

# Maximally node and SRLG-disjoint path pair of min-sum cost in GMPLS networks: a lexicographic approach

Teresa Gomes<sup>1,2</sup> · Luísa Jorge<sup>2,3</sup> · Paulo Melo<sup>2,4</sup> · Rita Girão-Silva<sup>1,2</sup>

Received: 2 September 2014 / Accepted: 15 June 2015 / Published online: 30 June 2015  
© Springer Science+Business Media New York 2015

**Abstract** Path protection is a fast and capacity-efficient approach for increasing the availability of end-to-end connections. However, sometimes it is not possible to obtain a fully disjoint path pair. In this case, it may be admissible to consider a path pair which is as disjoint as possible, and thus provide the best (in a certain sense) level of the single-fault protection that can be ensured using this type of approach. A shared risk link group (SRLG) is a group of links which have a common risk of failure. Two new heuristics for solving the min-sum maximally node and SRLG-disjoint path pair are presented. The relative performance of the new heuristics and also of two other previously proposed heuristics is evaluated using four different networks. Results, regarding accuracy and execution time of the studied heuristics, show that one of the new proposed algorithms can be a good compromise for use in the Generalized Multi-protocol Label Switching control plane.

**Keywords** Min-sum · SRLG maximally disjoint · Multi-objective

## 1 Introduction

Path protection is an effective recovery approach to increasing the availability of end-to-end connections. In normal conditions of the network, i.e., when there are no faults, the AP (active path, working path or primary path) is used. When a fault occurs, preventing the traffic from being carried in the AP, the BP (backup path or secondary path) is used to carry that traffic. For a network to be reliable, both paths must not share a common risk of failure. In this context, the concept of shared risk link group (SRLG) was introduced for the failure management in some telecommunication networks (e.g., Generalized Multi-protocol Label Switching—GMPLS). GMPLS [12] is deployed in the control plane of optical networks and is a technology that adaptively maps traffic flows onto the physical topology of a network and allocates resources to these flows [17]. Therefore, GMPLS performs a mapping between the IP and optical layers, so that all the necessary traffic-engineering functions, along with different protection and restoration capabilities, can be supported.

An SRLG is a group of network links that share a common risk, e.g., by sharing a physical resource, like a cable, for which a failure is equivalent to a failure of all the links in that group. The routing protocols in GMPLS networks support the distribution of SRLG information, and efforts have been made to extend the Resource Reservation Protocol-Traffic Engineering (RSVP-TE) [2] to support the automatic collection of SRLG information [30].

The AP and the BP should constitute a pair of SRLG-disjoint paths, i.e., the links along the AP should not share

✉ Teresa Gomes  
teresa@deec.uc.pt

Luísa Jorge  
ljorge@inescc.pt

Paulo Melo  
pmelo@inescc.pt

Rita Girão-Silva  
rita@deec.uc.pt

<sup>1</sup> Department of Electrical and Computer Engineering, University of Coimbra, 3030-290 Coimbra, Portugal

<sup>2</sup> INESC Coimbra, 3000-033 Coimbra, Portugal

<sup>3</sup> Polytechnic Institute of Bragança, 5301-857 Bragança, Portugal

<sup>4</sup> Faculty of Economics, University of Coimbra, 3004-512 Coimbra, Portugal

any common SRLGs with the links along the BP, so that a failure-independent protection can be provided to a connection. When no SRLG-disjoint path pairs can be obtained, then the aim of the routing algorithms can be the attainment of the least SRLG-coupled (maximally SRLG-disjoint) path pair. A path computation element (PCE) [5] should be able to efficiently calculate maximally SRLG-disjoint path pairs (which may possibly be totally SRLG disjoint).

Algorithms for obtaining totally disjoint SRLG path pairs can be found in [19,24,28]. In [19], a link-disjoint algorithm is extended to SRLG-disjoint routing, originating the CoSE (Conflicting SRLG Exclusion) algorithm. This algorithm solves the SRLG-disjoint path problem formulated as a min–min problem. In [7], a new version of the CoSE heuristic is proposed, to solve a min-sum problem, designated as CoSE-MS. In [24], the SRLG-disjoint path problem is also formulated as a min-sum problem and it is solved by an iterative heuristic, the Iterative Modified Suurballe's Heuristic (IMSH). In [28], a trap avoidance (TA) scheme is proposed, where the considered algorithm tries to avoid falling into traps (i.e., situations where the algorithm is not able to find SRLG-disjoint path pairs although they exist). In [11], segment protection is used for the purpose of avoiding traps. Other algorithms for obtaining SRLG-disjoint path pairs can be mentioned. In [15], an algorithm (weighted SRLG or WSRLG) based on the  $k$ -shortest path algorithm is considered, where costs are assigned to the links according to the number of SRLG members related to a link. The idea of assigning costs to the links according to the risks they share with other links is also considered in [29], where an active SRLG-diverse path selection (ASPS) algorithm is proposed. This algorithm allows for an improvement on the resource utilization ratio (as bandwidth resources are shared among BPs) and for the consideration of differentiated reliability (DiR) requirements of different customers. A shared protection context is also considered in [18]. A hybrid protection scheme is proposed in [21], where shared-path and shared-link protections are jointly used in WDM networks with SRLGs. In [25], two different algorithms are proposed to calculate an optimal path pair for a given origin-destination pair. In this context, the optimality of a path pair may be defined in two different ways, leading to two different versions of an impairment aware optimal path pair problem. A different context is considered in [4], where the authors try to find a AP and several BPs from a pre-computed routing table with the minimal total additional bandwidth to carry a specific connection.

The focus of this paper is finding maximally disjoint SRLG path pairs, for coping with situations where no SRLG-disjoint path pairs can be obtained in the network. There are some variants of SRLG-disjoint path pair calculation algorithms, in order to calculate maximally disjoint SRLG path pairs. In [19], a variant of the CoSE, the Modified CoSE, is

proposed. In this algorithm, if there is no SRLG-disjoint path pair, then the SRLG path pair where the AP is shortest and the BP has the smaller number of common SRLGs with the AP will be used. In [23], a variant of the TA algorithm [28] designated TA-max is proposed. In TA-max, the candidate APs are considered iteratively (in order to ascending cost) and a BP with minimum cost is calculated for each candidate AP. The final solution is the path pair with the least number of common SRLGs.

In [10], the author proves that the diverse routing problem is NP complete, but ILP formulations may still be viable methods to solve this problem. Formulations for finding two diversely SRLG paths with minimum total cost or least coupled SRLG paths are proposed. In [26], a two-step method is used to solve the network provisioning problem. In the first step, the diverse routing problem is formulated using ILP to find optimal routes with the minimum objective value (either cost or distance). The solution might be total or maximally SRLG-diverse routes (if no SRLG-diverse routes exist). The second step consists of a dense wavelength division multiplexing system selection, regenerator placement and wavelength assignment.

In [14], a heuristic restoration scheme called Two-Step Partial Protection Risk Algorithm is proposed. With this partial SRLG-diversity algorithm, three major problems are tackled: network survivability, optimization of spare bandwidth and the possibility of differentiated classes of service. In [6], a method is proposed to abstract a detailed topology into an aggregated topology which only shows limited information about the network, due to confidentiality and management considerations. This topology aggregation is SRLG aware and a pair of inter-domain paths with a minimum set of common SRLGs may be obtained, in order to generate a more survivable aggregated topology. In [20], maximally SRLG-disjoint paths are obtained with an algorithm based on an ant colony optimization approach. The application of this algorithm is limited to networks where a link can belong to a maximum of two SRLGs. In [22], the aim is the minimization of the network cost while ensuring the network survivability. The links with higher degree of usage are considered as critical links, as the survivability of the network may be severely affected in the event of failure of such links. An approach is proposed, in which a partial critical duct-disjoint BP is provided by choosing the BP with the least number of critical ducts (i.e., ducts that carry critical links) to make the impact of duct failures as low as possible.

