

Resilient routing in optical networks using SRLG-disjoint path pairs of min-sum cost

Teresa Gomes · Carlos Simões · Luís Fernandes

Received: date / Accepted: date

Abstract Ensuring network survivability is of the utmost importance in today's networks. A Shared Risk Link Group (SRLG) is the set of links in the network that share a common physical resource subject to fault(s). This concept allows an upper layer the ability to implement SRLG diverse routing.

Two algorithms, the Conflicting SRLG Exclusion (CoSE) and the Iterative Modified Suurballe's Heuristic (IMSH), will be shortly reviewed. The first solves the min-min problem and the second the min-sum problem, considering SRLG-disjoint paths. It will then be described a new version of CoSE, which will be designated as CoSE-MS, for solving the min-sum problem for SRLG diverse routing. Finally CoSE-MS and IMSH performance will be compared using random networks and an optical network. Results show that CoSE-

MS is a good compromise between the quality of obtained solutions and the used CPU time.

Keywords routing · SRLG-disjoint · paths with minimal cost sum · optical networks

1 Introduction and Motivation

Internet traffic growth is a result of the number of services and businesses depending on telecommunication networks. For this reason, telecommunication services availability and dependability are of the utmost importance. Service level agreements must be satisfied in the event of faults in network elements. The impact of a single (unprotected) fiber cut can be tremendous if we recall that a single fiber with Dense Wavelength Division Multiplexing (DWDM) can carry up to 100 terabits per second. To ensure network survivability, most network technologies consider some form of network recovery.

There are two basic recovery mechanisms: rerouting and protection switching. On the first case, recovery by rerouting is defined as establishing new paths or path segments on demand for restoring traffic after the occurrence of a fault. On

T. Gomes · L. Fernandes
Department of Electrical and Computer Engineering
University of Coimbra. INESC Coimbra. Portugal.
Tel.: +351-239-796261
Fax: +351-239-796247
E-mail: teresa@deec.uc.pt

L. Fernandes
E-mail: a2003107513@alunos.deec.uc.pt

C. Simões
Polytechnic Institute of Viseu. INESC Coimbra. Portugal
E-mail: csimoes@ipv.pt

the second case, protection switching recovery mechanisms pre-establish a recovery path or path segment. From a topological point of view the recovery may be local, segment or global. Local recovery protects against a link or neighbor node fault of a working path and seeks to minimize the amount of time required for failure propagation. Segment recovery protects against a failure in a path segment. The intent of global repair is to protect against any link or node fault on a path or on a segment of the protected path.

In global path protection, the path that carries traffic under normal operating conditions is called the Active Path (AP), and the path that carries traffic when some fault affects the AP is called the Backup Path (BP). Although transmission networks implement many protection techniques at the physical layer, recovery at upper layers is also necessary. A Shared Risk Link Group (SRLG) is a set of links (or edges) in the network that share a common physical resource subject to fault(s). Since telecommunication networks are intrinsically multi-layered, the fault of a single network resource implies the fault in all links in the corresponding SRLG, appears at an upper layer as a multi-fault event. In this context, the concept of SRLG allows an upper layer to select, for a given AP, a backup path, which avoids every SRLG that may involve the selected AP, in the event of a failure. That is a SRLG diverse path set may be defined as a set of paths, between a source and a destination, so that no pair of paths can simultaneously be affected by any given failure (or risk) in a single failure scenario.

Extensions to the Open Shortest Path First (OSPF) routing protocol in support of carrying link state information for Generalized Multi-Protocol Label Switching (GMPLS) are specified in [12]. The SRLG is defined in [12] as sub-type-

length-value (sub-TLV) of the Link TLV. The existence of SRLG information is also considered in Multi-Protocol Label Switching (MPLS) Fast Reroute [18]. This confirms the need for efficient routing algorithms capable of calculating SRLG diverse paths.

One of the most frequent goals, when protection schemes are used in telecommunication networks, is the minimization of the used bandwidth. Assuming that no bandwidth sharing is allowed among protection paths, this problem is designated as the min-sum problem. When bandwidth sharing is allowed among protection paths, this problem can be approached as an asymmetrically weighted pair of disjoint paths [13,23]. When the objective is to obtain the shortest AP (considering that this path will define the QoS of the traffic flow most of the time) that may be protected, the problem is designated as the min-min problem.

The problem of finding a SRLG diverse path pair has been shown to be NP-Complete [11]. Hu *et al.* [11] proposed an integer linear programming (ILP) formulation for the min-sum problem, and provided numerical results showing that the ILP formulation is quite effective in networks with a few hundreds of nodes. Rostami *et al.* [14] proposed an algorithm, named CoSE, which is an extension to SRLG-disjoint routing of a link-disjoint routing algorithm called Conflicting Link Exclusion (COLE), proposed in [24]. COLE algorithm has the drawback that requires the calculation of a $s-t$ min-cut with minimal positive capacity (where s and t are a source and destination node) for solving the min-min link-disjoint path pair problem. Xu *et al.* [24] introduce the concept of conflicting link set, T , for a given AP, which is the subset of the links on the AP such that no AP using all these links can find a link-disjoint BP. Then, set T is used to divide

the min-min problem in smaller problems. CoSE algorithm solves the min-min problem extending the COLE algorithm [24], by replacing the calculation of the conflicting link set by the calculation of the conflicting SRLG set (which does not require the calculation of any min-cut). CoSE algorithm iteratively splits the network SRLGs into two sets and computes the working and backup paths respectively from each set. The minimum-cost diverse routing problem has the objective of finding two paths, SRLG diverse, with minimal total link cost (min-sum problem). In global dedicated protection against a SRLG single fault, the minimization of bandwidth usage may be sought using path pairs of least additive cost. An iterative heuristic is proposed by Todimala et al. [22], based on a modification of Suurballe's algorithm [21], for diverse routing under SRLG constraints that computes the least cost SRLG diverse path pair. In [23] the same authors propose a heuristic for solving the problem of computing optimal SRLG/link diverse paths under shared protection (considering the definition of an optimal SRLG diverse path pair under shared protection as asymmetrically-weighted [13]). The heuristic proposed in [23] for solving this problem is an improvement of the Iterative Two-Step Approach (ITSA) [10]. In [19,20] the authors considered the problem of path protection in wavelength-routed networks with SRLGs, proposed a heuristic method, which they compared with the Trap Avoidance (TA) algorithm [25]. They concluded that the new algorithm, Minimum Total Weight (MTW), outperforms TA within the first few iterations. If more iterations are considered there is no clear advantage of one algorithm over the other.

In [8] an algorithm to enumerate SRLG diverse paths, by non decreasing order of their total (additive) cost, was

presented, which is based on an algorithm proposed in [4] to generate minimal cost node-disjoint path pairs. The SRLG diverse path pairs may be node or edge-disjoint, with or without length constraints.

In this work, a new version of CoSE, designated as CoSE-MS, to solve the min-sum problem, will be described. This is achieved by introducing a modified Bhandari's edge-disjoint shortest path pair algorithm [3] and using the Modified Suurballe's Heuristic (MSH) proposed in [22], in the original version of CoSE. In [7] a first version of CoSE-MS was proposed and some preliminary results using random networks were shown. In this work, additional new results using random networks with different ranges for the cost of the links, are now presented. Furthermore, a virtual topology of lightpaths over a WDM network is created, so that the existing SRLGs are determined by the underlying physical network, and simulation results using incremental traffic are presented and discussed. Although the Iterative Modified Suurballe's Heuristic (IMSH) proposed in [22] is quite efficient, it sometimes takes too long to detect that no solution exists. Experimental results will show that CoSE-MS has a lower CPU time than IMSH-MS, albeit with the handicap that the optimality of the solution can not be verified. Nevertheless it will also be shown (using IMSH as a reference) that it obtains a high percentage of optimal solutions. Therefore CoSE-MS is a good compromise between the quality of obtained solutions and used CPU time, if the objective is seeking a min-sum SRLG-disjoint pair in a very short time.

