

Design/Upgrade of a Transparent Optical Network Topology Resilient to the Simultaneous Failure of its Critical Nodes

Fábio Barbosa¹ | Amaro de Sousa¹ | Agostinho Agra²

¹*Instituto de Telecomunicações, DETI
Universidade de Aveiro, Portugal*

²*CIDMA/IT, Dept. Matemática, Universidade
de Aveiro, Portugal*

Correspondence

Fábio Barbosa, Instituto de
Telecomunicações, Campus Universitário,
3810-193 Aveiro, Portugal
Email: fabiobarbosa@av.it.pt

Funding information

Paper based upon work from COST Action CA15127 ("Resilient communication services protecting end-user applications from disaster-based failures - RECODIS") supported by COST Association. Work funded by FCT, Portugal, through project CENTRO-01-0145-FEDER-029312. Work also funded by FCT/MCTES through national funds and when applicable co-funded EU funds under the project UIDB/EEA/50008/2020. Fábio Barbosa funded by FCT through PhD grant SFRH/BD/132650/2017.

This paper addresses two related problems in the context of transparent optical networks. In the network design problem, the aim is to identify a set of fiber links to connect a given set of nodes. In the network upgrade problem, the aim is to identify a set of new fiber links to add to a given network topology. For a given fiber length budget, the aim in both problems is to maximize the network resilience to the simultaneous failure of its critical nodes. The resilience is evaluated by the Average 2-Terminal Reliability (A2TR) against a set of critical node failures and the critical nodes are the ones that minimize the A2TR of the network. So, the design/upgrade problem is a bi-level max-min optimization problem. Recently, a multi-start greedy randomized heuristic was proposed for both problems. Here, we propose an alternative method based on a greedy deterministic algorithm and we provide computational results showing that the new method obtains better solutions. The results show that the resiliency difference between existing network topologies and the best network design solutions is very high but this difference can be significantly reduced by network upgrades with small fiber length budgets.

KEYWORDS

Transparent Optical Networks, Critical Node Detection, Resilient Network Design, Disasters, Optimization, Heuristics

1 | INTRODUCTION

Large-scale failures can seriously disrupt a telecommunications network due to either natural, technological or malicious human activities [14]. Two recent surveys conducted within COST Action RECODIS are [10] on strategies to protect networks against large-scale natural disasters and [9] on security challenges in communication networks. When dealing with large-scale failures, it is important not only to recover from failures as quick as possible (the post-disaster problem), but also to prepare the network to minimize the impact of such failures (the pre-disaster problem).

This work deals with the pre-disaster problem by addressing the design/upgrade of telecommunication networks aiming to enhance their resilience to large-scale failures. To reach this goal, we first adopt a proper network resiliency metric and, then, we propose design methods aiming to optimize the network resiliency metric. We address the design of resilient network topologies in the context of transparent optical networks. Note that, in general, multiple failures might involve only links or nodes and links (a node failure implies that its links also fail). For example, in malicious human attacks, node shutdowns are harder to realize but they are the most rewarding in the attackers' perspective since the shutdown of a single node also shuts down its incoming/outgoing fiber links. Moreover, power outages can only shut down nodes since fiber links do not require power supply. Here, we consider as large-scale failures the case of multiple node failures as they are the most harmful cases.

For a given network topology, if some nodes are considered critical due to some reason, the network design should take it into consideration, as in [3] where the approach proposed in [1] is adapted to the design of a transparent optical network minimizing the failure impact of a given set of critical nodes. Here, we consider that the resilience of a network topology is evaluated by the Average 2-Terminal Reliability (A2TR) against a set of critical node failures. The A2TR metric is defined as the number of node pairs that remain connected if all critical nodes fail. The critical nodes of a network are the nodes that minimize the A2TR of the network, an optimization problem commonly named Critical Node Detection (CND) problem. So, the design problem is a bi-level max-min optimization problem.

CND problems have been considered in different contexts and are gaining special attention in the vulnerability evaluation of telecommunication networks to large-scale failures [10]. In [2], the CND problem is defined as the detection of a given number c of critical nodes aiming to minimize the number of connected node pairs. Recently, this and other variants of CND have been addressed [7, 18, 21, 22], but none of these works addresses the CND problem in the context of transparent optical networks. Other metrics have been used to evaluate the vulnerability of networks in other contexts [16] or assuming multiple geographically correlated failures [12]. There are also works on improving the preparedness of networks to multiple failures, some by changing the network topology [5, 11, 23], while others by proposing strategies to recover from failures [8, 17]. None of these works, though, uses the optimal solution of the CND problem to assess the vulnerability of networks. On the other hand, CND was used in [6] but resiliency improvement is exploited by optimal robust node selection on a given network topology. The advantage of using CND is that it provides a worst case resiliency analysis, i.e., in any failure involving the same number of failing nodes, the resulting A2TR is never worse than the value provided by the CND optimal solution.

In transparent optical networks, data is converted into light in the source node and transmitted through an all optical path, named *lightpath*, towards the destination node. Due to many optical degradation factors, like attenuation, dispersion, crosstalk and other non-linear factors, there is a maximum length, named *transparent reach*, for each lightpath to work properly. If a lightpath is required on a path whose length is higher than the transparent reach, regenerators must be placed at intermediate nodes to convert the optical signal into the electrical domain, regenerate the signal and reconvert it back to the optical domain (when regenerators are used, the network is referred to as a translucent network). Nevertheless, the use of regenerators is expensive and puts an additional burden on the network management and, so, they are avoided when possible. The design of translucent networks must take into

account the cost imposed by the required regenerators, which is out of the scope of this work. The methods proposed in this work are applicable to transparent optical networks, i.e., optical networks whose diameter (the highest optical length among the shortest paths of all node pairs) is not higher than the transparent reach. Note that the optical length of a path depends both on the length of its links and on its number of hops, i.e., number of intermediate nodes). We model the optical degradation suffered by a lightpath while traversing an intermediate node as a fiber length value Δ , i.e., by considering it equivalent to the degradation incurred due to the transmission over a fiber of length Δ . So, when accounting for the A2TR metric, the CND problem has to consider that two nodes are connected only if the surviving network provides them with at least one path whose optical length is within the required transparent reach.

In [4], a multi-start greedy randomized method was proposed to generate network topologies, with a given fiber length budget, that are resilient to critical node failures. The method is also adapted in [4] to the upgrade of an existing network topology. Here, we propose an alternative method for the same network design/upgrade problem based on a greedy deterministic algorithm and provide computational results showing that the new method obtains better solutions than the one proposed in [4]. With the updated results, we review the conclusions taken in [4] concerning the resiliency values obtained between the network design and the network upgrade solutions. The computational results will show that the resiliency difference between existing topologies and the best network design solutions is very high but this difference can be significantly reduced by network upgrades with small fiber length budgets.

The paper is organized as follows. Section 2 describes a path-based Mixed Integer Linear Programming (MILP) model defining the CND problem in the context of transparent optical networks and a row generation algorithm that is used to solve the problem. Section 3 proposes deterministic algorithms to generate network topologies resilient to the simultaneous failure of their critical nodes. The computational results are presented and discussed in Section 4. Finally, Section 5 presents the main conclusions of the work.

2 | CRITICAL NODE DETECTION PROBLEM

Consider a transparent optical network represented by an undirected graph $G = (N, E)$ where $N = \{1, \dots, n\}$ is the set of nodes and $E \subseteq \{(i, j) \in N \times N : i < j\}$ is the set of fiber links. For each link $(i, j) \in E$, parameter l_{ij} represents its length. The transparent reach of the network is denoted by parameter $T > 0$ and the fiber length equivalent to the degradation suffered by a lightpath while traversing an intermediate node is denoted by parameter $\Delta > 0$. We assume that $l_{ij} \leq T$ for all $(i, j) \in E$; otherwise, such link is worthless and can be removed from G .

The set of all paths in G between $i \in N$ and $j \in N$ (with $i < j$ and $(i, j) \notin E$) with length not greater than T is denoted by P_{ij} . Each path $p \in P_{ij}$ is defined by the binary parameters β_k^p , indicating whether node $k \in N$ (which can be an end node) is in p or not, and α_{kt}^p indicating whether link $(k, t) \in E$ is in p or not. So, P_{ij} is composed by all paths p such that
$$\sum_{(k,t) \in E} \alpha_{kt}^p l_{kt} + \Delta \left(\sum_{k \in N} \beta_k^p - 2 \right) \leq T.$$

To model the CND problem, we consider for each node $i \in N$ a binary variable v_i indicating whether i is a critical node or not. We consider also for each node pair (i, j) , with $i, j \in N : i < j$, a binary variable u_{ij} which is 1 if nodes i and j can be connected through a path satisfying the transparent reach T , and 0 otherwise. Then, for a given number c of critical nodes, a path-based formulation for the CND problem is given by the following Integer Linear Programming (ILP) model:

$$\min z := \sum_{i=1}^{n-1} \sum_{j=i+1}^n u_{ij} \quad (1)$$

$$\text{s.t. } \sum_{i=1}^n v_i \leq c, \quad (2)$$

$$u_{ij} + v_i + v_j \geq 1, \quad (i, j) \in E, \quad (3)$$

$$u_{ij} + \sum_{k=1}^n \beta_k^p v_k \geq 1, \quad (i, j) \notin E, p \in P_{ij}, \quad (4)$$

$$v_i \in \{0, 1\}, \quad i \in N, \quad (5)$$

$$u_{ij} \in \{0, 1\}, \quad i, j \in N : i < j. \quad (6)$$

The objective function (1) value z is the A2TR value defined as the total number of connected node pairs in the surviving graph (i.e., the graph given by removing all critical nodes from G). Constraint (2) ensures that at most c nodes are selected as critical nodes (in any optimal solution, c critical nodes are selected). Constraints (3) guarantee that a pair of adjacent nodes (i.e., with a direct link between them) is connected if none of the two nodes is a critical node. Constraints (4) are the generalization of constraints (3) for the node pairs that are not adjacent in G : node pair (i, j) is connected if there is one path $p \in P_{ij}$ such that none of its nodes is a critical node. Constraints (5)-(6) are the variable domain constraints. Note that, since variables v_i are binary, constrains (3) and (4) impose $u_{ij} \geq 1$ when nodes i and j are connected, which then, due to the objective function, forces $u_{ij} = 1$. Therefore, constraints (6) can be replaced by $u_{ij} \geq 0$. The resulting MILP model will be considered henceforward and is referred to as the *exact CND model*.