This paper is focused on the efficient calculation of maximally node and SRLG-disjoint path pairs of min-sum cost. The main contribution of this paper is the proposal of two new heuristics for solving a multi-objective optimization problem of maximally node and SRLG-disjoint path pairs. A detailed analysis of the relative performance of four algorithms for

solving the problem is also presented. Results show that one of the new proposed heuristics is adequate for use in the control plane of GMPLS networks, because it achieves a good compromise between accuracy and execution time.

The paper is organized as follows: In Sect. 2, the problem of calculating a maximally node and SRLG-disjoint path pair of minimal additive cost is formulated as a multi-objective problem; in Sect. 3, the two heuristics are presented; in Sect. 4, the experimental conditions are described and the performance of the algorithms is evaluated; some conclusions are presented in Sect. 5; an illustrative example of the heuristics is given in “Appendix.”

## 2 Problem formulation

In this section, we formulate the problem of calculating a maximally node and SRLG-disjoint path pair of minimal additive cost, as a multi-objective problem. Consider  $G = (V, A)$ , the directed graph representing the network, where  $V = \{v_1, \dots, v_n\}$  is the set of nodes and  $A = \{a_1, \dots, a_m\}$  is the set of arcs, such that  $\forall_{i \in \{1, 2, \dots, m\}} a_i = (v_j, v_h)$  and  $v_j, v_h \in V$ . The cost of using an arc  $a = (v_i, v_j) \in A$  in a path is given by  $l(a)$  or  $l(v_i, v_j)$  and is assumed to be nonnegative.

Let  $s$  and  $t$  ( $s, t \in V$ ) denote the source and destination nodes, respectively, of the path pair to calculate. Note that arc disjointness is ensured in [10] by SRLG diversity (assuming each arc belongs at least to one SRLG). In this work, however, it will be considered that some arcs do not belong to any SRLG (each of those arcs could have been considered as defining an SRLG with a single element, but that was not our approach). This allows to distinguish between paths sharing an edge that does not belong to any SRLG and sharing edges that belong to one or more SRLGs.

Let  $R$  be the set of risks in the network. Let  $A_r$  be the (non-empty) set of arcs affected by risk  $r \in R$ . We do not assume that each arc  $a \in A$  belongs to at least one  $A_r$  ( $r \in R$ ), so  $\cup_{r \in R} A_r$  may be only a subset of  $A$ . The set of all SRLGs in the network is  $\psi = \{A_r : r \in R\}$ . The set of SRLGs to which an arc  $a \in A$  belongs will be designated by  $\varphi_a$ .

Let  $\delta(v_i)^+$  be the set of arcs in  $A$  emergent from node  $v_i \in V$  and  $\delta(v_i)^-$  be the set of arcs in  $A$  incident on node  $v_i \in V$ . A path is a sequence of nodes (all different) from  $s$  to  $t$ , ( $s, t \in V$ ), and is represented by  $p = \langle s \equiv v_1, v_2, \dots, v_u \equiv t \rangle$ , where  $\forall_{i \in \{1, \dots, u-1\}} (v_i, v_{i+1}) \in A$ ,  $u$  being the number of nodes in the path. Let  $V_p$  be the set of nodes in the path  $p$  and  $A_p$  be the set of arcs that form the path,  $A_p = \cup_{i \in \{1, \dots, u-1\}} (v_i, v_{i+1})$ . The set of arcs symmetrical to the arcs in  $A_p$  is  $\bar{A}_p$ . The set of SRLGs which may affect a path  $p$  will be designated by  $\psi_p$ . The set of arcs affected by the SRLGs in the set  $\psi_p$  will be designated by  $A_{\psi_p}$  and is defined by  $\cup A_r : A_r \in \psi_p$ .

Let a path pair be represented by  $(p, q)$ . We wish to obtain a path pair such that it is maximally node and SRLG disjoint and has minimal cost. The cost of path pair  $(p, q)$  is given by  $c(p, q) = \sum_{a \in p} l(a) + \sum_{a \in q} l(a)$ . When there is no fully node and SRLG-disjoint path pair for a given source and destination nodes  $s$  and  $t$ , the problem becomes a multi-objective optimization problem. Given a path pair  $(p, q)$  from  $s$  to  $t$ , the objective functions (o.f.) are:

$$f_1(p, q) = |(V_p \cap V_q) \setminus \{s, t\}| \quad (1)$$

$$f_2(p, q) = |A_p \cap A_q| \quad (2)$$

$$f_3(p, q) = |\psi_p \cap \psi_q| \quad (3)$$

$$f_4(p, q) = c(p, q) \quad (4)$$

Function  $f_1(p, q)$  returns the number of intermediate common nodes of  $p$  and  $q$ ; the result is 0 if the path pair is node disjoint. Function  $f_2(p, q)$  returns the number of common arcs of  $p$  and  $q$ ; the result is 0 if the path pair is arc disjoint. Function  $f_3(p, q)$  returns the number of common SRLGs of  $p$  and  $q$ ; the result is 0 if the path pair is SRLG disjoint. Function  $f_4(p, q)$  returns the path pair cost.

Let  $\mathcal{P}$  be the feasible design space of paths pairs  $(p, q)$ , from  $s$  to  $t$ , such that  $p \neq q$ . A lexicographic approach for ranking the solutions was considered [13], where each of the optimization problems should be solved sequentially:

$$\text{Minimize}_{(p,q) \in \mathcal{P}} f_i(p, q) \quad (5)$$

subject to  $p \neq q$  and  $f_j(p, q) \leq f_j(p_j^*, q_j^*)$ ,  $j = 1, 2, \dots, i-1$  when  $i > 1$ ,  $i = 1, 2, 3, 4$ , where  $f_i$  is the  $i$ -th o.f. and  $i$  represents the position (order of importance) of a function in the preferred sequence. The path pair  $(p_j^*, q_j^*)$  represents the optimum of the  $j$ -th o.f., found in the  $j$ -th iteration. After the first iteration ( $j = 1$ ),  $f_j(p_j^*, q_j^*)$  is not necessarily the same as the independent minimum of  $f_j(p, q)$ , because of the new constraints. The vector of o.f.  $[f_1(p, q), f_2(p, q), f_3(p, q), f_4(p, q)]^T$  is a point in the criterion space.

A solution  $(p', q')$  lexicographically dominates a solution  $(p, q)$  if  $f_k(p', q') < f_k(p, q)$ , with  $k = \min_i : f_i(p, q) \neq f_i(p', q')$ .

## 3 Algorithms for maximally node and SRLG-disjoint routing

The problem given by Eq. (5) is difficult to solve, namely because minimizing the function  $f_3(p, q)$  (the number of common SRLGs in  $p$  and  $q$ ) is an NP-complete problem [10]. Two heuristics were proposed in [8] for solving a similar problem. The first one, Modified CoSE-MS (MdCoSE-MS), is a modification of CoSE-MS [9] and of Modified CoSE [19]

and the second, Modified IMSH (MdIMSH), is a modification of IMSH [24].

In this section, two new heuristics are proposed: Modified Trap Avoidance (MdTA) and Modified Weighted-SRLG path selection algorithm (MdWSRLG). The first one extends the Trap Avoidance algorithm [27] and the second one extends the Weighted-SRLG path selection algorithm (WSRLG) [15] for obtaining maximally SRLG-disjoint path pairs.

A solution to the multi-objective optimization problem formulated in the previous section is searched using these four heuristics (see “Appendix” for illustrative an example). The first solution obtained by the four heuristics is the same: The maximally node-disjoint path pair of min-sum cost between the considered end nodes is calculated [3]. If such a solution exists, it will correspond to the solution with the smallest number of nodes and arcs in common. If the path pair is SRLG disjoint, then it is an optimal solution, and the algorithms end. Otherwise a solution with the smallest number of SRLGs in common will be sought.

Two ancillary heuristics, originally proposed in [8], are reviewed in the next subsection.

### 3.1 Ancillary heuristics

The modified Bhandari’s edge-disjoint shortest path pair heuristic (MBH) and the Modified Suurballe’s Heuristic (MSH) were presented in [7] and [24], respectively. Both were extended for obtaining maximally node-disjoint path pairs in [8]. The new heuristics MdTA and MdWSRLG use extended versions of MSH and MBH (MSHE and MBHE, respectively).