The paper is organized as follows. Firstly the notation and the problem formulation, will be given. Secondly a short revision of CoSE [14] and of IMSH [22], will be presented in section 3. Thirdly a new version of CoSE, designated as

CoSE-MS, for solving the min-sum problem, will be proposed. Then CoSE-MS and IMSH relative performance, will be discussed, based on two different sets of experiments. The first set uses random undirected networks and the second set uses a virtual topology over a WDM network, so that the existing SRLGs are determined by the underlying physical network. Finally some conclusions will be presented in section 6.

2 Notation

Let $G = (N, A)$ be an undirected network with node set $V = \{v_1, v_2, \dots, v_n\}$ and edge (or link) set $A = \{a_1, a_2, \dots, a_m\}$ (where n and m designate the number of nodes and edges in G respectively). Let a non-negative cost function (or metric) in the edges (undirected arcs), be defined:

$$c_{v_a v_b} \geq 0, \quad (v_a, v_b) \in A \quad (1)$$

where $c_{v_a v_b}$ represents the cost of using edge (v_a, v_b) . The cost $c(p)$ of a path p in G with respect to metric c is:

$$c(p) = \sum_{(v_a, v_b) \in p} c_{v_a v_b}. \quad (2)$$

Definition 1 A path p is said to be simple (or loopless) if all its nodes are different.

We will use the word ‘‘path’’ to refer to simple paths, and the expression ‘‘simple path’’ we will only be used when required.

Let path $p = \langle v_1, a_1, v_2, \dots, v_{i-1}, a_{i-1}, v_i \rangle$, be given as an alternate sequence of nodes and edges of G , such that the tail of a_k is v_k and the head of a_k is v_{k+1} , for $k = 1, 2, \dots, i-1$ (all the v_i in p are different). Let the set of nodes in p be $V^*(p)$ and the set of edges in p be $A^*(p)$. Two paths $p = \langle v_1, a_1, v_2, \dots, v_{i-1}, a_{i-1}, v_i \rangle$ and q are edge-disjoint if

$A^*(p) \cap A^*(q) = \emptyset$. Two paths p and q are disjoint if $V^*(p) \cap V^*(q) = \emptyset$, and are internally disjoint [1] if $\{v_2, \dots, v_{i-1}\} \cap V^*(q) = \emptyset$. It will be said that two paths are node-disjoint if they are internally disjoint.

Let R be a set representing the risks (faults) in the functional network. Each risk may correspond to a fiber cut, a card fault at a node, a software fault, or any combination of these factors. Let A_r represent the sub-set of network edges (or links) in the network logical representation (corresponding to a capacitated graph) that can be affected by risk $r \in R$. Thence A_r is a SRLG (Shared Risk Link Group associated with r). Let,

$$R_p = \{r \in R : \text{path } p \text{ contains elements of } A_r\} \quad (3)$$

The SRLG problem can be defined as follows [11].

Definition 2 Find two paths p and q , between a pair of nodes, such that $R_p \cap R_q = \emptyset$. We also say that p and q are two SRLG diverse paths (with respect to R).

The min-min problem addressed in [14], for obtaining a SRLG-disjoint (and edge-disjoint path pair, assuming all links belong to at least one SRLG), can be formalized as follows, with $c_m[(p, q)] = \min[c(p), c(q)]$

$$\arg \min_{(p, q)} c_m(p, q) : R_p \cap R_q = \emptyset \quad (4)$$

Similarly, the min-sum problem solved in [22] can be formalized as follows, with $c_s[(p, q)] = c(p) + c(q)$:

$$\arg \min_{(p, q)} c_s[(p, q)] : R_p \cap R_q = \emptyset \quad (5)$$

3 Revision of CoSE and IMSH

3.1 Revision of CoSE Algorithm

For a known AP, its Conflicting Link Set T is a subset of the links on the active path such that no active path using

all these problematic links can find a link-disjoint backup path [24]. Let I (the inclusion set) and E (the exclusion set) be two disjoint subsets of A (the edge set of the network). CoLE [24] solves the min-min routing problem by using a divide-and-conquer technique based on the conflicting link set T . Let $P(I, E)$ designate the problem of finding a pair of active path and backup path, where the active path is the shortest among all possible active paths that must use the links in I but not the links in E . Then, the original min-min problem can be represented by $P(\emptyset, \emptyset)$. In fact, because Dijkstra's algorithm [5] is used for solving this problem [24], what is ensured is that the edges affected by the SRLGs in E are not used, and the edges in I may be used.

Let the conflicting link set be $T = \{a_1, a_2, \dots, a_{|T|}\}$, for a given AP. Then each problem $P(I, E)$ is divided into the following sub-problems, $P(\emptyset, \{a_1\})$, $P(\{a_1\}, \{a_2\})$, \dots , $P(\{a_1, a_2, \dots, a_{|T|-1}\}, \{a_{|T|}\})$. A problem P has no solution if its sub-problems (divided by this way) have no solution [24]. The number of sub-problems to be solved for each problem is equal to $|T|$.

CoSE algorithm solves the min-min problem extending the Conflicting Link Exclusion (CoLE) algorithm [24], by replacing the calculation of the conflicting link set by the calculation of the Conflicting SRLG Set (CoSE). The Conflicting SRLG Set T [14] for a given active path, is defined as a subset of the SRLGs along it, such that no active path using all these problematic SRLGs can find a SRLG-disjoint backup path. An algorithm (see appendix) is proposed [14] which sequentially removes the SRLGs along an active path from the network until no other path can be found: the resulting set of removed SRLGs defines T .

The min-min SRLG-disjoint path pair problem is solved starting with the $P(\emptyset, \emptyset)$ problem. A shortest path is calculated (the candidate AP) and a SRLG-disjoint path is sought in the network, after removing all the edges with any SRLG in common with that AP. If a solution exists the algorithm ends (in this case with an optimal solution); otherwise the conflicting SRLG set T , $T = \{A_1, A_2, \dots, A_{|T|}\}$, of this AP is calculated. Then the algorithm seeks to solve the sub-problems: $P(\emptyset, \{A_1\})$, $P(\{A_1\}, \{A_2\})$, \dots , $P(\{A_1, A_2, \dots, A_{|T|-1}\}, \{A_{|T|}\})$. Each sub-problem will possibly result in a new candidate AP; if that new AP has no SRLG-disjoint path the conflicting SRLG set for that AP is calculated and the algorithm tries to solve the new problem (in a network where the edges affected by the SRLGs in the exclusion set of the preceding problem(s) can not be used by the AP calculated in the new problem). A list stores all solutions, obtained from the generated sub-problems, and the path pair with the shortest path is selected.

3.2 Revision of IMSH Algorithm

Suurballe's algorithm [21] finds, in polynomial time, the least cost link-disjoint path pair in a graph, if such a path-pair exists.

In fact that algorithm can be used to find, for each possible sink v_i , a pair of edge-disjoint paths from s to v_i of minimum total edge cost. In Suurballe's algorithm the tree of shortest paths from a source node s to all nodes, τ_s , is calculated and then every edge cost is replaced by their reduced cost \bar{c} [6]. Let $\pi_{v_i}(\tau_s)$ denote the cost of a path from s to v_i in τ_s ; the reduced cost of the directed edge (v_i, v_j) , associated with τ_s is given by $\bar{c}_{v_i v_j} = c_{v_i v_j} - \pi_{v_j}(\tau_s) + \pi_{v_i}(\tau_s)$.

This guarantees that every edge in τ_s has zero reduced cost, and hence that every path in τ_s has zero cost. Also, any two paths, p and q , with the same source and destination node preserve their ordering, that is if $c(p) < c(q)$ then $\bar{c}(p) < \bar{c}(q)$.