The total number of constraints (4) depends on the graph topology, the link lengths and the values of T and Δ . However, the exact CND model becomes too large (i.e., with too many constraints) for relatively small sized instances which does not allow solution by any available solver for reasonable sized instances. Instead, a row generation approach can be used to solve the exact CND model as described in Algorithm 1.

Algorithm 1 Exact algorithm for the CND problem

- 1: Initialize and solve **MILP'** model (1)-(6) without constraints (4). Let (u^*, v^*) be the optimal solution
 - 2: **repeat**
 - 3: Set NCuts $\leftarrow 0$ and $K \leftarrow \{i \in N : v_i^* = 1\}$
 - 4: Compute subgraph $G_K = (N \setminus K, E_K)$, where $E_K = \{(i, j) \in E : i, j \notin K\}$
 - 5: **for all** node pair $(i, j) \notin E_K$ **do**
 - 6: Compute shortest path $p \in P_{ij}$ and its optical length d
 - 7: **if** $d \leq T$ **and** $u_{ij}^* + \sum_{k=1}^n \beta_k^p v_k^* = 0$ **then**
 - 8: Add to **MILP'** constraint (4) corresponding to path p
 - 9: NCuts \leftarrow NCuts + 1
 - 10: **end if**
 - 11: **end for**
 - 12: **if** NCuts > 0 **then**
 - 13: Solve **MILP'** model with the added constraints. Update (u^*, v^*) accordingly
 - 14: **end if**
 - 15: **until** NCuts = 0
-

In line 1, a **MILP'** model given by the exact **CND** model without constraints (4) is initialized and solved. Then, in the main cycle (lines 2–15), the separation problem associated with constraints (4) is solved (lines 3–11) where the identified violated constraints (whose number is accounted in **NCuts**) are added to **MILP'** (lines 7–10) and, finally, **MILP'** is solved again (lines 12–14). The algorithm ends when no violated constraint is found (line 15) and the optimal solution is the solution of the last solved **MILP'** model.

The separation problem associated with constraints (4) is solved as follows. In line 4, the subgraph $G_K = (N \setminus K, E_K)$ is computed by removing from G the critical nodes of set K (determined in line 3) and the corresponding incident edges and adding Δ to the length of each edge in E_K . Since the number of intermediate nodes of a path is equal to the number of edges minus one, the shortest path value in G_K is equal to the optical path length plus Δ . So, between each pair of nodes i and j in $N \setminus K$, such that $(i, j) \notin E_K$ (line 5), the shortest path p in G_K is determined (by Dijkstra algorithm) and its optical length d computed as the length of p minus Δ (line 6) and the violation of the constraint (4) associated to p is checked (line 7).

Note that if the imposition of the transparent reach T is relaxed (i.e., considering $T \rightarrow +\infty$), we get one of the classical **CND** problem variants for which there are known efficient compact **MILP** models. One such model, proposed in [18], is as follows. Consider for each pair of nodes (i, j) the set $N_{ij} \subset N$ which represents the set of adjacent nodes to i (on graph G) if the node degree of i is not higher than the node degree of j , or the set of adjacent nodes to j , otherwise. Then, the previous constraints (4) can be replaced by the following (polynomial sized) constraints:

$$u_{ij} \geq u_{\{ik\}} + u_{\{kj\}} - 1 + v_k, (i, j) \notin E, k \in N_{ij}. \quad (7)$$

In these constraints, $u_{\{ik\}}$ represents variable u_{ik} if $i < k$ or variable u_{ki} otherwise (the same meaning to $u_{\{kj\}}$). Constraints (7) guarantee that for each pair of nodes i and j not adjacent in G , they are connected if there is a non-critical node $k \in N_{ij}$ connected to both i and j . The resulting **MILP** model, i.e., replacing constraints (4) by constraints (7), is referred to as the *compact **CND** model*.

Note that the optimal solution value of the compact **CND** model is an upper bound on the optimal solution value of the exact **CND** model since all node pairs that are connected are accounted in the objective function of the compact **CND** model while the ones whose shortest path length over the surviving network is higher than the transparent reach T are not accounted in the exact problem. Nevertheless, in most cases, the optimal solution of the compact **CND** model is also the optimal solution of the exact problem and its resolution is much quicker when using a standard solver (it does not require the row generation method of Algorithm 1). As will be described in the next section, we use this fact to derive computationally efficient algorithms for the network design/upgrade problem.

3 | NETWORK DESIGN/UPGRADE PROBLEM

Consider an existing network $G = (N, E)$ with a total fiber length L and a fiber length budget $B = L + L'$, with $L' \geq 0$. The Network Design Problem aims to identify a new network topology connecting all nodes in N whose total fiber length is not higher than B . The Network Upgrade Problem aims to identify a set of fiber links within the budget L' to be added to the existing topology. In both cases, the aim is to obtain a network (design or upgrade) topology that maximizes the **A2TR** value of the simultaneous failure of its critical nodes.

In [4], a multi-start greedy randomized algorithm was proposed for this network design/upgrade problem. The method randomly generates multiple network topologies, with a fiber length budget given by B . Each topology is generated by a greedy randomized algorithm that builds a network by randomly selecting one link at a time until no

new link can be added within the fiber budget B . In that stochastic method, the evaluation of each network topology uses centrality based heuristics in a preliminary evaluation and, if necessary, Algorithm 1 to compute its exact A2TR value. At the end, the method outputs the best generated topology, i.e., the one with highest A2TR value. Here, we propose an alternative deterministic method based on a greedy approach. The main differences when compared with the previous approach are that a single greedy solution is generated and, on each greedy step, the selected link is based on the critical nodes of the current partial topology (resulting from all already selected links). As a consequence, the critical nodes must be computed at each step of the greedy algorithm.

The proposed method is composed by three tasks which are run in sequence. In the first task, a greedy deterministic algorithm is run so that a topology solution is computed. In the second task, a local search method is applied to the previous solution to try to find a better one. Both first and second tasks consider the network A2TR evaluation imposed by the critical nodes based on the compact CND model. Finally, the third task evaluates the previous solution in terms of optical transparency and using the exact CND model. If the previous solution is not optically transparent and/or the evaluation provided by the exact CND model is lower than the evaluation value provided by the compact CND model, the third task uses the unused budget to add new links so that the network topology becomes optically transparent and the exact A2TR value becomes as close as possible to the value provided by the compact CND. Next, we describe the tasks in three separate subsections and, then, describe the overall algorithms in the fourth subsection.

3.1 | Greedy deterministic approach

In the first task, both network design and upgrade problems are conceptually modeled as the upgrade of a network topology represented by graph $G = (N, E)$ where $E = \{\}$ in the network design problem and E is the set of fiber links of an existing network in the network upgrade problem. In both cases, parameter l_{ij} , with $i < j$, represents the length of the fiber link between nodes i and j (either an existing link or a possible new link).

Consider the following notation. For a general graph $G = (N, \mathcal{E})$ and a given set of critical nodes $K \subset N$, consider the surviving graph given by $G_K = (N \setminus K, \mathcal{E}_K)$, with $\mathcal{E}_K = \{(i, j) \in \mathcal{E} : i, j \notin K\}$. The surviving graph G_K might be composed by different connected components where a node in one component has no connectivity to a node of any other component. So, for a given set of critical nodes $K \subset N$, parameter m_K indicates the number of connected components of the surviving graph G_K and the binary parameters $cp_K(i, j)$ indicate that nodes $i, j \in N \setminus K$ are in different components of G_K (if $cp_K(i, j) = 1$) or in the same component (if $cp_K(i, j) = 0$).

Algorithm 2 presents a greedy deterministic algorithm to upgrade a network represented by graph $G = (N, E)$ with a fiber budget B . This algorithm iteratively selects a candidate link among the ones whose end-nodes belong to two different components in the current surviving network topology, i.e., the topology resulting by removing its critical nodes from the current partial topology.

Algorithm 2 has four input parameters (line 1): besides the graph G and the fiber budget B , the algorithm has a third integer parameter c representing the number of critical nodes of interest and a fourth parameter which is an input set of links E' considered as follows. In the network upgrade problem, we consider the input link set $E' = \{\}$. In the network design problem, we follow [4] considering the input link set E' given by the Relative Neighbourhood Graph (RNG) [20], which is defined as follows: nodes $i, j \in N$ are connected by a link if and only if there is no other node $k \in N \setminus \{i, j\}$ such that $l_{ik} \leq l_{ij}$ and $l_{jk} \leq l_{ij}$. The preliminary tests have shown that this set of links provides a good initial balance between connectivity and amount of used fiber budget in the network design problem.

In line 2, some relevant variables are initialized: \bar{z} which represents the CND optimal value of the current partial topology; B_R which represents the amount of fiber budget still available; and set \mathcal{K} which includes all computed sets of critical nodes. Algorithm 2 is an iterative process (lines 3-18) where each iteration is composed of two phases: an

Algorithm 2 Greedy deterministic algorithm

```

1: INPUT:  $G = (N, E)$ ,  $B$ ,  $c$ ,  $E'$ 
2:  $\bar{z} \leftarrow 0$ ,  $B_R \leftarrow B$  and  $\mathcal{K} \leftarrow \{\}$ 
3: repeat
4:    $E'_{\text{add}} \leftarrow \{\}$  and  $E'_{\text{rem}} \leftarrow \{\}$ 
5:   repeat
6:     Solve compact CND model for graph  $G \leftarrow (N, E \cup E')$ 
7:     Let  $z^*$  be the optimal value and  $(u^*, v^*)$  be the optimal solution
8:      $\bar{z} \leftarrow z^*$ ,  $K \leftarrow \{i \in N : v_i^* = 1\}$  and  $\mathcal{K} \leftarrow \mathcal{K} \cup \{K\}$ 
9:     Compute the  $m_K$  components of graph  $(N \setminus K, (E \cup E')_K)$ 
10:    if  $m_K \geq 2$  and  $\min_{(i,j)} \{l_{ij} : cp_K(i,j) = 1\} \leq B_R$  then
11:      Compute new link  $(i, j)$  corresponding to  $\min_{(i,j)} \{f(i, j) : cp_K(i, j) = 1, l_{ij} \leq B_R\}$ 
12:       $E' \leftarrow E' \cup \{(i, j)\}$ ,  $E'_{\text{add}} \leftarrow E'_{\text{add}} \cup \{(i, j)\}$  and  $B_R \leftarrow B_R - l_{ij}$ 
13:    end if
14:    until no new link added to set  $E'_{\text{add}}$ 
15:    if  $E'_{\text{add}} \neq \{\}$  then
16:      [Removing Phase]
17:    end if
18: until  $E'_{\text{add}} = \{\}$  or  $E'_{\text{rem}} = \{\}$  or  $E'_{\text{rem}} = \{\text{last } |E'_{\text{rem}}| \text{ links added to set } E'_{\text{add}}\}$ 

```

Adding Phase, where links are added to set E' in order to improve the CND value of the resulting topology until the available budget is not enough to add more links; and a **Removing Phase**, where all links in E' are reevaluated and removed if their elimination does not decrease the CND optimal value of the resulting topology. In each iteration, set E'_{add} (and set E'_{rem}) represents the set of links added to (and removed from) the network in the current iteration. These sets are initialized empty at the beginning of each iteration (line 4).

