparison of solution strategies for biobjective shortest path problems. *Computers & Operations Research*, 36:1299–1331.

Reinhardt, L. B. and Pisinger, D. (2011). Multi-objective and multi-constrained non-additive shortest path problems. *Computers & Operations Research*, 38:605–616.

Rostami, M., Khorsandi, S., and Khodaparast, A. (2007). CoSE: A SRLG-disjoint routing algorithm. In *IV European Conference on Universal Multiservice Networks*, pages 86–92.

Silva, J., Gomes, T., Fernandes, L., Simões, C., and Craveirinha, J. (2011). An heuristic for maximally SRLG-disjoint path pairs calculation. In *2011 3rd International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 1–8.

Todimala, A. and Ramamurthy, B. (2004). IMSH: An iterative heuristic for SRLG diverse routing in WDM mesh networks. In *Proceedings 13th International Conference on Computer Communications and Networks (ICCCN 2004)*, pages 199–204. IEEE.

Yen, J. (1971). Finding the K shortest loopless paths in a network. *Management Science*, 17:712–716.

Appendix A. MPS algorithm for ranking loopless paths

In the following we review the algorithm proposed in (Martins et al. 1999) for ranking K loopless paths by non-decreasing order of cost, for a given $K \in \mathbb{N}$.

Let X be a set that stores candidates to the k-th loopless path candidates, p_k, for $k = 1, \dots, K$. The set X is initialised with the shortest path from s to t in (N, A), p_1, and after paths $p_1, \dots, p_{k-1}$ having been determined, the path p_k is the next shortest path currently stored in the set X. When path p_k is selected and removed from X, its nodes are analysed in order to generate new candidate paths with a low cost. The shortest deviation of p_k at node v_i is given by the shortest path from v_i to t, after deleting the arc of p_k that starts at node v_i. This best path has the form:

$$p = \text{sub}_{p_k}(s, v_i) \diamond (v_i, j) \diamond T_t(j),$$

where the arc $(v_i, j) \in A$ does not belong to any of the candidate paths computed so far, T_t denotes the tree of the shortest paths from any node to t, $T_t(j)$ denotes the path from j to t in T_t, $\diamond$ stands for the concatenation of paths, and $\text{sub}_p(u, v)$ is the subpath of path p between nodes u and v. Node $v_i \in N$ is called the deviation node of path p, and will be denoted by d_p.

The MPS algorithm relies on two important ideas to define the new deviations to consider for each path p_k. The first is the use of reduced costs instead of the usual costs associated with the network arcs. The second is to represent the network in the sorted forward star form.

Let π_i denote the cost of the path in T_t from any $i \in N$ to node t. The reduced cost associated with $(i, j) \in A$ is defined by

$$\bar{c}_{ij} = c_{ij} - \pi_i + \pi_j$$

and it satisfies:

- $\bar{c}_{ij} \geq 0$, for any $(i, j) \in A$,
- and $\bar{c}_{ij} = 0$, for any $(i, j) \in T_t$.

Moreover,

$$c(p) = c(T_t(s)) + \bar{c}(p),$$

for any path p from s to t, and the use of reduced costs preserves the order in paths

$p_1, \ldots, p_K$. Therefore, $\bar{c}(T_t(u)) = 0$, for any $u \in \mathcal{N}$, and when scanning node $v \in p_k$, the deviation (v, u) with the minimum reduced cost has to be determined. Additionally, in the sorted forward star form network deviations are presented in non-decreasing order of their tail nodes, and deviations with the same tail node are sorted by non-decreasing order of their reduced costs.

The generation of a new candidate by this method depends on the selection of the arc (v_i, j), chosen in a way that the nodes of a path should not be repeated. This procedure may still generate some paths with loops, whenever $\text{sub}_{p_k}(s, v_i)$ and $T_t(j)$ share nodes. Therefore, only arcs that start at a node previous to the first loop have to be considered. An outline of this algorithm is presented in Algorithm 2.

Algorithm 2: MPS algorithm for ranking K loopless paths by order of cost

$T_t \leftarrow$ tree of the shortest paths from any $i \in N$ to t
for $(i, j) \in A$ **do** $\bar{c}_{ij} \leftarrow \pi_j - \pi_i + c_{ij}$
Represent A in the sorted forward star form
$p \leftarrow T_t(s)$
$d_p \leftarrow s$
$X \leftarrow \{p\}$
$k \leftarrow 0$
while $X \neq \emptyset$ *and* $k < K$ **do**
    $p \leftarrow \text{argmin}\{p \in X : \bar{c}(p)\}$ // $p = \langle v_1, v_2, \ldots, v_r \rangle$
    $X \leftarrow X - \{p\}$
    for $i = d(p), d(p)+1, \ldots, r-1$ *such that* $\text{sub}_p(s, v_i)$ *is loopless* **do**
        $(v_i, v) \leftarrow$ arc that follows (v_i, v_{i+1}) in A such that $v \notin \text{sub}_p(s, v_i)$
        if (v_i, v) *is defined* **then**
            $q \leftarrow \text{sub}_p(s, v_i) \diamond (v_i, v) \diamond T_t(v)$
            $d_q \leftarrow v_i$
            $X \leftarrow X \cup \{q\}$
        if $v_i = v_{r-1}$ **then**
            $k \leftarrow k + 1$
            $p_k \leftarrow p$