To determine the min-sum edge-disjoint path pair from s to t , the modified graph \bar{G} is obtained by the calculation of the reduced costs (\bar{c}). A new modified graph is derived, \bar{G}^* , identical to \bar{G} , but where every link (v_i, v_j) in the shortest path, p_1 , from s to t (in τ_s) is removed and replaced by a directed link (v_j, v_i) with cost $-\bar{c}_{v_i v_j}$ (which is zero in \bar{G} , due to the previous graph transformation); then the shortest path, p_1^* is calculated in \bar{G}^* using Dijkstra's algorithm. Every directed edge in p_1^* whose reversal appears in p_1 is designated as an interlacing edge. These interlacing edges must be removed from paths p_1 and p_1^* to get a pair of least cost link-disjoint paths. This approach can not be used to obtain an edge-disjoint path for the k -th shortest path p_k , because the cost of edge (v_i, v_j) in p_k may not be zero and a negative cycle could occur in \bar{G}^* .

In [22] it is proposed a modification of Suurballe's algorithm (MSH) so that it can be applied to the k -th shortest path for obtaining a pair of SRLG-disjoint paths. The reduced costs are not calculated in MSH (and therefore τ_s is not required). As in Suurballe's algorithm, the edges from s to t in p_k are removed from the new graph, G' , the directed links from t to s in p_k are added with null cost; then the cost of all the links in the graph that are in SRLG conflict with the links along the path p_k is increased by M (sum of the costs of all the links in the network). The rest of the heuristic is similar to the Suurballe's algorithm except that after removing the interlacing edges between p_k and p_1' , the obtained

path pair (p_k', q_k') is only considered an admissible solution if p_k' and q_k' are SRLG-disjoint.

The Iterative Modified Suurballe's Heuristic (IMSH) sequentially generates K shortest path using Yen's algorithm [26]; then for each obtained p_k (k -th shortest path, $k = 1, \dots, K$) it uses the Modified Suurballe's Heuristic to calculate a pair of SRLG-disjoint paths based on each p_k , and keeps a record of the path pair with current lowest additive cost. The algorithm ends after generating K shortest paths or earlier if the recorded SRLG-disjoint path pair (current best solution) was obtained, (p, q) such that $c[(p, q)] \leq 2 \times p_k$; this last condition is shown in [22] to ensure the optimality of the obtained solution.

4 Description of CoSE-MS

In [3] it is presented an algorithm, which uses negative costs, to obtain a min-sum edge-disjoint path pair. In this algorithm, the shortest path, p_1 , between a given source and a destination node (s, t) , is calculated; a modified graph, G' identical to G is created, where every link (v_i, v_j) in the active path is removed and replaced by the directed link (v_j, v_i) with cost $-c_{v_i v_j}$; then the shortest path, p_1' in this network is calculated, using a modified version of Dijkstra's algorithm [3]; finally, the interlacing edges on paths p_1 and p_1' , to get a pair of least cost link-disjoint paths, must be removed. A modified version of Dijkstra's algorithm can be used in G' because no negative cycles are created in G' . However if the same approach were to be used with the k -th shortest p_k , such that $c(p_k) > c(p_1)$, negative cycles may appear and the modified Dijkstra's algorithm could no longer be applied.

The algorithm proposed in this work is an efficient heuristic to calculate a pair of edge and SRLG-disjoint path pair

seeking to minimize its min-sum cost. This is achieved by introducing Bhandari's edge-disjoint shortest path pair algorithm, taking into account the SRLGs in the shortest path, designated here as Modified edge-disjoint Bhandari's heuristic (using an approach similar to MSH in [22]), and also incorporating MSH in the original version of CoSE.

In the Modified Bhandari's edge-disjoint shortest path pair Heuristic (MBH) the edges from s to t in p_1 (shortest path in the network) are removed from the new graph, G' , the directed links from t to s in p_1 are added with symmetric (negative) cost and the cost of all the links in the graph that are in SRLG conflict with the links along the path p_1 is increased by M (sum of the costs of all the links in the network). Then the shortest path in this new network is calculated, p'_1 , using the modified Dijkstra's algorithm. Finally one must remove the interlacing edges on paths p_1 and p'_1 to get a pair of least cost link-disjoint paths. Note the calculated path pair may not be SRLG-disjoint.

The algorithm CoSE presented in [14], is a recursive algorithm. In the recursive version every problem to be solved must know the problems it is derived from, in particular the set of SRLGs previously excluded by the problem(s) from which the current problem originated. The algorithm CoSE-MS is presented in iterative form. For that purpose, a third element was added to each problem the set of previously excluded SRLGs. Hence, now a problem is designated by $P(I, E, H)$, where I is the inclusion set, E the exclusion set and H the union of all the exclusion sets of the problems that originated the current problem P . The initial problem will have I , E and H equal to \emptyset . Note that the shortest path of problem $P(I, E, H)$ is calculated excluding from the graph all links affected by SRLGs in the set $E \cup H$.

Algorithm CoSE-MS: Conflicting SRLG exclusion for the Min-Sum problem (CoSE-MS)

Data: Network undirected graph $G = (N, A)$, node pair (s, t) , c cost of the links and R set of risks.

Result: (p_S, q_S) , a path pair from s to t , seeking to minimise $c_s[(p_S, q_S)]$ subject to $R_{p_S} \cap R_{q_S} = \emptyset$

```

1 Consider an empty stack  $S$ 
2 Initialize:  $(p_S, q_S) = (\emptyset, \emptyset)$ ,  $c_s(\emptyset, \emptyset) = \infty$ 
3 Consider  $P(\emptyset, \emptyset, \emptyset)$  as the original problem
4 push ( $P$ )
5 while  $\neg$  empty ( $S$ ) do
6    $P_c(I_c, E_c, H_c) \leftarrow$  top ( $S$ ) // current problem
7   pop ( $S$ ) // remove top of  $S$ 
8    $p_c \leftarrow$  Shortest path in the current problem
9   if  $c(p_c) \neq \infty$  //  $p_c$  exists
10  then
11    if  $I_c = \emptyset \wedge E_c = \emptyset \wedge H_c = \emptyset$  then
12      MBH ( $G, R, c, p_c, p, q$ )
13    end
14  else
15    MSH ( $G, R, c, p_c, p, q$ )
16  end
17  if  $c_s[(p, q)] \neq \infty$  then
18    //  $p$  is SRLG-disjoint with  $q$ 
19    if  $c_s(p, q) < c_s(p_S, q_S)$  then
20      // Updates current solution
21       $(p_S, q_S) \leftarrow (p, q)$ 
22    end
23  end
24  else
25    SRLG_Exclusion ( $I_c, p_c, T$ )
26    // Let  $T$  be  $\{A_1, A_2, \dots, A_{|T|}\}$ 
27     $H \leftarrow E_c \cup H_c$ 
28     $P_1(I_1, E_1, H_1) \leftarrow P(\emptyset, \{A_1\}, H)$ 
29    push ( $S, P_1$ )
30     $i \leftarrow 2$ 
31    while  $i \leq |T|$  do
32       $P_i(I_i, E_i, H_i) \leftarrow P(I_{i-1} \cup E_{i-1}, \{A_i\}, H)$ 
33      push ( $S, P_i$ )
34       $i \leftarrow i + 1$ 
35    end
36  end
37 end
38 // If  $c_s(p_S, q_S) = \infty$  no solution was found.
39 // Otherwise  $(p_S, q_S)$  is the solution to
40 // the min-sum problem.

```

The algorithm, in its non-recursive version, requires a stack S of problems to be solved with the usual operations:

push(S,P) (insert a new problem P at the top of stack S), pop(S) (remove the top element from stack S), top(S) (return the top element from stack S , without removing it, empty(S) (returns true if S is empty, and false otherwise).

In CoSE-MS, function MSH(G,R,c,p_c,p,q) represents the modified Suurballes's heuristic, where p_c is the shortest path in the current problem (and corresponding pruned network), and (p,q) is the resulting pair of SRLG-disjoint paths; similarly, the modified Bhandari's heuristic is represented by function MBH(G,R,c,p_c,p,q), where p_c is the shortest path in the original network, and (p,q) is the resulting pair of SRLG-disjoint paths in the network. If such pair does not exist its cost will be ∞ .