In the Adding Phase (lines 5-14), the compact CND model is solved for the current partial topology $(N, E \cup E')$ (lines 6-7), its set of critical nodes is stored in set K (line 8) and the set of network components imposed by the critical node set K is determined (line 9). In line 10, if the number of components m_K is one (the network is fully connected to any c node failures) or if the shortest available link connecting two components is larger than the remaining budget, no new link is added and the Adding Phase ends. Otherwise, the link $(i, j) \notin E \cup E'$ with the smallest value of a given function $f(i, j)$ that connects two nodes belonging to different components is selected (line 11) and the selected link is added to the current partial topology (line 12). Function $f(i, j)$ can be one of the two following possibilities:

- (i) $f_1(i, j) = l_{ij}$, i.e., the link length;
- (ii) $f_2(i, j) = l_{ij} \times (\delta_i + \delta_j)$, where δ_i and δ_j are the degree of nodes i and j in the current partial topology $(N, E \cup E')$.

The Adding Phase is repeated until no new link is selected (line 14). In lines 15-17, if at least one link was added in the Adding Phase, the Removing Phase (described in Algorithm 3) is run. Algorithm 2 runs until no links are added in the Adding Phase, or no links are removed in the Removing Phase or the last links added in the Adding Phase are the links removed in the Removing Phase (line 18).

The Removing Phase (Algorithm 3) is an iterative process (lines 2-14) that evaluates each link $(i, j) \in E'$ in decreasing order of its length l_{ij} . For each $(i, j) \in E'$, first, the algorithm computes (in line 3) graph G' that represents the upgraded network without link (i, j) . Then, the compact CND of G' is solved (line 8) and if its CND value z' is

Algorithm 3 Removing Phase of Algorithm 2

```

1: INPUT: Set of selected candidate links  $E'$ , ordered from the longest to the shortest
2: for all  $(i, j) \in E'$  do
3:    $G' \leftarrow (N, (E \cup E') \setminus \{(i, j)\})$ 
4:   for all  $K \in \mathcal{K}$  do
5:      $z'_K \leftarrow$  resiliency value given by set of critical nodes  $K$  in graph  $G'$ 
6:   end for
7:   if  $\min_{K \in \mathcal{K}} \{z'_K\} \geq \bar{z}$  then
8:     Solve compact CND model for graph  $G'$  and set  $z'$  as the optimal value
9:      $K \leftarrow \{i \in N : v_i^* = 1\}$  and  $\mathcal{K} \leftarrow \mathcal{K} \cup \{K\}$ 
10:    if  $z' \geq \bar{z}$  then
11:       $E' \leftarrow E' \setminus \{(i, j)\}$ ,  $E'_{\text{rem}} \leftarrow E'_{\text{rem}} \cup \{(i, j)\}$  and  $B_R \leftarrow B_R + l_{ij}$ 
12:    end if
13:  end if
14: end for

```

not lower than the current resiliency value \bar{z} (line 10), the link (i, j) is removed from E' since it does not degrade the resilience of the current topology (line 11).

Our preliminary tests showed that most of the CND optimal solutions in this phase overlap with previous solutions (stored in set \mathcal{K} , line 8 of Algorithm 2). So, in order to improve the computational efficiency of the Removing Phase, in lines 4-6 of Algorithm 3, the resiliency value z'_K is computed for each critical node set $K \in \mathcal{K}$ in graph G' (i.e., the number of connected node pairs in graph G' without nodes K). Then, if the minimum of these values is not lower than the current resiliency value (line 7), the algorithm needs to solve the compact CND model. Otherwise, we know that the current link (i, j) cannot be removed and, therefore, we do not need to solve the compact CND model for graph G' (line 8) which is the most time consuming part of the algorithm.

Figure 1 is an illustration of Algorithm 2 for a graph with 9 nodes shown in (a), a fiber budget $L' = 15\%L$, $f(i, j) = f_1(i, j)$ and $c = 2$ critical nodes. Initially, the compact CND model is solved for the initial graph to compute its critical nodes, highlighted in red in (a). By removing the critical nodes, it results in the surviving network in (b) with $m_K = 2$ network components (with 3 and 4 nodes, respectively) and the selected link is the shortest one (recall that $f_1(i, j) = l_{ij}$) that connects both components, highlighted in dashed blue in (b). This link is added to the topology, resulting in the upgraded graph in (c), and the optimal critical node set is recomputed. This process is repeated once more, obtaining the upgraded graph in (e). Then, by removing its critical nodes from the graph, there is no candidate link to be added within the remaining fiber budget. At this stage, the Removing Phase starts and the first added link, highlighted in dashed red in (f), is now removed because the resulting topology in (g) has the same resiliency value as the one in (e). This removal increases the available fiber budget B_R . The Adding Phase runs again which now adds a new link, highlighted in dashed blue in (h), obtaining the topology in (i). In this example, this last topology has maximal resiliency, i.e., by removing its critical set of 2 nodes, the surviving network in (j) is fully connected.

In Algorithm 2, each added link takes into account the set of critical nodes given by the optimal solution of the compact CND model. A potentially better algorithm is, for each partial topology, to consider the S best critical node sets, with $S > 1$, as in Algorithm 4. Algorithm 4 has many similarities with Algorithm 2: lines 2-10 are similar and lines 25-31 in Algorithm 4 are equal to lines 12-18 in Algorithm 2 (which means that the Removing Phase is also equal).

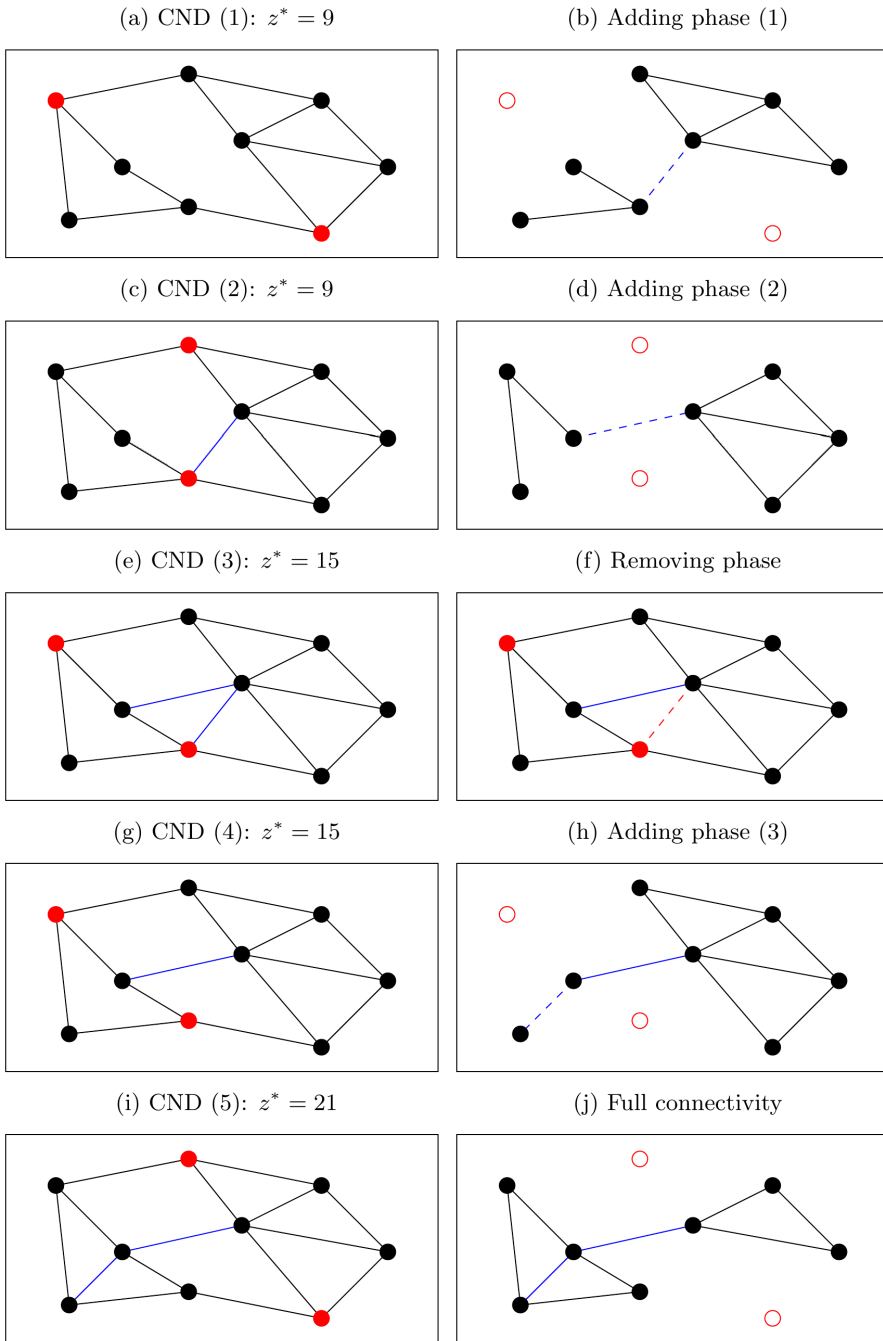


FIGURE 1 Example of Algorithm 2, considering $c = 2$, $f(i, j) = f_1(i, j)$, $L' = 15\%L$, the initial graph in (a). On left, graph $(N, E \cup E')$ in each iteration, with the critical nodes in red, links of E in black and links of E' in blue. On right, the surviving graph in each iteration, added link in dashed blue and removed link in dashed red.