Let  $G'$  be the graph where the seed path  $p'_g$  is calculated; that is,  $p'_g$  is the shortest path in  $G'$  which will be used to compute a path pair from node  $s$  to node  $t$  in the original network  $G$ . The MBHE can only be used when the seed path is calculated in the original network, that is if  $G' = G$ .

The network transformation in MBHE and MSHE, required for obtaining a maximally node and SRLG-disjoint path pair, uses an adaptation of the vertex-splitting method 2 in [3]—see Fig. 4 in “Appendix”. Given a seed path  $p'_g$  (the shortest path in  $G$  in the case of the MBHE), consider a copy of the original graph  $G^s = G$ , where the intermediate nodes of  $p'_g$  are divided in two nodes connected by two symmetrical arcs, such that for all  $v_i \in V_{p'_g} \setminus \{s, t\}$ ,  $v_i$  is replaced by two nodes  $v'_i$  and  $v''_i$ , arc  $(v''_i, v'_i)$  with null cost and arc  $(v'_i, v''_i)$  with a cost  $y$ , where  $y$  is  $\sum_{a \in A} l(a) + \varepsilon$ , where  $\varepsilon > 0$  is an arbitrary positive number. Furthermore, all the arcs in  $\delta(v_i)^+$  will now emerge from  $v''_i$  and all the arcs in  $\delta(v_i)^-$  will now be incident on  $v'_i$ ,  $\forall v_i \in V_{p'_g} \setminus \{s, t\}$ . Then, assign the cost  $l(a_i) + x$ , where  $x = (|V| - 1)y$ , to each arc of  $p'_g$ , originally of cost  $l(a_i)$ . To the symmetrical arc (which has to be created if it does not exist) assign the cost  $-l(a_i)$

in the case of MBHE and the cost 0 in the case of MSHE, for all  $a_i \in A_{p'_g}$ . Finally, all the other arcs in SRLG conflict with  $p'_g$  ( $A_{\psi_{p'_g}} \setminus (A_{p'_g} \cup \bar{A}_{p'_g})$ ) have their costs increased by  $x$ . Next the minimum cost path,  $p''_g$ , in the modified network is calculated by recurring to the modified Dijkstra’s algorithm [3]. The graph  $G^s$  is transformed back into the original graph  $G$  (the divided nodes are collapsed, and the costs are reset to their original value). After removing the possibly existing interlacing arcs (the arcs in  $\bar{A}_{p'_g} \cap A_{p''_g}$ ), a path pair  $(p_g, q_g)$  is obtained. However, if  $p'_g = p''_g$ , then  $p_g = q_g = \emptyset$ , and no maximally disjoint path pair could be obtained using  $p'_g$  as the seed path.

### 3.2 Modified Trap Avoidance (MdTA) algorithm

The heuristic Trap Avoidance was proposed for solving the min–min SRLG-disjoint path pair problem. It sequentially removes the most *risky link* [27] from the network, calculates the candidate AP in this reduced network and then tries to obtain an SRLG-disjoint BP in the original network, where the cost of the forward arcs of the AP is changed to infinity and the cost of the reverse arcs of the AP and of the arcs with SRLG in common with the AP takes a very large value. The algorithm ends as soon as an SRLG-disjoint path pair is obtained or if no candidate AP can be found in the pruned network or the number of allowed iterations was reached.

The first best current solution of MdTA is the maximally node disjoint path pair, of min-sum cost [3] between the considered end nodes, similar to the previously proposed heuristics. If the path pair is SRLG disjoint, the algorithm ends with an optimal solution. Otherwise it proceeds into the main cycle of MdTA. Note that if the first obtained path pair is not arc disjoint, the set of common arcs in that path pair is the set of critical arcs  $K_A$ , that is the set of arcs that cannot be removed, if a seed path is to be successfully calculated. This main cycle of MdTA uses MSHE. The seed path required by MSHE is calculated in the pruned network (as the candidate AP in TA). If the resulting path pair is not SRLG disjoint, the most risky arc must be identified. The most risky arc selection is done differently from [27] and is described by the following algorithm.

#### Most Risky Arc (MostRiskyArc( $K_A, p_g, \psi_x$ ))

**Require:**  $K_A$ , the set of critical arcs for the considered node pair;  $p_g$ , the seed path;  $\psi_x$  the set of common SRLGs in the path pair obtained by MSHE using  $p_g$  as seed path.

**Ensure:** Returns the most risky arc  $a_R$  ( $a_R \in p_g$ ) or 0 if all the links in  $p_g$  belong to  $K_A$

```

1:  $a_R \leftarrow 0$  ▷ Signals no risky arc was found
2:  $A_T \leftarrow A_{p_g} \setminus K_A$  ▷ Set of candidates for most risky arc
3: if  $A_T \neq \emptyset$  then
4:    $a_R \leftarrow$  element from  $A_T$ 
5: for every  $a \in A_T$  do
```



```

6:   if COUNTR( $a, \psi_x$ ) > COUNTR( $a_R, \psi_x$ ) then
7:      $a_R \leftarrow a$ 
8:   end if
9: end for
10: end if
11: return  $a_R$ 

```

where COUNTR( $a, \psi_x$ ) returns the number of SRLGs in  $\psi_x$  that contain arc  $a$ . So MOSTRISKYARC( $K_A, p_g, \psi_x$ ) in MdTA returns the arc in the seed path ( $p_g$ ), not contained in  $K_A$ , which appears more often in the SRLGs shared by the resulting path pair ( $\psi_x$ ). If the set  $\psi_x$  is empty, the algorithm returns an arbitrary arc from  $A_{p_g} \setminus K_A$ .

Let  $T$  designate the set of links of the BP which share an SRLG with the AP, and  $X$  designate the set of SRLGs of the links in set  $T$ . In TA [27] the most risky link is the link of the AP which belongs to the largest number of SRLGs in  $X$ . Using MOSTRISKYARC, set  $X$  is replaced by set  $\psi_x$  (the set of common SRLGs in the path pair obtained by the MSHE using  $p_g$  as seed path) and the path from which the most risky link is selected is the seed path  $p_g$ , which may not coincide with the candidate AP calculated by the MSHE. This approach tries to increase the possibility of obtaining a seed path for the MSHE, which will result in a maximally SRLG-disjoint path pair of min-sum cost.

Algorithm MdTA follows an approach similar to TA, but uses the new heuristic MSHE and requires the knowledge of the seed path which was calculated in the modified graph. Given the shortest path ( $p_g$  in line 9 of the algorithm) in the modified graph, and using it as the seed path in the MSHE, a path pair is calculated,  $(p, q)$ . Then, the best current solution,  $(p_b, q_b)$ , is possibly updated in line 14 of MdTA. If  $p$  and  $q$  are not SRLG disjoint, the set  $\psi_x = \psi_p \cup \psi_q$  is used by the function MOSTRISKYARC( $K_A, p_g, \psi_x$ ) to obtain the most risky arc. In MdTA,  $G^* = (V^*, A^*)$  is the pruned graph from which the seed path  $p_g$  is calculated (initially  $G^* = G$ ).