The function SRLG_Exclusion(I_c,p_c,T) represents the algorithm "Finding a conflicting SRLG set for a given active path p_c from node s to node t " in [14] which is given in the appendix.

The main steps of CoSE-MS are similar to CoSE, differing in the calculation of the path pair corresponding to each path p_c , using the modified Bhandari's algorithm in the original problem or MSH in all the problems derived from the original problem. Another difference between CoSE-MS and CoSE is in line 25, where I_1 is equal to \emptyset instead of I_c , because it was verified that this leads to a higher number of optimal solutions in CoSE-MS. This condition may lead to a larger set T , but the algorithm will end because every new problem has a larger set of excluded SRLGs (the union of E and H), until no AP can be found.

The usage of MBH when $p_c = p_1$, results in a path pair of cost lower or equal to the cost of the corresponding path pair obtained using MSH, if a SRLG-disjoint path pair exists derived from p_1 . This is relevant, because CoSE-MS ends if

a SRLG-disjoint path pair is obtained using p_1 . Otherwise the algorithm, after calculating the conflicting SRLG set, T , proceeds to solve the problem(s) derived from T . The algorithm ends when all problems have been solved and stack S is empty. The best found solution is stored in (p_S,q_S) ; if $(p_S,q_S) = (\emptyset,\emptyset)$ and $c_s(p_S,q_S) = \infty$, no solution exists.

The proposed algorithm obtains link-disjoint SRLG diverse path pairs, but it can be easily adapted to obtain node-disjoint solutions by using a vertex-splitting method [3].

4.1 An example

Consider the network in the Figure 1 (a) and (b), adapted from the network in [14]. Assume all edge costs are equal to 1 and that a SRLG-disjoint path pair is sought between nodes 1 and 3, using CoSE-MS.

The initial problem, $P(\emptyset,\emptyset,\emptyset)$, calculates the AP candidate, $p_c = \langle 1,e_5,3 \rangle$, which is affected by SRLGs g_1,g_4 and g_7 . MBH receives a copy of the original network, removes edge e_5 , adds the directed edge $(3,1)$ with cost -1 and increases the cost of edges e_1 , and e_3 (which have a SRLG in common with p_c) by $M = 5$; then, in this modified network, calculates the shortest path from node 1 to 3, $\langle 1,e_1,2,e_2,3 \rangle$; no interlacing edges exist between this path and p_c and $(p,q) = (\langle 1,e_5,3 \rangle, \langle 1,e_1,2,e_2,3 \rangle)$ is calculated; MBH returns an ∞ cost for $c(p,q)$, informing that p and q are not SRLG-disjoint. Hence the algorithm proceeds to step 23, where set $T = \{g_1,g_4\}$, is calculated. Two new problems are added to the stack: $P(\emptyset,\{g_1\},\emptyset)$ (in line 25 of CoSE-MS) and $P(\{g_1\},\{g_4\},\emptyset)$ (in the cycle controlled by line 28 of CoSE-MS).

The next problem to be solved is $P_c = P(\{g_1\},\{g_4\},\emptyset)$. The shortest path, from node 1 to 3, in a network where

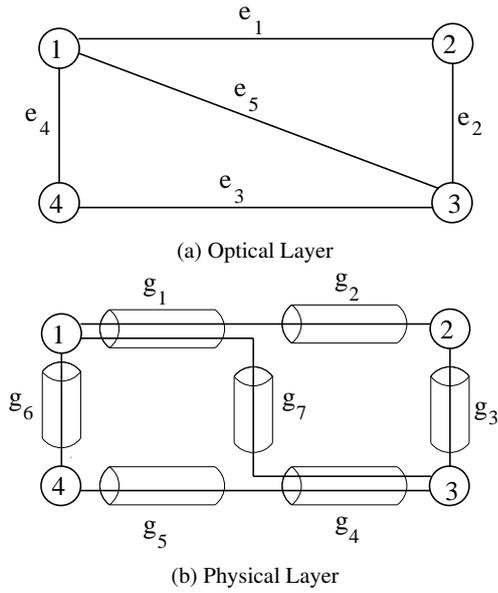


Fig. 1 Layered Architecture of an Optical Network based on [14].

edges e_3 and e_5 have been removed, because SRLG g_4 has been excluded from the network, is $p_c = \langle 1, e_1, 2, e_2, 3 \rangle$. Path p_c is affected by SRLGs g_1, g_2 and g_3 . MSH receives a copy of the original network, removes edges e_1 and e_2 (edges of p_c), adds the directed links $(3,2)$ and $(2,1)$ with cost 0, and increases the cost of link e_5 (which has a SRLG in common with p_c) by M ; the shortest path in this network is $\langle 1, e_4, 4, e_3, 3 \rangle$, which has no interlacing edges with the current p_c , and the resulting path pair is $(p, q) = (\langle 1, e_1, 2, e_2, 3 \rangle, \langle 1, e_4, 4, e_3, 3 \rangle)$, which is SRLG-disjoint; this path pair is returned by MSH with cost $c(p, q) = 4$ and hence $(p_S, q_S) = (p, q)$.

Then the algorithm would remove the next (and last) problem in stack S , $P_c = P(\emptyset, \{g_1\}, \emptyset)$. The solution to this problem would be the path pair $(p, q) = (\langle 1, e_4, 4, e_3, 3 \rangle, \langle 1, e_1, 2, e_2, 3 \rangle)$, identical to the previous path pair, and the current best solution, (p_S, q_S) , would remain unchanged. This will end the cycle controlled by line 5 of the algorithm (S is now empty). Finally, the best solution, (p_S, q_S) , would be returned, as indicated in line 35 of CoSE-MS.

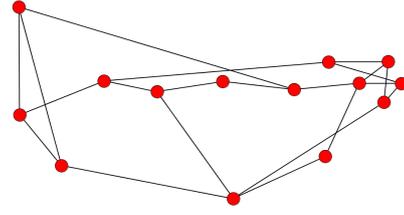


Fig. 2 Fourteen node NSFNET [2].

5 Performance Evaluation

Two different sets of experiments were made. The first set uses random undirected networks and the second uses a virtual topology of lightpaths over the physical topology of NSFNET [2] (see Figure 2), a well known optical transport network.

Random networks were used to evaluate the algorithm performance solving the SRLG-diverse min-sum routing problem, in very different size networks of low density, with randomly generated SRLGs. The lightpath network allows observing the impact on network performance of using CoSE-MS (which does not ensure optimal solutions) versus the larger CPU time required by IMSH, in a scenario where SRLGs were obtained using an underlying optical transport network.

In our implementation of IMSH we used the MPS algorithm [17] in its loopless version. Note that although Yen's algorithm has the lowest worst case complexity among k -shortest path ranking algorithms [26, 15], we prefer to use MPS because, in [16], experimental results show that in practical situations this algorithm is more efficient than Yen's, in terms of CPU time and RAM space.

5.1 Performance evaluation using random networks

For each number of nodes ($n = 25, 50, 100, 200, 400, 800$) ten different networks, with a given number of edges, were

randomly generated¹. For each of these networks the costs of the arcs were randomly generated in two different ranges, $[1, 65535]$ and $[1, 255]$. For every node of degree 3 or higher, every two out of three (emergent) adjacent links were randomly selected as belonging to the same SRLG.

For IMSH, the value of K was set equal to 1000, for tested networks (which will ensure, as will be shown, a very high number of optimal solutions).

In each experiment, for each network 100 end-to-end node pairs were randomly selected. To evaluate the impact of the range of the costs of the links in the algorithms' performance, the same set of 100 end-to-end node pairs were considered, for each pair of networks with the same topology, but with different ranges for the cost of the links.