Algorithm 4 Greedy deterministic algorithm, considering S CND solutions

```

1: INPUT:  $G = (N, E)$ ,  $B$ ,  $c$ ,  $E'$ ,  $S$ 
2:  $\bar{z} \leftarrow 0$ ,  $B_R \leftarrow B$  and  $\mathcal{K} \leftarrow \{\}$ 
3: repeat
4:    $E'_{\text{add}} \leftarrow \{\}$  and  $E'_{\text{rem}} \leftarrow \{\}$ 
5:   repeat
6:     Initialize and solve compact CND model for graph  $G \leftarrow (N, E \cup E')$ 
7:     Let  $z^*$  be the optimal value and  $(u^*, v^*)$  be the optimal solution. Set  $\bar{z} \leftarrow z^*$ 
8:      $K \leftarrow \{i \in N : v_i^* = 1\}$  and  $\mathcal{K} \leftarrow \mathcal{K} \cup \{K\}$ 
9:     Compute the  $m_K$  components of graph  $(N \setminus K, (E \cup E')_K)$ 
10:    if  $m_K \geq 2$  and  $\min_{(i,j)} \{l_{ij} : cp_K(i,j) = 1\} \leq B_R$  then
11:      Initialize  $s \leftarrow 1$  and  $c_{ij} \leftarrow 0$ , for all candidate links  $(i,j) \notin E \cup E'$  such that  $l_{ij} \leq B_R$ 
12:      while  $s \leq S$  and  $m_K \geq 2$  do
13:        for all  $(i,j) \notin E \cup E' : cp_K(i,j) = 1, l_{ij} \leq B_R$  do
14:           $c_{ij} \leftarrow c_{ij} + (\bar{z}/z^*)$ 
15:        end for
16:        if  $s < S$  then
17:          Add constraint  $\sum_{i=1}^n v_i^* v_i \leq \sum_{i=1}^n v_i^* - 1$  to compact CND model of graph  $G$ 
18:          Solve compact CND model. Let  $z^*$  be the optimal value and  $(u^*, v^*)$  be the optimal solution
19:           $K \leftarrow \{i \in N : v_i^* = 1\}$  and  $\mathcal{K} \leftarrow \mathcal{K} \cup \{K\}$ 
20:          Compute the  $m_K$  components of graph  $(N \setminus K, (E \cup E')_K)$ 
21:        end if
22:         $s \leftarrow s + 1$ 
23:      end while
24:      Compute new link  $(i,j) \notin E'$  corresponding to  $\max_{(i,j)} \{c_{ij}/f(i,j) : l_{ij} \leq B_R\}$ 
25:       $E' \leftarrow E' \cup \{(i,j)\}$ ,  $E'_{\text{add}} \leftarrow E'_{\text{add}} \cup \{(i,j)\}$  and  $B_R \leftarrow B_R - l_{ij}$ 
26:    end if
27:  until no new link added to set  $E'_{\text{add}}$ 
28:  if  $E'_{\text{add}} \neq \{\}$  then
29:    [Removing Phase]
30:  end if
31: until  $E'_{\text{add}} = \{\}$  or  $E'_{\text{rem}} = \{\}$  or  $E'_{\text{rem}} = \{\text{last } |E'_{\text{rem}}| \text{ links added to set } E'_{\text{add}}\}$ 

```

In line 11 of Algorithm 4, a set of variables c_{ij} are initialized to zero. These variables count, in a weighted manner, the number of times that each candidate link $(i,j) \notin E \cup E'$ connects two components in the S alternative surviving graphs, i.e., the surviving graphs of the S best CND solutions. Note that in line 7, \bar{z} is set with the solution value of the first compact CND (the lowest value among all S surviving graphs). Then, for each candidate link (i,j) that connects two components of the current surviving graph (line 13), c_{ij} increases by \bar{z}/z^* where z^* is the solution value of the current surviving graph. Note that the ratio \bar{z}/z^* is always less than or equal to 1. So, the idea behind this weighted sum is to give a lower weight in the link selection to the links connecting two components of the surviving graphs whose CND solution values are worst. Finally, in line 24, the link $(i,j) \notin E \cup E'$ with the largest value of $c_{ij}/f(i,j)$ is selected (like before, $f(i,j)$ is set either with $f_1(i,j)$ or with $f_2(i,j)$).

While the best CND solution of each partial topology is computed in line 6 of Algorithm 4, the additional alternative CND solutions are computed in the loop 12-23. This loop ensures that the algorithm will continually compute a compact CND solution until either it loops S times or $m_k = 1$ (line 12), i.e., the current surviving graph does not have multiple components. The condition in line 16 ensures that the compact CND model is optimized at most S times. Finally, given a set of critical nodes K , the added constraint (line 17) excludes this set from the set of feasible solutions of the compact CND model so that, when it is solved again (line 18), it will result in the next best set of critical nodes.

Note that, while computing the S sets of critical nodes for each partial topology, they are computed in increasing value of the A2TR (i.e., the number of connected node pairs). There might exist multiple optimal CND solutions, i.e., all solutions next to the first one such that their optimal value z^* is equal to \bar{z} . Since these solutions are more damaging than the subsequent ones, a meaningful alternative is to consider only the optimal CND solutions and ignore the remaining ones. This is easily done by adding in line 12 of Algorithm 4 a third condition in the form $z^* = \bar{z}$. In the computational results, we also test this algorithm variant.

3.2 | Local search approach

In the second task, a local search algorithm is applied to the solution provided by the first task. Note that in the Removing Phase of Algorithms 2 and 4, a link can only be removed if the resiliency value of the current network topology remains the same. Here, a local search algorithm (described in Algorithm 5) is proposed where each link $(i, j) \in E'$ is removed and excluded from the set of candidate links while re-running either Algorithm 2 or Algorithm 4. Removing the link from the graph decreases the resiliency value \bar{z} but increases the remaining fiber budget B_R , which might enable to find an alternative topology with a better resiliency value.

Algorithm 5 Local search algorithm

```

1: INPUT:  $G = (N, E), E', \bar{z}, B_R$ 
2: repeat
3:    $\bar{z}_{LS} \leftarrow \bar{z}, E_{LS} \leftarrow E'$  and  $B_{LS} \leftarrow B_R$ 
4:   for all  $(i, j) \in E'$  do
5:      $G' \leftarrow (N, (E \cup E') \setminus \{(i, j)\})$ 
6:     Run Algorithm 2 (or 4) starting from set  $E' \setminus \{(i, j)\}$ , fiber budget  $B_R + l_{ij}$  and excluding link  $(i, j)$  from the set of candidate links, obtaining a new set of selected links  $E'_{ij}$ , its resiliency value  $\bar{z}_{ij}$  and fiber budget  $B_{ij}$ 
7:     if  $\bar{z}_{ij} > \bar{z}_{LS}$  or ( $\bar{z}_{ij} = \bar{z}_{LS}$  and  $B_{ij} > B_{LS}$ ) then
8:        $\bar{z}_{LS} \leftarrow \bar{z}_{ij}, E_{LS} \leftarrow E'_{ij}$  and  $B_{LS} \leftarrow B_{ij}$ 
9:     end if
10:  end for
11:  if  $\bar{z}_{LS} > \bar{z}$  then
12:    Update  $\bar{z} \leftarrow \bar{z}_{LS}, E' \leftarrow E_{LS}$  and  $B_R \leftarrow B_{LS}$ 
13:  end if
14: until  $\bar{z}_{LS} \leq \bar{z}$ 

```

The inputs of Algorithm 5 are the original topology $G = (N, E)$, the added links E' , the resiliency value of graph $(N, E \cup E')$ and the remaining fiber budget B_R of the solution provided by the first task. The main loop (steps 2-14) ensures that the local search runs until the resiliency value is not improved. Variables \bar{z}_{LS} , E_{LS} and B_{LS} store the best alternative topology in the present iteration and are accordingly updated (line 8) when a better topology is found. The

inner loop (lines 4-10) removes each link $(i, j) \in E'$ from graph $(N, E \cup E')$ and runs the greedy deterministic algorithm (Algorithm 2 or 4) excluding the removed link to be selected. If two alternative topologies have the same resiliency value, the one with the highest remaining fiber budget is selected (line 7). Finally, in lines 11-13, the main variables \bar{z} , E' and B_R are updated if the best alternative topology has a higher resiliency value than the current topology.

After running some computational tests, we observed that in the network design problem, the local search approach is very inefficient as it does not provide relevant gains in the resiliency value and the running time becomes very high. Therefore, the second task is only included in the overall algorithm of the network upgrade problem.

3.3 | Transparent optical networks application

In the previous tasks, the A2TR value was computed by solving the compact CND model. So, in the network design problem, it is necessary to check if the solution provided by the previous task is optically transparent, i.e., for each node pair, the optical shortest path between them is not higher than the transparent reach T (this is not an issue in the network upgrade problem since the optical transparency is guaranteed by the original network topology). If not, the aim is to use the unused fiber budget to turn the solution optically transparent. Moreover, it is also necessary to compute the A2TR value with the exact CND model and, if the two values are not equal, again we use the unused fiber budget to add new links so that the two values become as close as possible.

In our preliminary tests, we observed that, from the three stopping criteria used in Algorithms 2 and 4, the most common stopping criterion is the last one: $E'_{\text{rem}} = \{\text{last } |E'_{\text{rem}}| \text{ links added to set } E'_{\text{add}}\}$, i.e., the last added links do not improve the A2TR value of the solution and, therefore, they are removed. Thus, the solutions tend to have a reasonable amount of available fiber budget B_R . Algorithm 6 presents a deterministic method to use the remaining budget B_R in order to make an input topology $G = (N, E)$ optically transparent. Initially, the shortest path distances between all node pairs without a direct link (i.e., $(i, j) \notin E$) are computed in loop 3-5. In line 6, variable M is set to the maximum distance between all node pairs $(i, j) \notin E$ and (i_M, j_M) is set to such a node pair. If M exceeds the transparent reach (line 7), the algorithm selects the shortest link (i, j) that, when added to the current topology, turns the distance between (i_M, j_M) within the transparent reach. If the selected link is within the available fiber budget, it is added to the topology (lines 9-11). The algorithm continues until no new link is added (line 13).