### Modified Trap Avoidance (MdTA)

**Require:** Network directed graph  $G = (V, A)$ ; node pair  $(s, t)$ ;  $l(a) \geq 0$  cost of the arc, for all  $a \in A$ ;  $\psi$ , the set of SRLGs in the network;  $i_{\max}$ , the allowed number of iterations with  $i_{\max} \geq 1$

**Ensure:** Returns a path pair from  $s$  to  $t$ , seeking to solve the problem in Eq. (5);  $(p_b, q_b) = (\emptyset, \emptyset)$  if no solution exists

```

1:  $(p_b, q_b) \leftarrow \text{MAXNODEDISJ}(s, t)$   $\triangleright (p_b, q_b) = (\emptyset, \emptyset)$  if
   no maximally disjoint path exists
2:  $i \leftarrow 1$   $\triangleright$  This was the first iteration
3: if  $(p_b, q_b) \neq (\emptyset, \emptyset) \wedge \psi_{p_b} \cap \psi_{q_b} \neq \emptyset$  then  $\triangleright$  First
   solution exists and is not SRLG-disjoint
4:    $K_A \leftarrow A_{p_b} \cap A_{q_b}$   $\triangleright$  Critical arcs
5:    $stop \leftarrow false$   $\triangleright$  There is at least one seed path
6:    $G^* \leftarrow G$   $\triangleright V^* = V, A^* = A$ 
7:   while  $\neg \text{EMPTY}(A^*) \wedge \neg stop \wedge i < i_{\max}$  do
8:      $i \leftarrow i + 1$   $\triangleright$  Current iteration

```

```

9:    $p_g \leftarrow$  shortest path in  $G^*$ 
10:  if  $p_g = \emptyset$  then  $\triangleright p_g$  does not exist
11:     $stop \leftarrow true$   $\triangleright$  No more seed paths
12:  else  $\triangleright p_g$  exists
13:     $(p, q) \leftarrow \text{MSHE}(G, \psi, l, p_g)$ 
14:     $(p_b, q_b) \leftarrow \text{MAYUPDATESOL}(p_b, q_b, p, q)$ 
15:     $\psi_x \leftarrow (\psi_p \cap \psi_q) \setminus K$ 
16:     $a_R \leftarrow \text{MOSTRISKYARC}(K_A, p_g, \psi_x)$ 
17:     $A^* \leftarrow A^* \setminus \{a_R\}$ 
18:  end if
19: end while
20: end if
21: return  $(p_b, q_b)$ 

```

Algorithms TA and MdTA target different problems. The first addresses the min–min SRLG-disjoint path pair problem, and the second addresses the maximally node and SRLG-disjoint path pair of minimal additive cost, formulated as a multi-objective problem. MdTA borrowed from TA the idea of successively pruning the most risky link from the network. However, the calculation of the path pair is different from TA, because instead of calculating the BP in a network where the directed arcs of the AP have their cost changed to infinity and every arc in SRLG conflict with the AP is changed to a large cost, MdTA uses MSHE for increasing the possibility finding a path pair of max-sum cost. Additionally, the selection of the most risky arc (as explained in the previous paragraph) is also adapted, taking into account the use of the MSHE in MdTA.

### 3.3 Modified Weighted-SRLG (MdWSRLG) algorithm

The Weighted-SRLG (WSRLG) searches for  $K$ -shortest SRLG-disjoint paths based on an approach that treats the number of SRLGs that affect a link as part of the link cost [15]. In [15], the authors claim that although they do not seek to optimize the total cost of the obtained set, the obtained set contains cost-effective disjoint paths. When  $K = 2$ , we have a disjoint path pair.

The simple iterative approach for obtaining a set of  $K$  SRLG disjoint paths consists in finding a shortest path, then pruning its links and nodes, as well as links that belong to SRLGs that affect its links and then calculating the shortest path in this pruned network; this procedure is repeated in each iteration, after a new shortest path is calculated, successively pruning the network, until the intended number of paths is found or no more disjoint paths can be found. The authors in [15] call this the  $K(s, t)$  algorithm, where  $K$  is the number of paths to be obtained. This approach often falls into a trap and may fail to find the required number of disjoint paths, especially when a link which belongs to the shortest paths shares several risks with other links.

The weighted-SRLG approach aims to improve the effectiveness of the simple iterative approach by modifying the cost of all the network links, in each iteration, before the  $K(s, t)$  algorithm is executed. In each iteration, the cost of the network links is given by Eq. (6). A binary search method is used for the value of  $\alpha$ . The value of  $\alpha$  increases as the number of iterations increases, whenever the target value of  $K$  disjoint paths is not obtained; if the obtained value is  $K$ , then  $\alpha$  is decreased. Observing Eq. (6), as  $\alpha$  decreases the contribution of the original cost of a link to its new cost increases, while the contribution of the number of SRLGs that affect the link decreases.

$$l_w(a) = \frac{1 - \alpha}{l_{\max}} l(a) + \frac{\alpha}{\bar{\varphi}_{\max}} \max\{\bar{\varphi}_a, 1\} \quad (6)$$

where:

- $l_w(a)$  is the cost of link  $a$  in MdWSRLG.
- $\alpha$  is a parameter defined by the binary search method.
- $l_{\max}$  is  $\max_a l(a)$ , the largest of the original costs of the network arcs.
- $\bar{\varphi}_a$  is  $\sum_{A_r \in \varphi_a} |A_r|$ , the sum of the size of the SRLGs arc  $a$  belongs to.
- $\bar{\varphi}_{\max}$  is  $\max_a \bar{\varphi}_a$ .
- $\max\{\bar{\varphi}_a, 1\}$  is the maximum between  $\bar{\varphi}_a$  and 1.

The modified version of the WSRLG algorithm (MdWSRLG) allows to search for a maximally node and SRLG-disjoint path pair. The initial solution to the MdWSRLG algorithm is the maximally node-disjoint path pair of minimum cost, obtained in the original network. If the initial path pair exists and is SRLG disjoint, that pair is the best possible solution and so the algorithm ends. Otherwise, while the convergence condition for the value of  $\alpha$  is not met, the following steps are executed: The value of  $\alpha$  is updated; the cost of all links in the network is modified according to Eq. (6); the shortest path in this network is calculated; the MBHE is used to calculate a maximally node and SRLG disjoint path pair; then, a possible update of the current best solution is done according to Eq. (5); finally, the value of  $\alpha_{\max}$  is updated.

#### Modified Weighted-SRLG (MdWSRLG)

**Require:** Network directed graph  $G = (V, A)$ ; node pair  $(s, t)$ ;  $l(a) \geq 0$  cost of the arc, for all  $a \in A$ ;  $\psi$  set of SRLGs in the network;  $\epsilon$  convergence parameter; initial value of  $\alpha_{\min}$  and  $\alpha_{\max}$

**Ensure:** Returns a path pair from  $s$  to  $t$ , seeking to solve the problem in Eq. (5);  $(p_b, q_b) = (\emptyset, \emptyset)$  if no solution exists

- 1:  $(p_b, q_b) \leftarrow \text{MAXNODEDISJ}(s, t) \triangleright (p_b, q_b) = (\emptyset, \emptyset)$  if no maximally disjoint path exists
- 2:  $i \leftarrow 1 \triangleright$  This was the first iteration
- 3: **if**  $(p_b, q_b) \neq (\emptyset, \emptyset) \wedge \psi_{p_b} \cap \psi_{q_b} \neq \emptyset$  **then**  $\triangleright$  First solution exists and is not SRLG-disjoint

- 4: **while**  $\alpha_{\max} - \alpha_{\min} > \epsilon$  **do**
- 5:    $\alpha \leftarrow (\alpha_{\min} + \alpha_{\max})/2$
- 6:   Re-calculate  $l_w(a)$ , for all  $a \in A$ , using Eq. (6)
- 7:    $p_s \leftarrow$  Shortest path from  $s$  to  $t$  using costs  $l_w$
- 8:    $(p, q) \leftarrow \text{MBHE}(G, \psi, l_w, p_s)$
- 9:    $(p_b, q_b) \leftarrow \text{MayUpdateSol}(p_b, q_b, p, q)$
- 10:    $\alpha_{\max} \leftarrow \alpha$
- 11: **end while**
- 12: **end if**
- 13: **return**  $(p_b, q_b)$

One must recall that algorithms WSRLG and MdWSRLG have different objectives. The first one seeks to obtain a set of  $K$  SRLG-disjoint paths, while MdWSRLG seeks to calculate a maximally node and SRLG-disjoint path pair of minimal additive cost. Hence, instead of just pruning from the network the arcs in SRLG conflict with the seed path (the approach used in WSRLG), MdWSRLG uses MSHE in order to make it more likely to obtain a maximally disjoint solution and also a solution of lower cost.

## 4 Results analysis

To analyze the proposed heuristics performance, tests were run in four networks from [16]. For each network, edge costs were taken as the integer component of the first of the module costs present in the cited networks.