The considered performance measures were: the number of optimal solutions, the CPU time to obtain those solutions and the relative error of sub-optimal solutions of CoSE-MS. In [7] results were presented for similar experiments, with link costs in the range $[1, 65535]$. The results in this work are slightly different for that same link costs range, because new seeds for the random generation of the set of end nodes were used. This set of experiments used an Intel(R) Core(TM)2 CPU 6320+ @ 1.86 GHz under Ubuntu.

In Figures 3 and 5 the average CPU time per node pair, considering node pairs for which a solution was obtained, is presented for $n = 25, 50, 100$ and for $n = 200, 400, 800$, respectively, when link costs are in $[1, 255]$. Equivalent results are given in Figures 4 and 6 when link costs are in $[1, 65535]$. The vertical bars represent the 95% confidence interval for the average CPU time. It can be seen that CoSE-MS has a

significantly better performance than IMSH, ranging from 3 times faster for $n = 25$ to over 10 times faster for $n = 800$, when $m = 2n$, for both ranges of the cost of the links. For $m = 3n$ CoSE-MS is 3 to 11 times faster than IMSH, again for both ranges of link costs.

Regarding the obtained optimal solutions, it can be seen in Figures 7 and 8 that the number of optimal solutions found by CoSE-MS is very high (the number of CoSE-MS's optimal solutions was calculated comparing the cost of each solution with the corresponding optimal cost obtained by IMSH). The lowest value in the Figure 7 is 87.4% for CoSE-MS versus 99.7% for IMSH and in the Figure 8 is 87.5% for CoSE-MS versus 99.7% for IMSH, in both cases when $n = 400$ and $m = 2n$. For $m = 3n$ the number of optimal solutions is larger, in the interval 92%-94.3%, as can be seen in Figures 7 and 8. In relative terms, CoSE-MS has 88%-90% and 91%-93% of the optimal solutions found by IMSH, when $m = 2n$ and $m = 3n$, respectively. The performance of

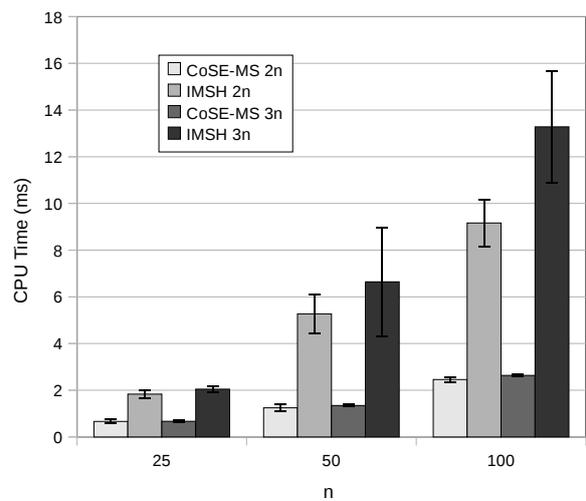


Fig. 3 Average CPU time for obtaining a solution, considering node pairs for which a solution was obtained, for $n = 25, 50, 100$, $m = 2n, 3n$ and costs in the range $[1, 255]$.

¹ The program used for network generation was kindly borrowed from José Luis Santos of the Department of Mathematics of the University of Coimbra.

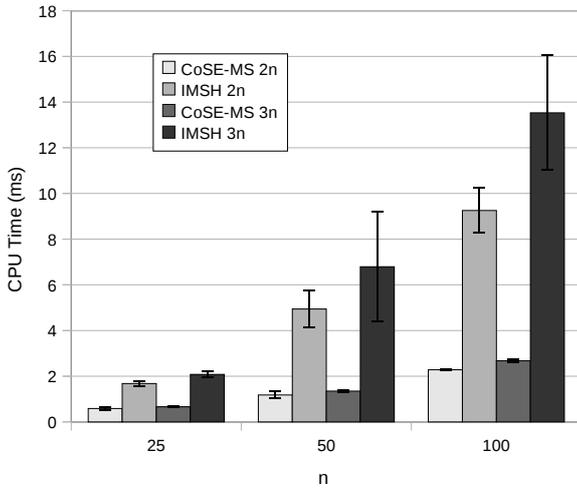


Fig. 4 Average CPU time for obtaining a solution, considering node pairs for which a solution was obtained, for $n = 25, 50, 100$, $m = 2n, 3n$ and costs in the range $[1, 65535]$.

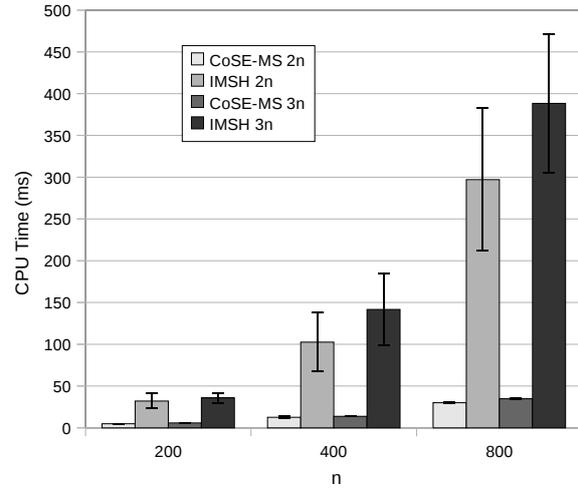


Fig. 6 Average CPU time for obtaining a solution, considering node pairs for which a solution was obtained, for $n = 200, 400, 800$, $m = 2n, 3n$ $[1, 65535]$.

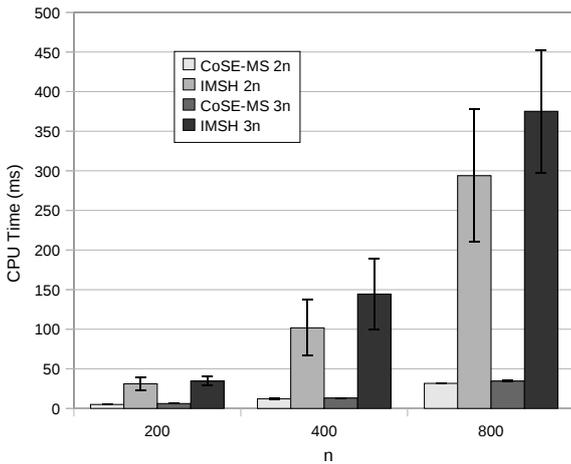


Fig. 5 Average CPU time for obtaining a solution, considering node pairs for which a solution was obtained, for $n = 200, 400, 800$, $m = 2n, 3n$ costs in the range $[1, 255]$

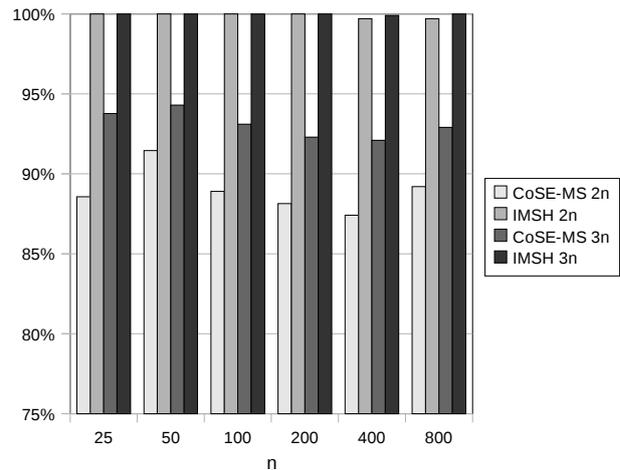


Fig. 7 Average Percentage of optimal solutions for $m = 2n, 3n$ and costs in the range $[1, 255]$.

both algorithms does not seem to depend on the range of the cost of the links.