Algorithm 6 Network validation algorithm

- 1: **INPUT:** $G = (N, E)$, B_R
 - 2: **repeat**
 - 3: **for all** node pairs $(i, j) \notin E$ **do**
 - 4: Compute shortest path $p_{ij} \in P_{ij}$ and its length d_{ij}
 - 5: **end for**
 - 6: Compute $M \leftarrow \max\{d_{ij} : (i, j) \notin E\}$ and the corresponding node pair (i_M, j_M)
 - 7: **if** $M > T$ **then**
 - 8: Compute link $(i, j) \notin E$ corresponding to $\min_{(i,j)} \{l_{ij} : d_{i_M j_M}^{ij} \leq T\}$, where $d_{i_M j_M}^{ij}$ is the distance between nodes i_M and j_M on graph $(N, E \cup \{(i, j)\})$
 - 9: **if** $l_{ij} \leq B_R$ **then**
 - 10: $E \leftarrow E \cup \{(i, j)\}$ and $B_R \leftarrow B_R - l_{ij}$
 - 11: **end if**
 - 12: **end if**
 - 13: **until** no new link added to set E
-

Algorithm 6 is run when we aim to design a new topology based on the fiber budget of an existing optical network and the existing optical network is not 2-connected (in the context of transparent optical networks, a topology is 2-connected if it is optically transparent for every single node deletion). On the other hand, when the existing network topology is 2-connected, we also require the solution obtained by the network design problem to be 2-connected. Algorithm 7 is a generalization of Algorithm 6 aiming to use the remaining budget to turn an input topology $G = (N, E)$ into a 2-connected topology. In Algorithm 7, instead of computing the distance between all node pairs without a direct link in G , these distances are computed in loop 3-8 for all reduced graphs G_k , i.e., graphs without node k and its links, for all nodes $k \in N$. Then, variable M is computed (line 9) with the maximum distance between all node pairs over all reduced graphs G_k and the selected link (i, j) is computed in a way similar to Algorithm 6 but now considering the reduced graph G_k over which the maximum distance M was computed.

Algorithm 7 Network validation algorithm (2-connected)

```

1: INPUT:  $G = (N, E)$ ,  $B_R$ 
2: repeat
3:   for all  $k \in N$  do
4:      $G_k \leftarrow (N \setminus \{k\}, E_k)$ , with  $E_k = \{(i, j) \in E : i, j \neq k\}$ 
5:     for all node pairs  $(i, j) \notin E_k$  with  $i, j \in N \setminus \{k\}$  do
6:       Compute shortest path  $p_{ij}^k$  (on graph  $G_k$ ) and its length  $d_{ij}^k$ 
7:     end for
8:   end for
9:   Compute  $M \leftarrow \max\{d_{ij}^k : (i, j) \notin E_k, k \in N\}$ , the corresponding node pair  $(i_M, j_M)$  and removed node  $k_M$ 
10:  if  $M > T$  then
11:    Compute new link  $(i, j) \notin E_{k_M}$  corresponding to  $\min_{(i,j)} \{l_{ij} : d_{i_M j_M}^{k_M, ij} \leq T\}$ , where  $d_{i_M j_M}^{k_M, ij}$  is the distance
    between nodes  $i_M$  and  $j_M$  on graph  $(N \setminus \{k_M\}, E_{k_M} \cup \{(i, j)\})$ 
12:    if  $l_{ij} \leq B_R$  then
13:       $E \leftarrow E \cup \{(i, j)\}$  and  $B_R \leftarrow B_R - l_{ij}$ 
14:    end if
15:  end if
16: until no new link added to set  $E$ 

```

Finally, in Algorithm 8, we present a method that simultaneously solves the exact CND model (using Algorithm 1) and uses the remaining fiber budget B_R to move the resiliency value provided by the exact CND solution as close as possible to the resiliency value of the compact CND model. The inputs of Algorithm 8 are the network topology G , the remaining fiber budget B_R , the target CND optimal value \bar{z} and all the sets of critical nodes \mathcal{K} previously generated by the greedy deterministic algorithm (either Algorithm 2 or 4). Similarly to Algorithm 1, the **MILP'** model is initialized without the path constraints (line 2). Then, in order to accelerate the row generation process, the path constraints associated to each set of critical nodes $K \in \mathcal{K}$ are added to **MILP'** model (lines 3-11). Next, the optimal solution of the exact CND model is solved as in Algorithm 1. In lines 15-17, if the exact resiliency value z^* is lower than the target value \bar{z} , the algorithm uses the remaining budget to compute new links to be added to the graph so that each component of the surviving graph induced by the optimal critical node set becomes optically transparent (this is equivalent to run Algorithm 6 in this surviving graph, line 16). Algorithm 8 is repeated until no new link is added (line 18), which happens either if $z^* = \bar{z}$ (no need to add new links) or if no new link can be added due to the remaining fiber budget.

Algorithm 8 Exact CND upgrade algorithm

```

1: INPUT:  $G = (N, E)$ ,  $B_R$ ,  $\bar{z}$ ,  $\mathcal{K}$ 
2: Initialize MILP' MILP model (1)-(6) without constraints (4)
3: for all  $K \in \mathcal{K}$  do
4:   Compute subgraph  $G_K = (N \setminus K, E_K)$  where  $E_K = \{(i, j) \in E : i, j \notin K\}$ 
5:   for all node pairs  $(i, j) \notin E_K$  do
6:     Compute shortest path  $p_{ij}$  and its length  $d_{ij}$ 
7:     if  $d_{ij} \leq T$  then
8:       Add to MILP' constraint (4) corresponding to path  $p_{ij}$ 
9:     end if
10:  end for
11: end for
12: Solve the MILP' model. Let  $z^*$  be the optimal value and  $(u^*, v^*)$  the optimal solution
13: repeat
14:   [Lines 2-15 of Algorithm 1]
15:   if  $z^* < \bar{z}$  then
16:     Run Algorithm 6 in the surviving graph induced by the current optimal critical nodes
17:   end if
18: until no new link added to set  $E$ 

```

3.4 | Overall algorithm

Recall that, for a given network $G = (N, E)$ with a total fiber length L and a fiber length budget of $B = L + L'$, with $L' \geq 0$, the network design problem aims to identify a new network topology connecting all nodes in N whose total fiber length is not higher than B and the network upgrade problem aims to identify a set of fiber links within the budget L' to be added to the existing topology. So, the overall algorithm is a combination of the previous algorithms that depends on the problem type (network design or network upgrade). Algorithms 9 and 10 describe how the different algorithms are put together to solve the network design and the network upgrade problem, respectively. As previously explained, the network design algorithm (Algorithm 9) does not include the local search algorithm (Algorithm 5) and the network upgrade algorithm (Algorithm 10) does not need to run the network validation algorithms (Algorithms 6 and 7). On both Algorithms 9 and 10, we consider six algorithm variants: (i) Algorithm 2, (ii) Algorithm 4 considering S CND solutions and (iii) Algorithm 4 considering the optimal CND solutions, each case using either $f_1(i, j)$ or $f_2(i, j)$ as the criteria to select each new link.

Algorithm 9 Network design algorithm

```

1: INPUT:  $G = (N, E)$ ,  $B$ 
2: Run Algorithm 2 (or 4) for graph  $(N, \{\})$ , with  $E'$  as the set of links of RNG topology
3: if  $G$  is optically 2-connected then
4:   Run Algorithm 7 for graph  $(N, E')$ , with  $E'$  and  $B_R$  resulting from the previous method
5: else
6:   Run Algorithm 6 for graph  $(N, E')$ , with  $E'$  and  $B_R$  resulting from the previous method
7: end if
8: Run Algorithm 8 for graph  $(N, E')$ , with variables  $\bar{z}$ ,  $B_R$  and  $\mathcal{K}$  resulting from the previous methods

```

Algorithm 10 Network upgrade algorithm

-
- 1: **INPUT:** $G = (N, E)$, L'
 - 2: Run Algorithm 2 (or 4) for graph G , with $B \leftarrow L'$ and $E' \leftarrow \{\}$
 - 3: Run Algorithm 5 for graph G , with variables E' , \bar{z} and B_R resulting from the previous method
 - 4: Run Algorithm 8 for graph $(N, E \cup E')$, with variables \bar{z} , B_R and \mathcal{K} resulting from the previous methods
-

4 | COMPUTATIONAL RESULTS

All results reported in this section were obtained using the optimization software *Gurobi Optimizer* version 8.0.0, with programming language *Julia* version 0.6.2, running on a PC with an Intel Core i7-8700, 3.2 GHz and 16 GB RAM. Following [15], we have assumed a transparent reach $T = 2000$ km corresponding to the use of OTU-4 lightpaths with a line rate of 100 Gbps. Moreover, we have considered $\Delta = 60$ km. This value considers an optical node architecture with an input and an output WSS (Wavelength Selective Switch) per fiber port and assumes an attenuation of 6.0 dB inserted by each WSS. Then, assuming that the attenuation on each WSS is the main optical degradation factor suffered by a lightpath, an optical node introduces a total of 12.0 dB, equivalent to the attenuation on a fiber of 60 km, with an attenuation of 0.2 dB/km.

The network topologies used in these computational results are Germany50 [13], PalmettoNet [19] and Missouri Network Alliance (MissouriNA) [19]. Table 1 presents their topology characteristics in terms of number of nodes $|N|$ and fiber links $|E|$, total number of node pairs, minimum (δ_{\min}), average ($\bar{\delta}$) and maximum (δ_{\max}) node degree and an indication if the topology is (or is not) 2-connected.

TABLE 1 Topology characteristics of each tested network.

Network	$ N $	$ E $	no. Pairs	δ_{\min}	$\bar{\delta}$	δ_{\max}	2-Connected
Germany50	50	88	1225	2	3.52	5	Yes
PalmettoNet	45	64	990	1	2.84	5	No
MissouriNA	64	80	2016	1	2.50	5	No

In all cases, the geographical location of nodes is publicly available but the geographical routes of fiber links is not known. So, we have considered that each (existing or possible) link follows the shortest path over the surface of a sphere representing Earth. Table 2 presents the resulting length characteristics in terms of minimum (l_{\min}), average (\bar{l}), maximum (l_{\max}) and total link length (L), and diameter (i.e., the highest optical length among the shortest paths of all node pairs adding Δ for each intermediate node). Note that the three topologies are optically transparent for $T = 2000$ km since all diameter values are below 2000 km.

TABLE 2 Length characteristics (in km) of each tested network.