For each network, SRLGs in number equal to half of the number of edges in that network were generated. For each generated SRLG, the target number of edges was randomly (uniformly) selected from the set  $\{2, 3, 4\}$ . Each edge was assigned a target number of SRLGs uniformly selected in the set  $\{0, 1, 2, 3, 4\}$ . Each SRLG was defined through random (uniform) selection of a node, and selecting the candidate edges for that SRLG as those with an end node at a distance less than  $d$  from the selected node. For each SRLG, each suitable edge (one that still has not exceeded its assigned SRLG target) is selected for inclusion to the present SRLG with a probability of 10 %. The value of  $d$  was defined for each network taking into account the average and minimum length of all the node pairs in that network (with each length computed using the straight-line distance between GPS coordinates of the nodes). In this way, while SRLG are not strictly local, the edges in each SRLG are located in a  $d$ -radius neighborhood. Considering that edges in close geographical proximity should share common risk faults, the procedure used to generate the SRLG should probably result in a realistic SRLG distribution.

Table 1 presents the networks used in the tests performed, where  $n$  is the number of nodes,  $m$  the number of edges and  $\delta$  the average node degree. For each network, the total number of pairs considered was therefore  $n \times (n - 1)$ .

**Table 1** Network data and exact algorithm results (networks from [16])

| Network   | $n$ | $m$ | $\delta$ | CPU time (s)                          | Fully disjoint solutions (%) | Avg. number of common SRLGs |
|-----------|-----|-----|----------|---------------------------------------|------------------------------|-----------------------------|
| nobel-eu  | 28  | 41  | 2.9      | $5.06\text{E-}02 \pm 1.75\text{E-}03$ | $65.19 \pm 3.934$            | $1.4 \pm 0.15$              |
| cost266   | 37  | 57  | 3.1      | $6.65\text{E-}02 \pm 4.05\text{E-}03$ | $82.48 \pm 5.638$            | $1.2 \pm 0.12$              |
| germany50 | 50  | 88  | 3.5      | $7.26\text{E-}02 \pm 2.54\text{E-}03$ | $94.75 \pm 3.156$            | $1.0 \pm 0.01$              |
| ta2       | 65  | 108 | 3.3      | $8.92\text{E-}02 \pm 2.66\text{E-}03$ | $79.84 \pm 1.668$            | $1.0 \pm 0.01$              |

Ignoring the SRLGs, networks nobel-eu, cost266 and germany50 are biconnected; that is, they would remain connected if any single node was removed. Network ta2 is connected, but not biconnected. Originally, all those networks were undirected, but in the algorithms each undirected edge was replaced by a pair of symmetrical arcs (directed edges), with opposite directions, both with cost equal to the cost of the replaced undirected edge; if the undirected edge belonged to a given SRLG then the corresponding symmetrical arcs will also belong to that same SRLG..

The CPU time per node pair was calculated from the total CPU time for solving for all node pairs in a Dell Precision 7500, Intel(R) Xeon(R) CPU X5660 (Six Core, 2.80 GHz, 6.4 GT/s, 12 MB), with 48 GB of RAM.

For each of the four networks in Table 1, ten different random seeds were used to generate ten different SRLG distributions ( $\psi$ ). Those ten networks with the same topology but different  $\psi$  were then used in the tests with each heuristic, and their results averaged. Each heuristic was executed with a maximum number of iterations ( $i_{max} \in \{5, 10, 25, 50, 100, 250, 500, 1000\}$ ), but in the following figures and tables, results will be presented only for ( $i_{max} \in \{5, 10, 50, 100, 500, 1000\}$ ). Unless otherwise stated, results are presented considering all node pairs for all networks and with 95 % confidence intervals (CI). The results in Table 1 were obtained using a linear integer programming formulation (henceforth called the Exact Algorithm), solved using CPLEX version 12.5 [1].

#### 4.1 Performance evaluation of the four heuristics

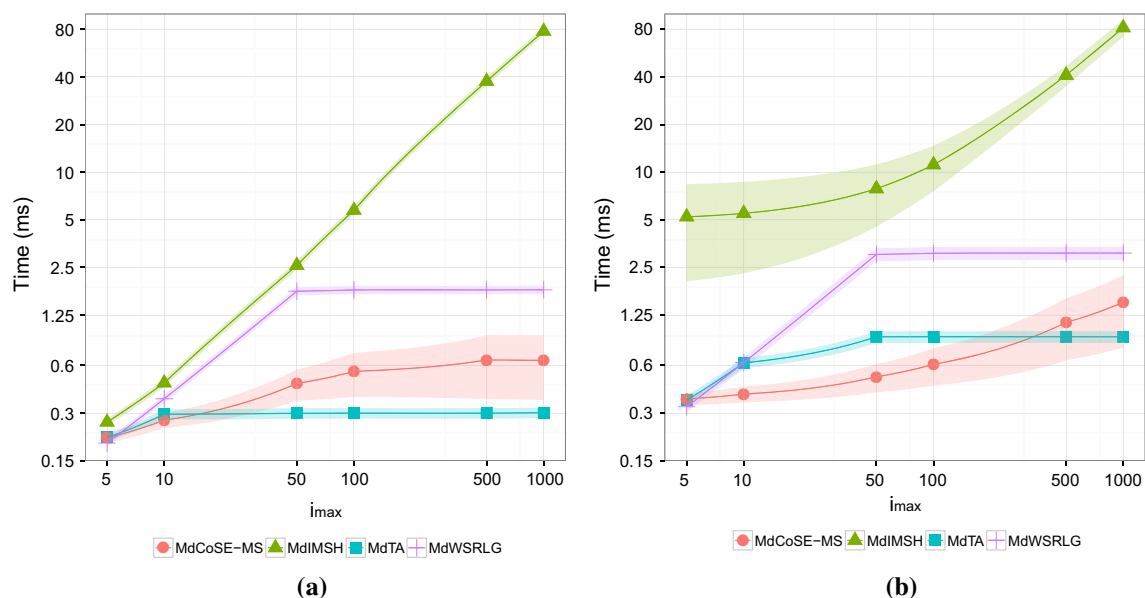
Using the exact algorithm, Table 1 presents average CPU time spent per node pair in seconds, percentage of fully disjoint solutions and average number of common SRLG for non-SRLG disjoint path pairs. In this table, we can see that the times increase with network dimension and range between 50ms to about 90ms per node pair.

The average CPU times per pair for each heuristic are presented only for the nobel-eu network (in Fig. 1a) and the ta2 network (in Fig. 1b), because the pattern is similar in the other two networks. In all figures, the width of the CI is represented by a shadow around the average value. Since in Fig. 1 the y axis uses logarithms, the upper and lower CI appear of different size, although they are identical.

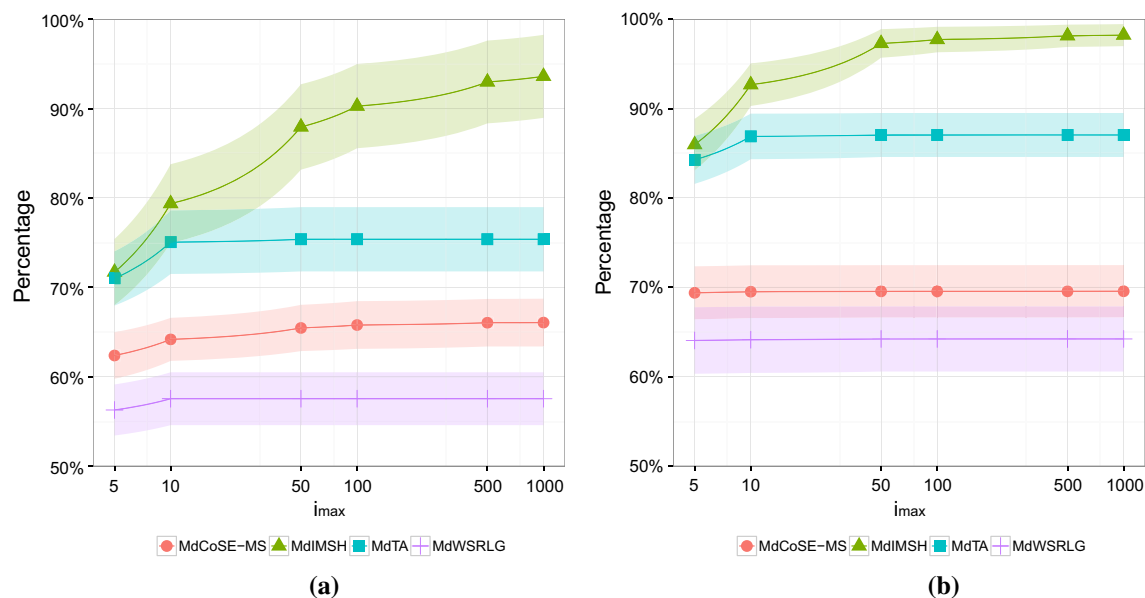
In the considered networks, the time per node pair usually increases for all heuristics with the maximum number of iterations allowed ( $i_{max}$ ), but at different rates depending on the heuristic, network and  $i_{max}$  value. For smaller values of  $i_{max}$  ( $i_{max} \leq 500$ ), all heuristics are faster than the exact solution, but when more iterations are allowed ( $i_{max} > 500$ ) the MdIMSH heuristic can on average take longer per node pair than the exact solution. For all networks, the fastest heuristics are MdTA and MdCoSE-MS, followed by MdWSRLG, with MdIMSH as the slowest one, for larger values of  $i_{max}$ . The values per pair range are below 10ms for all heuristics with  $i_{max} \leq 50$ , but can go up to 85ms per node pair for MdIMSH with  $i_{max} = 1000$ . The fact that for some heuristics the average time per node pair does not increase with  $i_{max}$  suggests that for those heuristics the bound on the number of maximum iterations is not active (i.e., that those heuristics stop iterating before reaching  $i_{max}$ ).