The same node pairs were identified by both algorithms, as having no solution, but CoSE-MS identified them in much shorter time than IMSH. In the tested networks, the highest number of node pairs for which no solution was obtained was observed in the networks with $n = 25$ and $m = 2n$. For

these networks the CPU time per node pair without solution was in average 18 ms and 19 ms for CoSE-MS, 706 ms and 728 ms for IMSH (which are much higher than the average CPU time when a solution was obtained), for costs in the range $[1, 65535]$ and $[1, 255]$, respectively. In the tested cases IMSH ended with very few sub-optimal solutions: one for $n = 400$, $m = 3n$, three for $n = 400, 800$ and $m = 2n$,

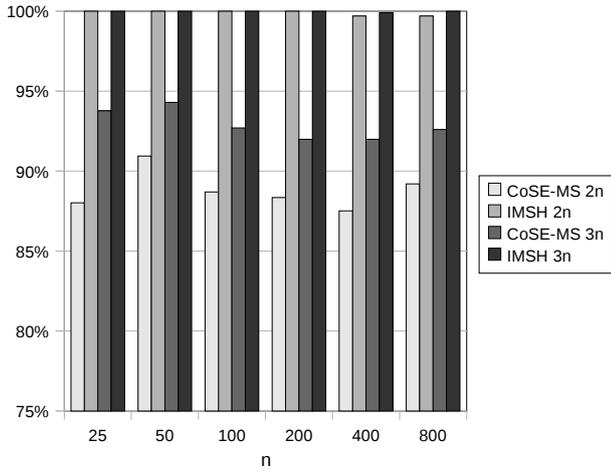


Fig. 8 Average Percentage of optimal solutions for $m = 2n, 3n$, and costs in the range $[1, 65535]$.

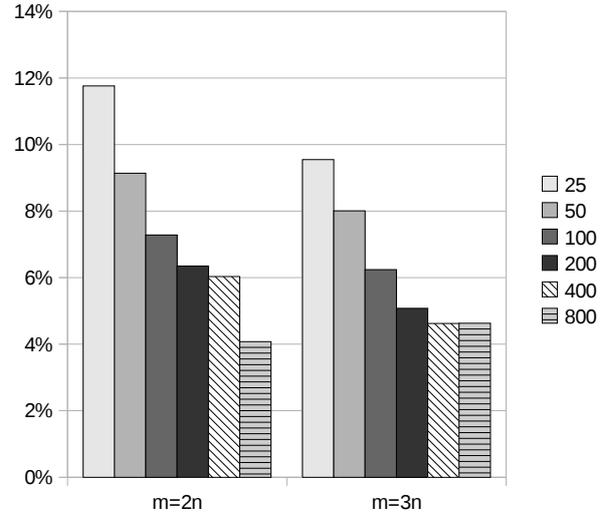


Fig. 10 Average relative error of the sub-optimal solutions obtained by CoSE-MS, when IMSH obtains an optimal solution for $m = 2n, 3n$, and costs in the range $[1, 65535]$.

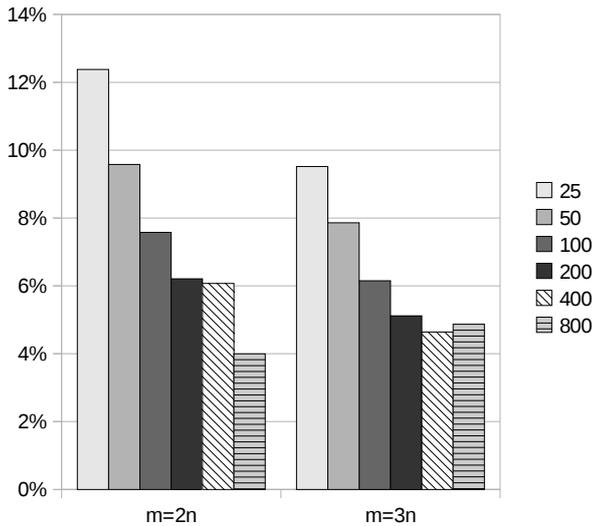


Fig. 9 Average relative error of the sub-optimal solutions obtained by CoSE-MS, when IMSH obtains an optimal solution for $m = 2n, 3n$, and costs in the range $[1, 255]$.

regardless of considered ranges (6 sub-optimal solutions in 6000 node pairs, for each range cost).

Regarding the accuracy of the sub-optimal solutions calculated by CoSE-MS, the average relative error of the sub-optimal solutions obtained by CoSE-MS, when IMSH obtains an optimal solution, is presented in Figures 9 and 10. The relative error of the sub-optimal solution is marginally

higher for the smaller range cost, especially for the smaller networks. The smaller networks, $n = 25$ for $m = 2n$, presented the highest average relative error, 11.8% and 12.4% for costs in the range $[1, 65535]$ and $[1, 255]$, respectively; also for $m = 3n$, for both range costs, the smaller networks, $n = 25$, presented the highest average relative error, 9.5%. For larger networks ($n = 400, 800$) the relative error was less than 6%, as can be seen in Figures 9 and 10.

5.2 Performance evaluation in an optical network

This second experiment considers a virtual topology over NSFNET (see Figure 2). The virtual arcs in this virtual topology consist of direct lightpaths between every pair of nodes whose distance does not exceed 2500 Km [9]. SRLGs are associated with the virtual arcs so that two arcs that transverse the same physical link share a common SRLG. This resulted in a network with 14 nodes and 54 undirected arcs. The number of lightpaths needed in each arc was determined by dimensioning the virtual topology for supporting a traffic

matrix proportional to the one presented in [2]. For routing purposes, the cost of an arc in the virtual topology was considered equal to 1 added with $\alpha \times (h - 1)$ where h is the number of hops that lightpath transverses in the underlying physical layer, and α is a small number. The cost of an active path in the virtual topology, given by equation (2), will reflect the number of used lighpaths and of the underlying physical hops, minimizing not only the virtual hop distance, but also the physical hop length. The cost of the links were in [1,1.3], taking $\alpha = 0.1$. The nodes were considered to have full wavelength conversion capability. Twenty experiments were made, using different seeds for the generation of connection requests. The bars around average performance values represent the width of the corresponding 95% confidence interval.

Using incremental traffic, according to the traffic matrix in [2], the following performance measures were considered: number of accepted requests, used bandwidth, number of blocked requests in each batch of 50 connection requests, average CPU time per request (accepted of blocked), and average CPU time per request (accepted of blocked) in each batch of 50 connection requests. These measures seek to evaluate the relative performance of CoSE-MS and IMSH when the SRLG information has a relation with the underlying optical network. In this second set of experiments a AMD 64X2 4800 @ 2.4 GHz computer, also under Ubuntu, was used.

In the previous experiments with random networks, it was observed that the larger the networks the more significant was the difference of the CPU times required by CoSE-MS and IMSH. For the smaller networks, $n = 25$, CoSE-MS was only 3 times faster than IMSH, but CoSE-MS had

a significant advantage over IMSH when no solutions were found. When incremental traffic is used, eventually the network will become congested and IMSH's performance will depend on the value of K . The value of K , such that IMSH still obtains a high number of optimal solutions without compromising its performance when no solution can be found, is not easy to determine. Considering that the virtual topology used in the second set of experiments is smaller (only 14 nodes), but slightly denser ($m = 54$) than the previously considered random networks, two smaller values for K were used, $K = 100, 500$.

It was verified that both $K = 100$ and $K = 500$ led to the same solutions in the tested network (all calculated solutions were optimal, or no solution could be obtained), so results will be given for IMSH without reference to K , except when CPU time is discussed. Until the 900th request no difference can be observed regarding the used bandwidth by both algorithms. Therefore in Figures 11 and 12 results are presented with the number of requests starting in 900. The difference between the number of accepted requests in CoSE-MS and IMSH can only be visible after 1050 requests and the final difference is in average only 4.5 requests in over 1000 accepted requests. In Figure 12, it can be seen that CoSE-MS uses slightly more bandwidth than IMSH, as would be expected, because CoSE-MS solutions are not always optimal. After 1200 requests CoSE-MS approaches IMSH, but it has slightly less accepted requests (see Figure 11).