Network	l_{\min}	\bar{l}	l_{\max}	L	Diameter
Germany50	26	100.7	252	8859	1417
PalmettoNet	19	67.0	177	4286	1298
MissouriNA	7	50.0	307	4001	1301

In the computational experiments, we have considered $c \in \{2, 3, 4, 5, 6\}$ as the number of critical nodes. For each network and each c , we started by computing a topology with a fiber budget B equal to the total fiber length L of the original topology using Algorithm 9. Then, we computed an upgraded topology for each original topology assuming a fiber budget $L' = p \times L$ with $p = 10\%$ and 20% using Algorithm 10. Finally, we computed a topology with a fiber budget $B = L + p \times L$ also for $p = 10\%$ and 20% using Algorithm 9. These cases are the same as the ones considered in [4] so that we can compare the efficiency of the methods proposed here with the ones proposed in [4].

In all cases and in both types of problems (network design and network upgrade), we have run the six algorithm variants (see Section 3.4). In the variants with Algorithm 4 considering S CND solutions, we present the results with $S = 10$ as our preliminary tests have shown that this value is a good compromise between the running time and the algorithm efficiency. Tables 3, 4 and 5 present the resiliency values of the network upgrade solutions for the three network topologies. In these tables, in addition to the number of critical nodes c , column 'MS' refers to the solutions obtained by using the Multi-Start Greedy Randomized method proposed in [4]. Columns 'S1' and 'S2' refer to the solutions obtained by Algorithm 2 (the values 1 and 2 mean the use of function $f_1(i, j)$ and $f_2(i, j)$, respectively). Columns 'A1' and 'A2' refer to the solutions obtained by Algorithm 4 considering the optimal CND solutions and columns 'M1' and 'M2' refer to the solutions obtained by Algorithm 4 considering $S = 10$ best CND solutions. Finally, the best values of each problem instance are highlighted in bold.

TABLE 3 Resiliency values z of the network upgrade solutions for Germany50.

c	Germany50, with Upgrade = 10%							Germany50, with Upgrade = 20%						
	MS	S1	S2	A1	A2	M1	M2	MS	S1	S2	A1	A2	M1	M2
2	1081	1081	1081	1081	1081	1128	1128	1128	1128	1128	1128	1128	1128	1128
3	991	991	991	991	991	991	991	1035	1035	1035	1035	1035	1035	1035
4	830	830	830	830	867	795	867	906	946	947	946	947	906	947
5	640	616	640	640	640	666	666	756	756	756	790	790	790	790
6	498	483	478	478	478	511	487	606	583	543	606	606	583	606

TABLE 4 Resiliency values z of the network upgrade solutions for PalmettoNet.

c	PalmettoNet, with Upgrade = 10%							PalmettoNet, with Upgrade = 20%						
	MS	S1	S2	A1	A2	M1	M2	MS	S1	S2	A1	A2	M1	M2
2	821	821	821	821	821	821	821	861	861	861	861	861	821	821
3	616	616	616	616	616	616	616	709	709	709	709	709	709	744
4	427	400	389	400	389	406	412	510	510	532	524	532	532	532
5	325	333	333	337	333	325	319	380	380	380	380	380	380	380
6	235	223	238	235	238	238	234	286	295	307	325	319	310	325

The first and most important observation of these computational results is that, with the exception of one instance (PalmettoNet topology for $c = 4$ and $p = 10\%$), the resiliency value obtained by at least one of the proposed algorithm variants is not lower (in many cases, it is significantly higher) than the resiliency value of the method proposed in [4]. This means that the best obtained upgrade topologies, in general, have an higher resiliency to critical node failures

TABLE 5 Resiliency values z of the network upgrade solutions for MissouriNA.

c	MissouriNA, with Upgrade = 10%							MissouriNA, with Upgrade = 20%						
	MS	S1	S2	A1	A2	M1	M2	MS	S1	S2	A1	A2	M1	M2
2	1555	1659	1659	1659	1659	1371	1371	1659	1771	1771	1771	1771	1771	1371
3	1362	1362	1242	1362	1172	1362	1320	1446	1500	1500	1500	1500	1452	1500
4	762	870	870	874	870	906	895	1039	1194	1246	1246	1270	1270	1194
5	618	667	653	711	653	753	733	758	843	841	861	871	896	897
6	457	505	460	505	460	526	520	550	701	688	722	681	701	722

when compared to the ones provided in [4]. Additionally, the comparison of the different algorithm variants (proposed in this work) does not provide clear evidence that one of them is consistently better than the others. This means that, in practice, we might need to run all of them to compute the best upgrade solution.

Tables 6, 7 and 8 present the resiliency values of the network design solutions for the three network topologies (the meaning of each column is similar to the previous tables). In the Germany50 network, the knockout values represent invalid solutions where the algorithm variant was not able to compute a 2-connected network design solution.

TABLE 6 Resiliency values z of the network design solutions for Germany50.

c	Germany50, Network Design Problem							Germany50, Network Design Problem + 10%							Germany50, Network Design Problem + 20%							
	MS	S1	S2	A1	A2	M1	M2	MS	S1	S2	A1	A2	M1	M2	MS	S1	S2	A1	A2	M1	M2	
2	1081	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128	1128
3	991	1035	1035	961	1035	1035	1035	991	1035	1035	1035	1035	1035	1035	1035	1080	1035	1054	1081	1081	1081	1081
4	830	906	947	906	947	946	906	906	947	947	947	947	947	947	906	947	990	990	990	990	990	990
5	640	724	790	756	787	790	826	756	790	826	826	862	826	826	790	862	826	864	903	864	903	903
6	498	573	631	623	651	651	658	606	631	687	658	718	717	718	658	687	718	751	745	751	751	751

TABLE 7 Resiliency values z of the network design solutions for PalmettoNet.

c	PalmettoNet, Network Design Problem							PalmettoNet, Network Design Problem + 10%							PalmettoNet, Network Design Problem + 20%							
	MS	S1	S2	A1	A2	M1	M2	MS	S1	S2	A1	A2	M1	M2	MS	S1	S2	A1	A2	M1	M2	
2	861	861	861	861	861	861	861	861	861	902	900	903	903	903	861	903	903	903	903	903	903	903
3	676	709	744	709	744	709	709	709	780	744	744	781	781	744	744	781	781	820	781	781	781	781
4	510	510	556	556	490	556	556	582	582	610	636	640	640	640	582	605	672	669	672	640	706	706
5	379	373	380	380	380	380	379	409	429	380	429	461	461	444	480	461	549	444	524	549	549	549
6	266	295	246	315	297	325	313	322	342	337	325	342	346	337	358	346	361	367	403	391	381	381

When comparing the results using the different algorithm variants with the method in [4], it is possible to observe that, in general, the network topologies obtained in this work are much more resilient to critical node failures than the ones obtained in [4]. In the Germany50 network, all invalid topologies were obtained using function $f_1(i, j)$. So, in this case, it is preferable to use function $f_2(i, j)$ in the link selection. The reason for the superior performance of $f_2(i, j)$ is because it favors the selection of candidate links connecting nodes with lower degrees. So, there is a lower chance that the generated topology has leaves (nodes with degree one) and, even when this happens, Algorithm 7 needs, in general, a lower amount of fiber budget to make it 2-connected. Nevertheless, in the other networks (PalmettoNet and MissouriNA), there are some cases where the variants using $f_1(i, j)$ provide the best resiliency results. So, like in the network upgrade problem, in the network design problem there is no clear evidence that one of the variants is consistently better than the others.

TABLE 8 Resiliency values z of the network design solutions for MissouriNA.

c	MissouriNA, Network Design Problem							MissouriNA, Network Design Problem + 10%							MissouriNA, Network Design Problem + 20%						
	MS	S1	S2	A1	A2	M1	M2	MS	S1	S2	A1	A2	M1	M2	MS	S1	S2	A1	A2	M1	M2
2	1714	1771	1771	1771	1771	1714	1771	1714	1830	1830	1830	1830	1771	1771	1771	1830	1830	1830	1830	1771	1830
3	1495	1550	1550	1550	1550	1550	1452	1500	1599	1654	1599	1602	1654	1654	1550	1656	1656	1656	1656	1656	1656
4	1126	1231	1231	1231	1231	1311	1311	1194	1354	1399	1354	1354	1354	1399	1311	1389	1441	1491	1495	1491	1446
5	841	931	1051	1025	951	1023	1051	917	1081	1113	1091	1147	1113	1147	1081	1113	1147	1183	1221	1261	1221
6	694	784	732	764	768	742	776	717	893	861	842	893	918	926	784	972	992	956	992	1037	1066

Table 9 presents the resiliency value z of the best topologies obtained for each instance presented in Tables 3–8. Rows 'Original' refer to the original topologies (in column '0%') and upgraded topologies (in columns '10%' and '20%') while rows 'Alternative' refer to the best network design solutions with a fiber budget $B = L + p \times L$ with $p = 0\%$, 10% and 20% . For each case, columns 'UB' presents the trivial upper bound of the CND problem given by the number of pairs of $n - c$ surviving nodes, i.e., $(n - c)(n - c - 1)/2$, with $n = |N|$.

TABLE 9 Best obtained resiliency value z for all tested instances.

c	Network	Germany50				PalmettoNet				MissouriNA			
	Instance	0%	10%	20%	UB	0%	10%	20%	UB	0%	10%	20%	UB
2	Original	1036	1128	1128	1128	513	821	861	903	946	1659	1771	1891
	Alternative	1128	1128	1128		861	903	903		1771	1830	1830	
3	Original	711	991	1035	1081	346	616	744	861	602	1362	1500	1830
	Alternative	1035	1035	1081		744	781	820		1550	1654	1656	
4	Original	640	867	947	1035	284	427	532	820	455	906	1270	1770
	Alternative	947	947	990		556	640	706		1311	1399	1495	
5	Original	496	666	790	990	176	337	380	780	338	753	897	1711
	Alternative	826	862	903		380	461	549		1051	1147	1261	
6	Original	415	511	606	946	123	238	325	741	253	526	722	1653
	Alternative	658	718	751		325	346	403		784	926	1066	

An initial observation of these results is that the resiliency values are lower for higher number of critical nodes c , which is without surprise since more node failures disrupt an higher percentage of the network. Moreover, the resilience of the upgraded topologies is always significantly better for higher budget values.