To examine the quality of the heuristic results, we checked the number of path pairs in the network for which the heuristic reached the exact (optimal) solution (or an alternative optimum, i.e., solutions with the same values for arc disjunction, node disjunction, SRLG disjunction and the cost of the pair of paths of the exact solution). Notice that the optimal solution was not always fully disjoint, because of common SRLG(s) and/or, in the case of ta2 network, topological reasons.

In Fig. 2a (cost266 network) and Fig. 2b (germany50 network), we can find the percentage of node pairs where the heuristic reaches a solution equivalent to the optimal. Although small variations in actual percentage exist per network, it can be seen that the MdIMSH reaches the optimal more often, followed by the MdTA heuristic—which is also true for the nobel-eu and ta2 networks (not shown). Both MdCoSE-MS and MdWSRLG reach the optimum much less often, and particularly increasing  $i_{max}$  almost does not increase the accuracy of those heuristics, unlike MdIMSH. Roughly, the worst performing heuristics, MdWSRLG and MdCoSE-MS, reach at most 68 and 73 % of the optimal solutions, respectively, while MdIMSH can reach up to more than 90 % of the optimal solutions in the tested networks (reaching over 98 % of the optimal solutions on the germany50 network). MdTA reaches 5–15 % more optimal solutions than MdCoSE-MS in the case of the nobel-eu network; on the other three networks, MdTA gets 9–17 % more optimal solu-



**Fig. 1** Average CPU time per node pair (in milliseconds). **a** nobel-eu network. **b** ta2 network



**Fig. 2** Percentage of pairs with the optimal solutions. **a** cost266 network. **b** germany50 network

tions than MdCoSE-MS, for all values of  $i_{max}$ ; MdIMSH reaches about 10 % more optimal solutions than MdTA for  $i_{max} \geq 50$ .

#### 4.2 Further analysis between MdIMSH and MdTA

Since MdIMSH and MdTA provided the best results regarding the number of optimal solutions found, they were chosen for additional study. Both heuristics were compared for lexicographic dominance (according to the criteria presented in Sect. 2). This is illustrated in Table 2, which shows the percentage of node pairs where dominance is found, for cost266

and ta2 networks. It can be seen that for  $i_{max} > 5$ , MdIMSH dominates MdTA in more pairs than it is dominated. However, it can also be seen that in the majority of pairs the heuristics return the same or incomparable solutions.

Another relevant aspect is the number of fully disjoint solutions. Table 1 shows the total average percentage of fully disjoint solutions found in each network (computed using the exact algorithm). It can be seen that the smaller percentage of fully disjoint solutions can be found in the nobel-eu network (with only 65 % fully disjoint pairs), whereas the largest occurs in the germany50 network (where almost 95 % of pairs have fully disjoint solutions). Regarding the heuris-



**Table 2** Average dominance relation between heuristics

| $i_{max}$ | cost266                   |                           | ta2                       |                           |
|-----------|---------------------------|---------------------------|---------------------------|---------------------------|
|           | MdIMSH dominates MdTA (%) | MdTA dominates MdIMSH (%) | MdIMSH dominates MdTA (%) | MdTA dominates MdIMSH (%) |
| 5         | 8.8 ± 1.5                 | 9.1 ± 1.0                 | 7.8 ± 1.4                 | 10.4 ± 1.5                |
| 10        | 12.3 ± 1.8                | 7.1 ± 0.8                 | 8.8 ± 1.5                 | 9.8 ± 1.6                 |
| 50        | 18.5 ± 1.7                | 1.8 ± 0.4                 | 13.2 ± 1.9                | 4.0 ± 0.6                 |
| 100       | 19.9 ± 1.7                | 0.8 ± 0.3                 | 14.7 ± 2.3                | 2.2 ± 0.4                 |
| 500       | 21.3 ± 2.1                | 0.1 ± 0.2                 | 16.6 ± 2.9                | 0.4 ± 0.2                 |
| 1000      | 21.6 ± 2.2                | 0.0 ± 0.1                 | 17.1 ± 3.1                | 0.2 ± 0.1                 |

tics, MdIMSH usually reaches a percentage of fully disjoint solutions very similar to the exact results, for  $i_{max} > 50$ , while MdTA is usually slightly worse (about 2–5 % for larger values of  $i_{max}$ ).

The AP in MdTA is calculated in a successively pruned network, and hence, MdTA does not perform many iterations. In fact, for  $i_{max} > 50$ , the average number of iterations made by MdTA is constant on average and less than 8 for the considered networks. In contrast, MdIMSH performs more iterations (about 60–75 % of the allowed value for  $i_{max}$ ). The increasing number of iterations of MdIMSH with  $i_{max}$  helps explain both its better results and increased time expenditure when compared to MdTA.

Table 1 shows the average number of SRLGs in common for path pairs which are not SRLG disjoint, obtained using the exact algorithm. It can be seen that the average number of SRLGs in common decreases with network size, reaching almost one unit value for the larger networks. It was observed that as expected both heuristics have on average more SRLGs in common than an optimal solution. For  $i_{max} > 50$ , however, MdIMSH finds non-disjoint paths with nearly as few SRLG in common as the exact solution and MdTA has consistently worse performance than MdIMSH.

The performance of the algorithms was also evaluated using a PCE, model UNICOM-V5, G2\_LE CPU (PowerPC compatible core) with 330MHz core clock, 128 MB of RAM. The average CPU time per node pair was verified to be below 160 and 16ms for the MdIMSH and MdTA algorithms, respectively, for the ta2 network when the number of iterations was 50 (for the other networks and same number of iterations, the CPU times were smaller). Hence, MdTA can be used in the control plane of a GMPLS network using PCEs, while MdIMSH is more adequate for the management plane.