To evaluate the quality of the solutions obtained by CoSE-MS in this network, the number of optimal solutions and the relative error of sub-optimal solutions of CoSE-MS were calculated running the simulator using IMSH, and for each connection request calculating the corresponding solution

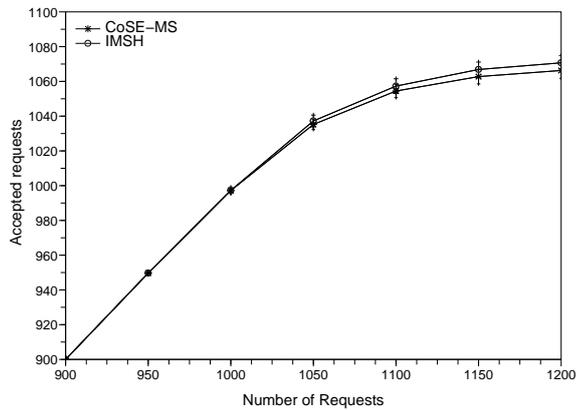


Fig. 11 Average number of accepted requests.

using CoSE-MS (for the same network state); and then the simulator proceeded using the solution obtained by IMSH. In a total of 24000 requests (1200 requests \times 20 different runs), both algorithms found a solution in 21414 cases, and only 100 times did the cost of the obtained solutions present a different value. Up to the 900th request both algorithms gave solutions with the same (optimal) cost. The observed 100 different solutions had an average relative error of 14,7% with a standard deviation of 12.5%.

Then the symmetrical experiment was done: running the simulator using CoSE-MS, and for each connection request calculating the corresponding solution using IMSH (for the same network state); and then the simulator proceeded using the solution obtained by CoSE-MS. Of the accepted 21326 requests with both algorithms, only 105 solutions of CoSE-MS had higher cost than the ones obtained by IMSH. In this test an average relative error of 12,8% with a standard deviation of 10.4% was observed.

These two experiments confirm the high number of optimal solutions and small relative error obtained using random networks.

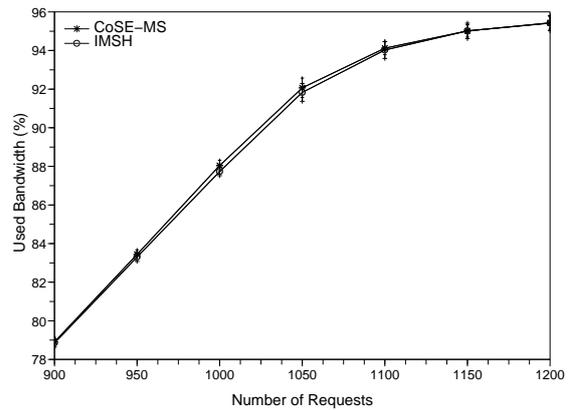


Fig. 12 Average used bandwidth.

Only after the 900th request there is a connection request rejection. In Figure 11 the number of accepted requests decreases very fast after the 1000th request (when the network has close to 88% of used bandwidth).

The average CPU time per request (accepted or rejected) is presented in Figure 13. IMSH uses only 25% more CPU time than CoSE-MS when the network is not congested. After the 950th connection request, when blocking appears (see Figure 11), IMSH CPU time grows significantly, especially when $K = 500$, while the CPU time used by CoSE-MS does not seem to be affected by network congestion. The confidence intervals for CoSE-MS are very narrow and not visible in Figures 13 and 14.

The average CPU time per request in each batch of 50 requests is given in Figure 14, where it can be confirmed that the CPU time per request, when there is a high rate of unsuccessful requests, strongly depends on K . Figure 14 confirms that, regarding CPU time, IMSH performance degrades significantly in the presence of network congestion, and more acutely for $K = 500$. Although it was verified that $K = 100$ was sufficiently large for a good performance of IMSH, this value is not easy to estimate. After the 1100 request the net-

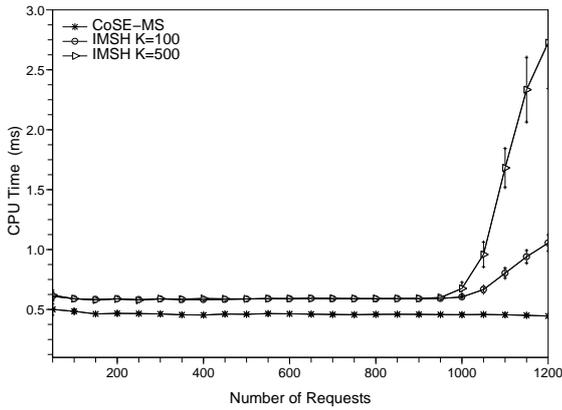


Fig. 13 Average CPU per connection request (accepted or rejected).

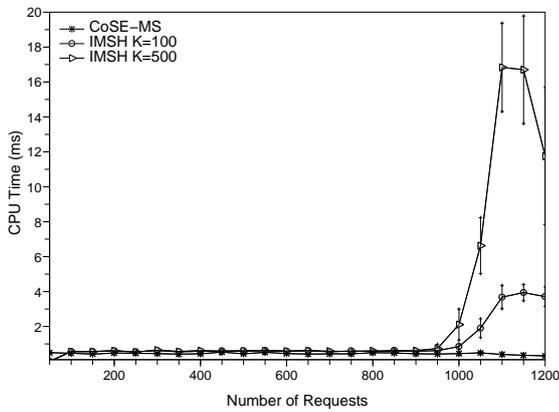


Fig. 14 Average CPU per connection request (accepted or rejected) in each batch of 50 requests.

work is so congested (it has 94% of used bandwidth – see Figure 12) that the CPU time per request decreases for both algorithms, because the residual network has less links than the original network, and is likely disconnected. Even for $K = 100$, the CPU time per request, after the 1000th request is significantly higher in IMSH, when compared with CoSE-MS.

To conclude it can be said that, regarding the number of accepted request and the amount of used bandwidth, IMSH slightly outperforms CoSE-MS. However all solutions obtained by CoSE-MS were optimal until congestion occurred in the network, and only 0.467% of the obtained solutions

were suboptimal. Regarding CPU time, due to the small size of the considered network, and in the absence of congestion, the CPU difference between CoSE-MS and IMSH is not as significant as observed in random networks, but CoSE-MS is advantageous since it always finds the optimal solution in less CPU time. Moreover, as soon as blocking appears, the difference between the two algorithms is significant, with a clear advantage for CoSE-MS, at the expense of a slightly lower number of successful connections.

6 Conclusion

In [11] the problem of finding a SRLG-disjoint path pair was shown to be NP-Complete. In today's networks, with the introduction of GMPLS, SRLG-disjoint path pairs have to be calculated for routing protected traffic flows. In this context it is important to obtain a solution in a very short time, in order to allow on-line SRLG diverse routing.

In this work an efficient heuristic (CoSE-MS) was proposed for calculating a SRLG-disjoint path pair, seeking to minimize its total (additive) cost. This heuristic modifies CoSE [14] (proposed for solving the min-min SRLG-disjoint path pair problem) by introducing a modified Bhandari's edge-disjoint shortest path pair heuristic (MBH) and using the Modified Suurballe's Heuristic (MSH) proposed in [22]. Although CoSE-MS does not give any information regarding the optimality of the obtained solutions, computational results using random networks showed that it requires considerably less CPU time than IMSH (especially in larger networks) and generates a significant number of optimal solutions. The results obtained using random networks were confirmed using a virtual lightpath network based on NSFNET network, with incremental traffic. The high number of op-

timal solutions obtained by CoSE-MS resulted in an average of used bandwidth and an average number of rejected requests very close to IMSH, using much less CPU time, especially when the network was congested. Therefore in the context of on-line protected routing, CoSE-MS is a good compromise between the quality of obtained solutions and of the used CPU time.

Acknowledgements

This work was financially supported by programme COMPETE of the EC Community Support Framework III and cosponsored by the EC fund FEDER and national funds (FCT), Project U308/1.3/NRE/07 and Ph.D. scholarship SFRH/BD/49739/2009 from Instituto Politécnico de Viseu.