The best topologies are always significantly better than the original/upgraded ones, with the exception of Germany50 for $c = 2$ and with $p = 10\%$ and 20% , where the trivial upper bound is reached in both cases. Nevertheless, when comparing the differences between the original/upgraded and the best topologies, it is significantly higher for PalmettoNet and MissouriNA topologies than for the Germany50 topology. This means that the latter topology is significantly more resilient to critical node failures than PalmettoNet and MissouriNA. To understand this fact, recall from the topology characteristics of the different networks (Table 1) that Germany50 is the topology with the highest average node degree and the only one which is 2-connected. These two characteristics make this network more resilient than the two other networks.

More important than analyzing the absolute resiliency values, we need to analyze the resiliency gap between the original/upgraded topologies and the best topologies computed with the same fiber budget values. Figure 2 plots in

a bar chart these gaps, for all networks and all values of c , computed as $\frac{z_B - z_{O/U}}{z_B} \times 100\%$ where z_B is the resiliency value of the best topology and $z_{O/U}$ is the resiliency value of the original/upgraded topology. Blue bars present the resiliency gap between the best topology and the original topology. The resiliency gaps between the best topologies and the upgraded topologies are presented in the red and green bars for $\rho = 10\%$ and 20% , respectively.

Firstly, the blue bars of Figure 2 show that the resiliency gaps are lower for Germany50 (but still significant for a number of critical nodes $c \geq 3$) and very large for PalmettoNet and MissouriNA. These results reinforce the previous conclusion that Germany50 is more resilient than the others but also show that, in general, existing network topologies are not resilient to critical node failures. Secondly, the resiliency gaps shown in the red bars (corresponding to topology designs with 10% more total fiber length) represent, in all cases, a significant gap reduction when compared with the blue bars. This means that for all tested instances, adding new links to an existing topology with a fiber budget of 10% enables solutions with resiliency to critical node failures much closer to a topology designed to maximize this resiliency with the same amount of fiber. Thirdly, the results of the green bars (corresponding to topology designs with 20% more total fiber length) are mixed, i.e., in some cases, the additional 10% fiber budget enables a significant gap reduction while in other cases, the reduction is negligible.

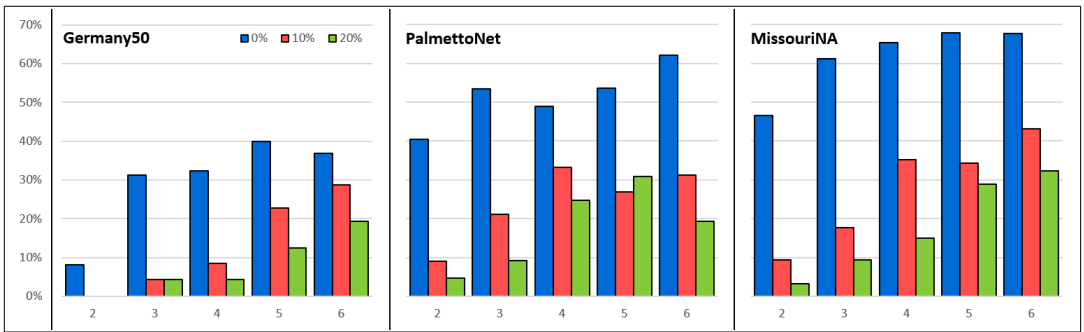


FIGURE 2 Resiliency gaps of all tested networks and for all $c \in \{2, \dots, 6\}$.

Finally, we can distinguish two cases. For a number of critical nodes $c \leq 3$, the additional fiber budget of 20% allows in all cases the resiliency gap to become small (below 10%). For a number of critical nodes $c \geq 5$ (in the Germany50 network) and $c \geq 4$ (in the less resilient PalmettoNet and MissouriNA networks), the additional fiber budget of 20% is still not enough to make the resiliency gap small. This means that more fiber links are required in the upgrade of existing networks to reach the best resiliency to higher number of critical nodes.

For illustrative purposes, Figure 3 presents the original topologies, the best upgraded topologies with $L' = 10\%L$ and $L' = 20\%L$ (with the additional links highlighted in blue) and the best topologies with the same fiber budget L obtained considering $c = 4$ critical nodes. To highlight the differences, links of the best topology (and the best upgraded topologies) not in the original topology are highlighted in blue and, in all cases, critical nodes are represented with red squares. The analysis of these topologies show that:

Germany50 The critical node set splits the original network in three components (1, 10 and 35 nodes each) while it only isolates a pair of nodes from the others in the best topology. Moreover, the critical node set isolates 4 nodes from the others in the 10% upgraded topology and a pair of nodes in the 20% upgraded topology. So, an upgrade of 20% has the same resiliency to 4 critical nodes as the best topology.

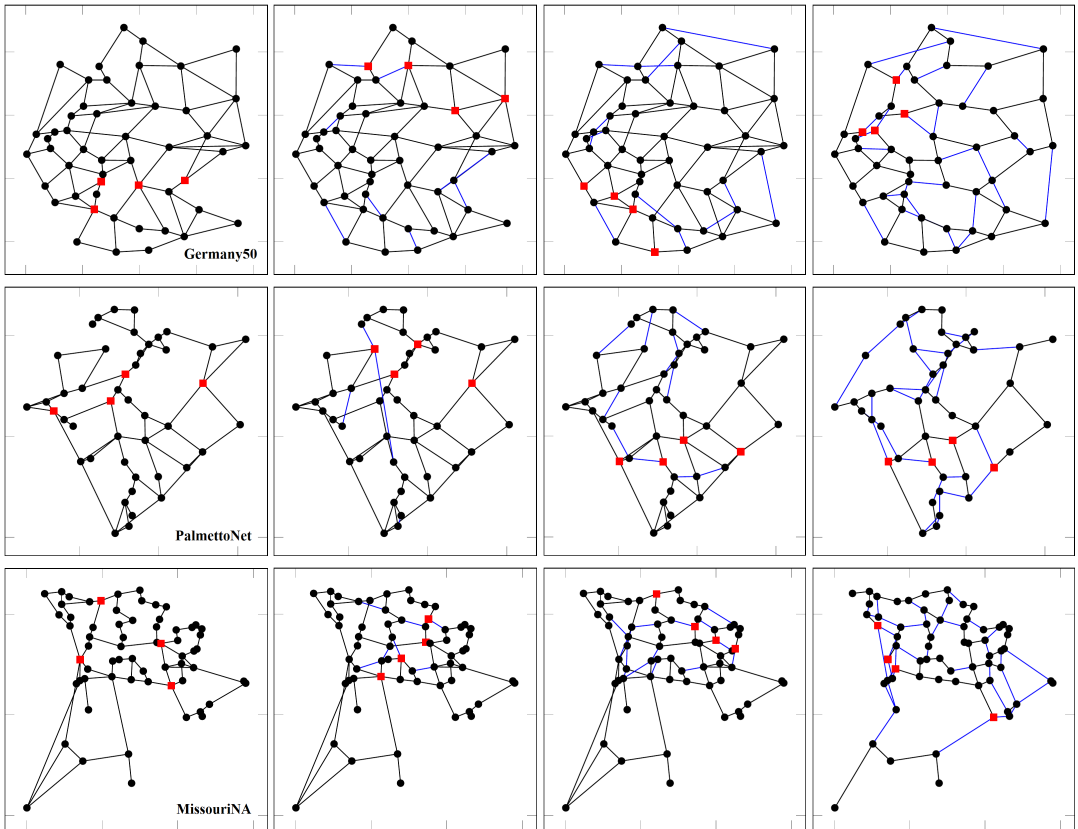


FIGURE 3 Original topologies (first column), best upgraded topologies with $L' = 10\%L$ and $20\%L$ (second and third columns, respectively) and the best topologies with $B = L$ (fourth column), considering $c = 4$ critical nodes (represented in red squares). Links not in the original topologies highlighted in blue.

PalmettoNet The critical node set splits the original network in four components (2, 6, 13 and 20 nodes each) while it splits the best topology in only two components (8 and 33 nodes each). Moreover, the critical node set splits the 10% upgraded topology in four components (2, 5, 5 and 29 nodes) and the 20% upgraded topology in two components (9 and 32 nodes). In this case, the resilience to 4 critical nodes of the 20% upgraded topology is still slightly lower than the resilience of the best topology.

MissouriNA The critical node set splits the original network in four components (8, 16, 17 and 19 nodes each) while it splits the best topology in only two components (9 and 51 nodes each). Moreover, the critical node set splits the 10% upgraded topology in two components (24 and 36 nodes) and the 20% upgraded topology in two components (10 and 50 nodes). As in the previous case, the resilience to 4 critical nodes of the 20% upgraded topology is still slightly lower than the resilience of the best topology.

Concerning the running time of the proposed algorithm variants, Table 10 presents the average running time of the Network Upgrade Problem (Algorithm 10), among all five values $c = 2, \dots, 6$ of each problem instance. In the instance name, 'Ger', 'Pal' and 'Mis' refers to the Germany50, PalmettoNet and MissouriNA networks, respectively,

while the '10' and '20' refer to the fiber budget $L' = 10\%L$ and $20\%L$, respectively. Similarly, Table 11 presents the average running time of the Network Design Problem (Algorithm 9), among all five values $c = 2, \dots, 6$ of each instance. In the instance name, the '0', '10' and '20' refer to the fiber budget $B = L + 0\%L$, $B = L + 10\%L$ and $B = L + 20\%L$, respectively.

TABLE 10 Average running time (HH:MM:SS) of the Network Upgrade Problem.

Instance	S1	S2	A1	A2	M1	M2
Ger_10	00:06:19	00:04:44	00:06:32	00:06:57	00:34:51	00:29:51
Ger_20	00:36:37	00:31:30	01:11:44	01:09:32	02:28:51	02:35:05
Pal_10	00:00:49	00:00:53	00:01:21	00:01:11	00:03:14	00:02:55
Pal_20	00:04:53	00:06:07	00:14:51	00:09:25	00:20:19	00:20:56
Mis_10	00:05:58	00:02:17	00:09:40	00:03:03	00:16:08	00:09:26
Mis_20	01:56:47	00:47:10	01:54:14	01:42:59	03:28:53	01:32:35

TABLE 11 Average running time (HH:MM:SS) of the Network Design Problem.