## 5 Conclusion

Two heuristics, MdWSRLG and MdTA, were proposed for solving the min-sum maximally node and SRLG-disjoint path pair. This problem was also formulated as a multi-objective optimization problem. The relative performance of

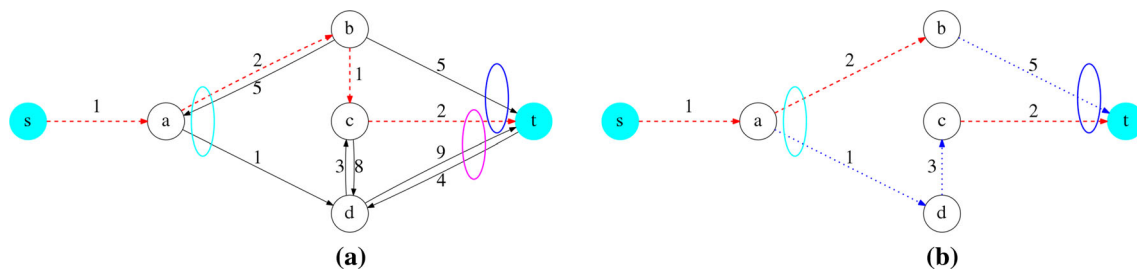
these heuristics and of two other (MdIMSH and MdCoSE-MS [8]) was evaluated using four networks from [16], where SRLGs were generated, taking into account the edges geographical proximity. MdIMSH and MdTA provided the best quality results regarding the number of exact solutions found and were further analyzed. It was shown that MdIMSH's solutions dominated a significant number of the solutions obtained by MdTA. MdIMSH also had better performance than MdTA regarding the average number of node and SRLG-disjoint path pairs, and the average number of common SRLGs for path pairs which are not SRLG disjoint. However, the number of iterations used by MdTA is on average smaller than MdIMSH, resulting in lower CPU time. Hence, it appears that MdTA can be a good compromise for using the GMPLS control plane and MdIMSH for the GMPLS management plane.

**Acknowledgments** We thank Sérgio Mendes for implementing part of the code used in this work. The authors acknowledge financial support through project QREN 23301 PANORAMA II, co-financed by European Union's FEDER through "Programa Operacional Factores de Competitividade" (POFC) of QREN (FCOMP-01-0202-FEDER-023301), by the Portuguese Foundation for Science and Technology under project grant UID/MULTI/00308/2013 and by PT Inovação R&D Project "End to End Protection considering SRLGs-II".

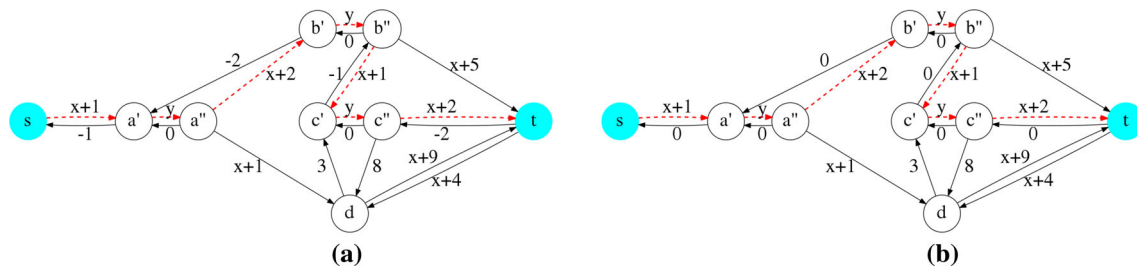
## Appendix: illustrative example

This appendix will try to clarify the main differences between the considered heuristics. The network which will be used in this example is in Fig. 3a, where the ellipses represent different SRLGs and the arcs in dashed red represent the shortest path from  $s$  to  $t$ . Let the SRLGs be  $A_1 = \{(a, b), (a, d), (b, a)\}$ ,  $A_2 = \{(b, t), (c, t)\}$  and  $A_3 = \{(c, t), (d, t), (t, d)\}$ .

The initial path pair obtained by the four considered heuristics is the same: The maximally node-disjoint path pair of min-sum cost is shown in Fig. 3b where the arcs in dashed red are from the shortest path; also note that arc  $(s, a)$  is shared by both paths. This initial solution has a minimal cost of 15 (corresponding to  $8+7$ ), where the cost of arc  $(s, a)$  is counted in both paths, and this path pair has two SRLGs in



**Fig. 3** **a** Network with SRLGs as ellipses and shortest path from  $s$  to  $t$  in dashed red. **b** Subgraph representing the maximally node-disjoint path pair of min-sum cost from  $s$  to  $t$  (Color figure online)



**Fig. 4** Transformed network using as seed the shortest path: **a** in MBHE, used by MdCoSE-MS and MdWSRLG. **b** in MdCoSE-MS, IMSH and MdTA

common. This is the initial solution, and the algorithms will seek to improve.

After calculating the seed path  $\langle s, a, b, c, t \rangle$ , the network is modified, as illustrated in Fig. 4a, b for MBHE and MSHE, respectively.

The maximally node-disjoint path pair of min-sum cost is not SRLG-disjoint, and hence, algorithm IMSH will sequentially generate paths until  $i_{\max}$  is reached (or no more paths exists), by non-decreasing cost, using each of them as a seed path in the MSHE, seeking to obtain a maximally node and SRLG-disjoint path pair. In this example, the used  $k$ -shortest path enumeration algorithm will generate five paths, and the best solution (path pair  $(\langle s, a, b, t \rangle, \langle s, a, d, t \rangle)$  of cost 19) will be obtained using the third shortest path  $\langle s, a, b, t \rangle$  as seed path.

MdWSRLG will always use MBHE in each iteration, because the seed path is always the shortest path (in the network where the costs have been modified)—note that MBHE tends to obtain path pairs with smaller cost than MSHE. MdWSRLG will change the cost of the edges according to Eq. (6), increasing the importance of the original link cost. In the case of the network in the example, the optimal solution is found with  $\alpha = 0.5$ , but the algorithm will keep trying to improve this solution, until the convergence condition for  $\alpha$  is met.

Algorithm MdCoSE-MS starts by using MBHE with seed path  $\langle s, a, b, c, t \rangle$ , obtaining the maximally disjoint path pair of Fig. 3b. The set of common SRLGs is  $\{A_1, A_2\}$ . The conflicting SRLG set will be  $\{A_2\}$ , because  $\{A_1\}$  is the critical

SRLG set (if the links in  $A_1$  are removed, no seed path can be found). The algorithm will then remove the arcs in  $A_2$  from the network before calculating a new seed path—the goal is to obtain a seed path that does not use the SRLG(s) that was (were) common to the previously obtained path pair. These arcs are re-inserted in the network before calling MSHE with seed path  $\langle s, a, d, t \rangle$ , resulting in the optimal path pair. As the set of common SRLGs of that path pair is  $\{A_1\}$  (the critical SRLG set), the algorithm ends.

Algorithm MdTA first identifies  $\{(s, a)\}$  the critical arc set (see line 4 of algorithm MdTA). MSHE, with seed path  $\langle s, a, b, c, t \rangle$ , will obtain the path pair in Fig. 3b. Then, the most risky arc (see function MOSTRISKYARC), that is the arc in the seed path considered responsible for MSHE not being able to find a node and SRLG-disjoint path (excluding arc  $(s, a)$ ), will be selected. In this example, the candidates for most risky arc are  $(c, t)$  and  $(a, b)$ , where  $(c, t)$  is selected because in the case of a tie the arc with the tail node with highest out-degree is chosen. This tends to increase the probability of being able to calculate seed paths in the successively pruned network. Removing arc  $(c, t)$  will result in the seed path  $\langle s, a, b, t \rangle$ ; after reinserting  $(c, t)$  MSHE with seed path  $\langle s, a, b, t \rangle$  will obtain the path pair  $(\langle s, a, b, t \rangle, \langle s, a, d, t \rangle)$ , which is the optimal solution. However, because it is not an SRLG-disjoint path pair, the algorithm will try to improve this solution. The next most risky arc will be  $(a, b)$ , followed by  $(a, d)$ , and then the algorithm will end with the best-found solution, because it will no longer be able to calculate a seed path.