A Conflicting SRLG set

In the algorithm to find a conflicting SRLG set for a given active path p from node s to node t [14] the function `insert` inserts a new element in a list of paths or a new element in a set of SRLGs and the function `erase` removes an element from a set of SRLGs and returns its value.

References

1. Bang-Jensen, J., Gutin, G. (2002). *Digraphs: Theory, Algorithms and Applications*. Springer Monographs in Mathematics. Springer-Verlag, Great Britain.
2. Betker, A., Gerlach, C., Jäger, M., Barry, M., Bodamer, S. Späth, J., Gauger, C. M., Köhn, M. (2003). Reference Transport Network Scenario. Technical report, MultiTeraNet Report.
3. Bhandari, R. (1999). *Survivable Networks, Algorithms for Diverse Routing*. Kluwer Academic Publishers, Norwell, Massachusetts, USA.
4. Clímaco, J., Pascoal, M. M. B. (2009). Finding non-dominated bicriteria shortest pairs of disjoint simple paths. *Computers & Operations Research*, 36(11), 2892–2898.

Algorithm 1: Finding a conflicting SRLG set for a given active path p from node s to node t , adapted from [14].

Data: Network undirected graph $G = (N, A)$, a path p from s to t , c cost of the links, R set of risks, and set I of the current problem.

Result: T , the conflicting SRLG set for a given active path p

```

1  $T \leftarrow \emptyset$  // Empty set of SRLGs
2  $L \leftarrow \emptyset$  // Empty list of paths
3 insert ( $L, p$ )
4  $X \leftarrow$  set of SRLGs that affect  $p$ 
5  $X \leftarrow X \setminus I$  // excluding SRLGs in  $I$ 
6  $stop \leftarrow false$ 
7 while  $\neg \text{empty}(X) \wedge \neg stop$  do
8    $g \leftarrow \text{erase}(X)$ 
9   insert ( $T, g$ )
10  Remove the links affected by  $g$  from the network
11   $p_g \leftarrow$  Shortest path from  $s$  to  $t$ 
12  if  $c(p_g) \neq \infty$  then
13    insert ( $L, p_g$ )
14     $X_{p_g} \leftarrow$  Set of SRLGs that affect  $p_g$ 
15     $X \leftarrow X \cap X_{p_g}$  // for reducing  $T$ 
16  end
17  else
18     $stop \leftarrow true$  //  $T$  is calculated
19  end
20 end

```

5. Dijkstra, E. W. (1959) A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271.
6. Eppstein, D. (1999). Finding the k shortest paths. *SIAM J. Comput.*, 28(2), 652–673.
7. Gomes, T., Fernandes, L. (2010). Obtaining a SRLG-disjoint path pair of min-sum cost. In Jacek Rak, David Tipper, and Krzysztof Walkowiak, editors, *RNDM'10 – 2nd International Workshop on Reliable Networks Design and Modeling*, ISBN: 978-I-4244-7283-3, 116–122, Moscow.
8. Gomes, T., Craveirinha, J. (2010). An algorithm for enumerating SRLG diverse path pairs. *Journal of Telecommunications and Information Technology*, (3), 5–12.
9. Gouveia, L., Patrício, P., Sousa, A. (2008). Hop-constrained node survivable network design: An application to MPLS over WDM. *Networks and Spatial Economics*, 8(1), 3–21, .
10. Ho, P.-H., Mouftah, H. T. (2004). Shared protection in mesh WDM networks. *IEEE Communications Magazine*, 42(1), 70–76.

11. Hu, J. Q. (2003). Diverse routing in optical mesh networks. *IEEE Transactions on Communications*, 51(3), 489–494.
12. Kompella, K., Rekhter, Y. (2005). OSPF extensions in support of generalized multi-protocol label switching (GMPLS). IETF RFC 4203.
13. Laborczy, P., Tapolcai, J., Ho, P.-H., Cinkler, T., Recski, A., Moutah, H. T. (2001). Algorithms for asymmetrically weighted pair of disjoint paths in survivable networks. In T. Cinkler, editor, *Proceedings of Design of Reliable Communication Networks (DCRN 2001)*, 220–227. Hungary.
14. Rostami, M. J., Khorsandi, S., Khodaparast, A. A. (2007). CoSE: A SRLG-disjoint routing algorithm. In *Proceedings of the Fourth European Conference on Universal Multiservice Networks (ECUMN'07)*, Toulouse, France.
15. Martins, E., 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(2), 121–134.
16. Martins, E., Pascoal, M., Santos, J. (1999). An algorithm for ranking loopless paths. Technical Report 99/007, CISUC.
17. Martins, E., Pascoal, M., Santos, J. (1999). Deviation algorithms for ranking shortest paths. *International Journal of Foundations of Computer Science*, 10(3), 247–263.
18. Pan, P., Swallow, G., Atlas, A. (2005). Fast reroute extensions to RSVP-TE for LSP tunnels. IETF RFC 4090.
19. Pan, X., Xiao, G. (2004). Algorithms for the diverse routing problem in WDM networks with shared risk link groups. In *International Conference on Computational Science 2004 (ICCS 2004)*, 381–385, Krakow, Poland.
20. Pan, X., Xiao, G. (2006). Heuristics for diverse routing in wavelength-routed networks with shared risk link groups. *Photonic Network Communications*, 11(1), 29–38.
21. Suurballe, J. W., Tarjan, R. E. (1984) A quick method for finding shortest pairs of disjoint paths. *Networks*, 14(2), 325–336.
22. Todimala, A., Ramamurthy, B. (2004). IMSH: An iterative heuristic for SRLG diverse routing in WDM mesh networks. In *13th International Conference on Computer Communications and Networks, ICCCN'2004*, 199–204.
23. Todimala, A., Ramamurthy, B. (2005). A heuristic with bounded guarantee to compute diverse paths under shared protection in WDM mesh networks. In *IEEE Globecom 2005, November 28 - December 2, 2005, St. Louis, MO, USA*, 1915–1919.
24. Xu, D., Chen, Y., Xiong, Y., Qiao, C., He, X. (2004). On finding disjoint paths in single and dual link cost networks. In *IEEE INFOCOM 2004*, Hong Kong.
25. Xu, D., Xiong, Y., Qiao, C., Li, G. (2003). Trap avoidance and protection schemes in networks with shared risk link groups. *Journal of Lightwave Technology*, 21(11), 2683–2693.
26. Yen, J. Y. (1971). Finding the k shortest loopless paths in a network. *Management Science*, 17(11), 712–716.



Teresa Gomes is Assistant Professor in telecommunications at the Department of Electrical and Computer Engineering of the Faculty of Sciences and Technology of the University of Coimbra, Portugal, since 1998, and a researcher at the INESC-Coimbra R&D Institute. She obtained the following degrees: undergraduate diploma in electrical engineering science (E.E.S.)-informatics (1984), M.Sc. in computer science (1989) and Ph.D. in E.E.S.-telecommunications and electronics (1998), all at the University of Coimbra. Her main present interests are routing, protection and reliability analysis models and algorithms for optical and MPLS networks.



Carlos Simões obtained an undergraduate diploma in Electrical Engineering Science-Telecommunications & Electronics in 1995 and a M.Sc. degree in Systems and Automation-Telecommunications in 1999, both from the University of Coimbra, Portugal. Since 1998, he has been an Assistant at the Polytechnic Institute of Viseu, Portugal, and a researcher at INESC-Coimbra R&D Institute. He is currently working towards his Ph.D. degree in Electrical Engineering at the University of Coimbra and is involved in the research of multicriteria models for routing in optical networks.



Luís Fernandes received his Master's Degree in Electrical and Computer Engineering (with specialization in the area of Telecommunications) by the University of Coimbra on September of 2010. From September 2008 until February 2009 he was an Erasmus student at Warsaw University of Technology, Poland. From February 2010 until July 2010 he was the recipient of a research scholarship at INESC Coimbra R&D Institute, where he carried out work related to SRLG diverse routing. His areas of interest are path protection schemes in telecommunication networks.