Instance	S1	S2	A1	A2	M1	M2
Ger_0	00:45:24	00:55:20	00:45:49	01:06:20	02:03:28	02:13:56
Ger_10	01:06:45	01:19:15	02:11:49	02:42:54	03:28:40	04:00:03
Ger_20	04:00:59	02:17:02	04:12:33	07:25:25	11:53:53	10:22:25
Pal_0	00:04:21	00:06:15	00:06:49	00:06:51	00:08:31	00:08:36
Pal_10	00:08:44	00:06:46	00:10:15	00:13:07	00:15:16	00:13:09
Pal_20	00:08:58	00:12:35	00:18:22	00:19:07	00:21:36	00:27:16
Mis_0	00:35:08	00:35:01	00:35:56	00:32:50	00:58:57	00:46:51
Mis_10	00:53:47	00:55:09	01:03:13	01:03:54	01:09:10	01:34:19
Mis_20	01:08:20	01:12:30	01:57:30	01:31:11	03:40:24	02:50:15

The analysis of the running times of Tables 10 and 11 let us draw the following conclusions. First, all algorithm variants have higher running times when the problems consider higher fiber budget values. This was expected since more links are added with an higher fiber budget and, therefore, the algorithms run a larger number of iterations.

Second, even without using the local search algorithm (Algorithm 5) in the Network Design Problem, this problem has much longer running times than the Network Upgrade Problem. The main reason is that Network Upgrade Problem starts with a fixed set of fiber links (the links of the original topology), while the Network Design Problem has to built a solution from scratch.

Third, note that at each iteration, Algorithm 2 solves a single CND model, Algorithm 4 considering the optimal CND solutions solves a variable number of CND models and Algorithm 4 considering S CND solutions solves an even higher number of CND models (in our case, 10 CND models, as we consider $S = 10$). Moreover, solving the CND models is the most time-consuming part of all algorithms. As a consequence, the running times of both Algorithms 4 are higher, on average, than the running times of Algorithm 2 and the running times of Algorithm 4 considering S CND solutions are higher, on average, than the running times of Algorithm 4 considering the optimal CND solutions. Note that the use of $f_1(i, j)$ or $f_2(i, j)$ as the criterion to select each new link does not have a significant impact in the

obtained running times.

Another aspect of interest is the comparison of the node degree distributions between the original topologies and the best topologies with the same total fiber length L . Figure 4 shows these distributions for the three network cases with the best topologies obtained for $c = 4$ critical nodes (original topologies in blue and best topologies in green). For example, in Germany50 original topology, there are 10 nodes with the minimum degree of 2 and 11 nodes with the maximum degree of 5 while in the best topology all nodes have a degree between 3 and 4. In the other two networks, we observe from the original topology to the best topology that the number of nodes with degree 1 and 2 decreases and the maximum network degree also decreases from 5 to 4 in both cases. So, the conclusion is that in the best topologies, there is a decrease in the number of nodes with the lowest and highest degrees and an increase in the number of nodes with degrees closer to the average. This observation also stands in the best topologies for the other values of c showing that resilient topologies tend to have more homogeneous node degrees.

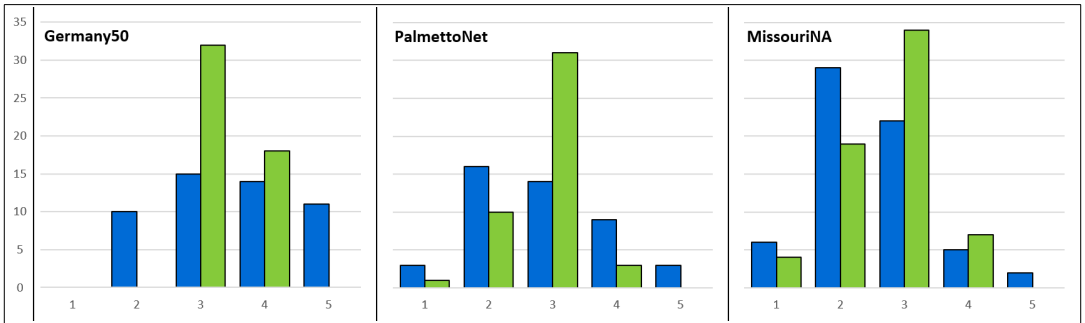


FIGURE 4 Node degree histograms of original topology (in blue) and the best topology (in green) for $c = 4$.

5 | CONCLUSIONS

In this work, we have addressed the topology design of transparent optical networks aiming to maximize their resilience to the simultaneous failure of their critical nodes. We have proposed different algorithm variants of a deterministic method that can be used both in the design of network topologies and in the upgrade of existing topologies. We have run the proposed algorithm on three network topologies with publicly available information comparing the resiliency gap between the existing and upgraded topologies with the best topologies designed to maximize its resilience with the same fiber budget.

The results have shown that the resiliency gap of existing topologies is significantly large but network upgrades with $L' = 10\%L$ can already reduce significantly the resiliency gaps provided that such upgrades are aimed at maximizing the network resiliency to the critical node failures. Finally, comparing the best topologies with the existing ones, the best topologies are characterized by a decrease of the number of nodes with the lowest and highest degrees and an increase of the number of nodes with degrees closer to the average node degree. This clearly shows that network topologies resilient to critical node failures tend to have more homogeneous degrees among all their nodes.

References

- [1] A. Agra, A. de Sousa, and M. Doostmohammadi, *The minimum cost design of transparent optical networks combining grooming, routing, and wavelength assignment*, IEEE/ACM Trans. Networking 24 (2016), 3702–3713.

- [2] A. Arulsevelan, C.W. Commander, L. Elefteriadou, and P.M. Pardalos, *Detecting critical nodes in sparse graphs*, *Comput. & Oper. Res.* **36** (2009), 2193–2200.
- [3] F. Barbosa, A. de Sousa, and A. Agra, *The design of transparent optical networks minimizing the impact of critical nodes*, *Electron. Notes Discr. Math.* **64** (2018), 165–174.
- [4] F. Barbosa, A. de Sousa, and A. Agra, *Topology design of transparent optical networks resilient to multiple node failures*, 10th International Workshop on Resilient Networks Design and Modeling (RNDM), Longyearbyen, Svalbard, Norway, 2018, pp. 1–8.
- [5] A. Beygelzimer, G. Grinstein, R. Linsker, and I. Rish, *Improving network robustness by edge modification*, *Physica A: Stat. Mechanics its Appl.* **357** (2005), 593–612.
- [6] A. de Sousa, D. Mehta, and D. Santos, *The robust node selection problem aiming to minimize the connectivity impact of any set of p node failures*, 13th International Conference on Design of Reliable Communication Networks (DRCN), 2017, pp. 138–145.
- [7] M. Di Summa, A. Grosso, and M. Locatelli, *Branch and cut algorithms for detecting critical nodes in undirected graphs*, *Comput. Optim. Appl.* **53** (2012), 649–680.
- [8] F. Dikbiyik, M. Tornatore, and B. Mukherjee, *Minimizing the risk from disaster failures in optical backbone networks*, *J. Lightwave Technology* **32** (2014), 3175–3183.
- [9] M. Furdek, L. Wosinska, R. Gosciencin, K. Manousakis, M. Aibin, K. Walkowiak, S. Ristov, M. Gushev, and J.L. Marzo, *An overview of security challenges in communication networks*, 8th International Workshop on Resilient Networks Design and Modeling (RNDM), IEEE, 2016, pp. 43–50.
- [10] T. Gomes, J. Tapolcai, C. Esposito, D. Hutchison, F. Kuipers, J. Rak, A. de Sousa, A. Iossifides, R. Travanca, J. Andre, L. Jorge, L. Martins, P.O. Ugalde, A. Pasic, D. Pezaros, S. Jouet, S. Secci, and M. Tornatore, *A survey of strategies for communication networks to protect against large-scale natural disasters*, 8th International Workshop on Resilient Networks Design and Modeling (RNDM), IEEE, 2016, pp. 11–22.
- [11] C. Natalino, A. Yayimli, L. Wosinska, and M. Furdek, *Link addition framework for optical CDNs robust to targeted link cut attacks*, 9th International Workshop on Resilient Networks Design and Modeling (RNDM), IEEE, 2017.
- [12] S. Neumayer, G. Zussman, R. Cohen, and E. Modiano, *Assessing the vulnerability of the fiber infrastructure to disasters*, *IEEE/ACM Trans. Networking* **19** (2011), 1610–1623.
- [13] S. Orłowski, R. Wessäly, M. Pióro, and A. Tomaszewski, *SNDlib 1.0 - survivable network design library*, *Networks* **55** (2010), 276–286.
- [14] J. Rak, D. Hutchison, E. Calle, T. Gomes, M. Gunkel, P. Smith, J. Tapolcai, S. Verbrugge, and L. Wosinska, *RECODIS: Resilient communication services protecting end-user applications from disaster-based failures*, 18th International Conference on Transparent Optical Networks (ICTON), IEEE, 2016.
- [15] F. Rambach, B. Konrad, L. Dembeck, U. Gebhard, M. Gunkel, M. Quagliotti, L. Serra, and V. López, *A multilayer cost model for metro/core networks*, *J. Optical Commun. Networking* **5** (2013), 210–225.
- [16] D.F. Rueda, E. Calle, and J.L. Marzo, *Robustness comparison of 15 real telecommunication networks: Structural and centrality measurements*, *J. Network Syst. Manage.* **25** (2017), 269–289.
- [17] K.A. Sabeh, M. Tornatore, and F. Dikbiyik, *Progressive network recovery in optical core networks*, 7th International Workshop on Reliable Networks Design and Modeling (RNDM), IEEE, 2015, pp. 106–111.
- [18] D. Santos, A. de Sousa, and P. Monteiro, *Compact models for critical node detection in telecommunication networks*, *Electron. Notes Discr. Math.* **64** (2018), 325–334.

- [19] Topology-zoo, *The internet topology zoo*, IEEE J. Selected Areas Commun. **29** (2011), 1765–1775.
- [20] G.T. Toussaint, *The relative neighbourhood graph of a finite planar set*, Pattern Recognition **12** (1980), 261–268.
- [21] A. Veremyev, V. Boginski, and E.L. Pasilio, *Exact identification of critical nodes in sparse networks via new compact formulations*, Optim. Lett. **8** (2014), 1245–1259.
- [22] A. Veremyev, O.A. Prokopyev, and E.L. Pasilio, *Critical nodes for distance-based connectivity and related problems in graphs*, Networks **66** (2015), 170–195.
- [23] A. Zeng and W. Liu, *Enhancing network robustness against malicious attacks*, Physical Review E **85** (2012), 066130.