## References

- [1] IBM ILOG CPLEX Optimization Studio V12.5. IBM (2012)
- [2] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., Swallow, G.: RSVP-TE: extensions to RSVP for LSP tunnels. IETF RFC 3209 (2001)
- [3] Bhandari, R.: Survivable Networks, Algorithms for Diverse Routing. Kluwer Academic Publishers, Norwell, Massachusetts (1999)
- [4] Cheng, X., Shao, X., Wang, Y.: Multiple link failure recovery in survivable optical networks. *Photonic Netw. Commun.* **14**, 159–164 (2007)
- [5] Farrel, A., Vasseur, J.P., Ash, J.: A path computation element (PCE)-based architecture. IETF RFC 4655 (2006)
- [6] Gao, C., Zhu, Y., Jue, J.: SRLG-aware topology aggregation for survivable multi-domain optical networks. In: 2011 IEEE International Conference on Communications (ICC), pp. 1–5 (2011). doi:[10.1109/icc.2011.5963224](https://doi.org/10.1109/icc.2011.5963224)
- [7] Gomes, T., Fernandes, L.: Obtaining a SRLG-disjoint path pair of min-sum cost. In: Rak, J., Tipper, D., Walkowiak, K. (eds.) RNDM 2010–2nd International Workshop on Reliable Networks Design and Modeling, Colocated with ICUMT 2010, pp. 116–122. ISBN: 978-1-4244-7283-3. Moscow (2010)
- [8] Gomes, T., Jorge, L., Melo, P., Girão-Silva, R., Mendes, S.: Calculating a maximally node and SRLG-disjoint path pair of min-sum cost in GMPLS networks. In: 9th International Conference on Design of Reliable Communication Networks—DRCN 2013, pp. 314–321. Budapest, Hungary (2013)
- [9] Gomes, T., Simões, C., Fernandes, L.: Resilient routing in optical networks using SRLG-disjoint path pairs of min-sum cost. *Telecommun. Syst. J.*, 1–13 (2011). doi:[10.1007/s11235-011-9574-5](https://doi.org/10.1007/s11235-011-9574-5)
- [10] Hu, J.Q.: Diverse routing in optical mesh networks. *IEEE Trans. Commun.* **51**(3), 489–494 (2003)
- [11] Luo, H., Li, L., Yu, H.: Insights for segment protection in survivable WDM mesh networks with SRLG constraints. *Photonic Netw. Commun.* **14**, 361–368 (2007)
- [12] Mannie, E.: Generalized multi-protocol label switching (GMPLS) architecture, IETF RFC 3945 (2004)
- [13] Marler, R., Arora, J.: Survey of multi-objective optimization methods for engineering. *Struct. Multidiscip. Optim.* **26**(6), 369–395 (2004). doi:[10.1007/s00158-003-0368-6](https://doi.org/10.1007/s00158-003-0368-6)
- [14] Mendonça, J., Naser, H.: Class of service with partial protection in multilayered networks with SRLG constraints. In: Proceedings of the IEEE Symposium on Computers and Communications (ISCC'10), pp. 279–284. Riccione, Italy (2010)
- [15] Oki, E., Matsuura, N., Shiimoto, K., Yamanaka, N.: A disjoint path selection scheme with shared risk link groups in GMPLS networks. *IEEE Commun. Lett.* **6**(9), 406–408 (2002)
- [16] Orlowski, S., Pióro, M., Tomaszewski, A., Wessäly, R.: SNDlib 1.0-survivable network design library. *Networks* **55**(3), 276–286 (2010). doi:[10.1002/net.20371](https://doi.org/10.1002/net.20371)
- [17] Palmieri, F.: GMPLS control plane services in the next-generation optical Internet. *Internet Protoc. J.* **11**(3), 2–18 (2008)
- [18] Pan, X., Xiao, G.: Heuristics for diverse routing in wavelength-routed networks with shared risk link groups. *Photonic Netw. Commun.* **11**, 29–38 (2006)
- [19] Rostami, M.J., Khorsandi, S., Khodaparast, A.A.: CoSE: A SRLG-disjoint routing algorithm. In: Proceedings of the Fourth European Conference on Universal Multiservice Networks (ECUMN'07). Toulouse, France (2007)
- [20] Rostami, M.J., Zarandi, A.A.E., Hoseininasab, S.M.: MSDP with ACO: a maximal SRLG disjoint routing algorithm based on ant colony optimization. *J. Netw. Comput. Appl.* **35**, 394–402 (2012)
- [21] Shao, X., Xiao, G., Zhou, L., Cheng, X., Wang, Y.: Hybrid protection in WDM networks with shared risk link groups. *Photonic Netw. Commun.* **12**, 295–307 (2006)
- [22] Sheela, D., Chellamuthu, C.: A cost effective approach for WDM network protection under critical duct constraints. *WSEAS Trans. Commun.* **11**(3), 103–112 (2012)
- [23] Silva, J., Gomes, T.M., Fernandes, L., Simões, C., Craveirinha, J.: An heuristic for maximally SRLG-disjoint path pairs calculation. In: 3rd International Congress on Ultra Modern Telecommunications and Control Systems (ICUMT 2011 Budapest). Budapest, Hungary (2011)
- [24] Todimala, A., Ramamurthy, B.: IMSH: An iterative heuristic for SRLG diverse routing in WDM mesh networks. In: 13th International Conference on Computer Communications and Networks, ICCCN'2004, pp. 199–204 (2004). doi:[10.1109/ICCCN.2004.1401627](https://doi.org/10.1109/ICCCN.2004.1401627)
- [25] Wang, S., Li, L.: Impairment aware optimal diverse routing for survivable optical networks. *Photonic Netw. Commun.* **13**, 139–154 (2007)
- [26] Xu, D., Li, G., Ramamurthy, B., Chiu, A., Wang, D., Doverspike, R.: On provisioning diverse circuits in heterogeneous multi-layer optical networks. *Comput. Commun.* **36**(6), 689–697 (2013). doi:[10.1016/j.comcom.2012.08.014](https://doi.org/10.1016/j.comcom.2012.08.014). Reliable Network-based Services
- [27] Xu, D., Xiong, Y., Qiao, C.: Novel algorithms for shared segment protection. *IEEE J. Sel. Areas Commun.* **21**(8), 1320–1331 (2003)
- [28] Xu, D., Xiong, Y., Qiao, C., Li, G.: Trap avoidance and protection schemes in networks with shared risk link groups. *J. Lightw. Technol.* **21**(11), 2683–2693 (2003)
- [29] Yan, W., Junhui, Z., Jiazhi, Z.: New SRLG-diverse path selection algorithm in survivable GMPLS networks. *J. Syst. Eng. Electron.* **20**(2), 412–419 (2009)
- [30] Zhang, F., Li, D., de Dios, O.G., Margaria, C.: RSVP-TE extensions for collecting SRLG information. IETF Draft (2014)



**Teresa Gomes** is Assistant Professor in telecommunications at the Department of Electrical Engineering and Computers of the Faculty of Sciences and Technology of the University of Coimbra, Portugal, since 1998, with tenure since 2003, and a researcher at the INESC Coimbra, a nonprofit R&D institute. She obtained a M.Sc. in computer science (1989) and Ph.D. in electrical engineering-telecommunications and electronics (1998),

all at the University of Coimbra. She is the author/co-author of more than 50 technical publications in international journals and conference proceedings and of one European Patent. Her main present interests are routing, protection and reliability analysis models and algorithms for optical, MPLS and GMPLS networks.



**Luísa Jorge** received Ph.D. degree in Electrical Engineering (telecommunications and electronics) after an M.Sc. degree in Systems and Automation and a B.Sc. degree in Computer Engineering, all from the University of Coimbra. She has published several papers in telecommunication networks. She is a teacher at the Polytechnic Institute of Bragança and a researcher at INESC Coimbra.

Her main research interests at the moment are MPLS/GMPLS, QoS analysis in multi-service telecommunication networks, networking simulation and teletraffic engineering.



**Paulo Melo** is a teacher at the Faculty of Economics of the University of Coimbra and a researcher at INESC Coimbra. He has got a PhD degree in Management, following a M.Sc. degree in Systems and Automation and a B.Sc. degree in Computer Science. He has published several dozen papers in peer-reviewed publications and has been in the organizational and scientific committees in various international meetings, in areas com-

prising decision science, services science, group decision support, information systems research and user experience evaluation. He has been lead researcher, co-lead or participant in several research and development projects (funded by grants provided either by national or European funding institutions or by industry partners).



**Rita Girão-Silva** graduated in Electrical Engineering (Telecommunications) in 1999 and received her Ph.D. diploma in Electrical Engineering (Telecommunications and Electronics) in 2009, both at the University of Coimbra. She is an Assistant Professor at the Department of Electrical and Computer Engineering, University of Coimbra, and a researcher at INESC Coimbra. Her research

areas include routing models in telecommunications networks and multi-objective optimization.