Computers & Operations Research 136 (2021) 105453

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor

The minimum cost network upgrade problem with maximum robustness to multiple node failures

Fábio Barbosa [a], Agostinho Agra [a,b,*], Amaro de Sousa [a]

[a] Instituto de Telecomunicações, Universidade de Aveiro, 3810-193 Aveiro, Portugal
[b] CIDMA, Departamento de Matemática, Universidade de Aveiro, 3810-193 Aveiro, Portugal

ARTICLE INFO

Keywords:
Robust network design
Critical node detection
Mixed integer linear programming
Pareto frontier
Telecommunications

ABSTRACT

The design of networks which are robust to multiple failures is gaining increasing attention in areas such as telecommunications. In this paper, we consider the problem of upgrading an existent network in order to enhance its robustness to events involving multiple node failures. This problem is modeled as a bi-objective mixed linear integer formulation considering both the minimization of the cost of the added edges and the maximization of the robustness of the resulting upgraded network. As the robustness metric of the network, we consider the value of the Critical Node Detection (CND) problem variant which provides the minimum pairwise connectivity between all node pairs when a set of c critical nodes are removed from the network. We present a general iterative framework to obtain the complete Pareto frontier that alternates between the minimum cost edge selection problem and the CND problem. Two different approaches based on a cover model are introduced for the edge selection problem. Computational results conducted on different network topologies show that the proposed methodology based on the cover model is effective in computing Pareto solutions for graphs with up to 100 nodes, which includes four commonly used telecommunication networks.

1. Introduction

The design of networks which are robust to multiple failures is gaining increasing attention. In the area of telecommunications, which has motivated this work, multiple failures can occur due to many different reasons, as natural disasters (Gomes et al., 2016) or malicious human activities (Furdek et al., 2016), and different techniques are being investigated to enhance the preparedness of telecommunication networks for such events (Rak and Hutchison, 2020). Depending on the causes, multiple failures might involve only edges or nodes and edges (a node failure implies that its incident edges also fail). For example, in malicious human attacks, node shutdowns are harder to realize but they are the most rewarding in the attacker's perspective as the shutdown of a single node also shuts down its incident edges. Here, we address the case of multiple node failures as they are the most harmful cases of malicious human attacks.

In this work, we consider the minimum cost network upgrade problem with maximum robustness to multiple node failures (for short, robust network upgrade problem, RNUP). Given an undirected complete graph $G = (N, E_n)$ and a subset of edges $E_0 \subset E_n$, representing an existent network topology, the RNUP aims to determine a set of additional edges E' from $E_n \setminus E_0$, that maximizes the robustness of the upgraded graph $G^U = (N, E_0 \cup E')$ to multiple node failures while minimizing the total cost of the added edges. The robustness of the graph is measured by the lowest pairwise connectivity value (i.e., the lowest number of node pairs that have connectivity in the remaining graph) among all the possible scenarios of c removed nodes.

Concerning the robustness value of an upgraded graph, it can be computed with the optimal solution of an optimization problem, commonly named Critical Node Detection (CND). For a given graph and a given number c of nodes, the most common CND variant is the optimization problem that consists of computing the set of c nodes, named critical nodes, such that their deletion maximally degrades the network connectivity according to a given connectivity metric. Therefore, the optimal value of the CND variant considering the pairwise connectivity minimization represents the worst degradation that the deletion of any set of c nodes can impose in the given graph in our RNUP.

Note that, in some contexts, the determination of a set of critical nodes of a given graph is the ultimate goal of the optimization problem as, for example, the identification of the communities to be immunized in the spread of diseases (de Sousa et al., 2019). In other contexts, as the

* Corresponding author.
E-mail address: aagra@ua.pt (A. Agra).

https://doi.org/10.1016/j.cor.2021.105453
Received 5 August 2020; Received in revised form 29 May 2021; Accepted 28 June 2021
Available online 11 July 2021

one addressed in this work, the CND problem is only one part of a more general problem.

The use of the CND value as a metric to evaluate the robustness of networks to multiple node failures has been recently used in the preparedness of telecommunication networks to large-scale failures, as in de Sousa et al. (2017) where some nodes of the network are optimally selected to be made robust such that they never fail and in Barbosa et al. (2020) where the network upgrade problem (i.e., the optimal selection of new edges to be added to an existent network) is addressed. In both cases, for a given budget (in the robust nodes and in the new edges, respectively), the aim is to improve as much as possible the worst degradation imposed by the failure of the critical nodes of the resulting upgraded network. By considering the budget as a constraint, these works address a single objective optimization problem.

When dealing with multi-objective optimization problems, the main objective is to obtain optimal Pareto solutions (Caballero et al., 2002; Shukla and Deb, 2007). Instead of considering a given budget, the main goal of this work is to obtain the optimal Pareto frontier between minimizing the total cost of the edges added to the existent topology and maximizing the CND value of the upgraded topology G^U. This approach allows to generate detailed information regarding the trade-off between the upgrade cost and the gains in robustness resulting from that upgrade.

There are very few works that seek to enhance the robustness of a network to multiple node failures by increasing the topology connectivity through the addition of new links. To the best of our knowledge, in the context of network design, the recent paper (Hadian et al., 2021) is the unique to consider two objectives. One of these objectives is common to our work (minimizing the link insertion cost). Regarding the other objective, the authors aim to maximize the network technical power (measured in terms of total potential for sending and receiving flow on all nodes) which leads to a simpler bi-objective problem. Different approaches are designed to find non-dominated solutions, but the purpose is not to obtain the complete Pareto frontier. In Natalino et al. (2019), at the infrastructure level of optical networks, the robustness to link cut attacks is enhanced by increasing the topology connectivity with sparse link addition. In Barbosa et al. (2018) and Barbosa et al. (2020), the network upgrade problem aims to identify a set of links, within a given link length budget, to be added to an existing topology in order to obtain the upgraded network that maximizes the robustness in case of a simultaneous failure of a set of c critical nodes. In these works, the problem is solved by resorting to heuristic methods: a multi-start greedy randomized method in Barbosa et al. (2018) and a greedy deterministic algorithm in Barbosa et al. (2020). To the best of our knowledge, the robust network upgrade problem (as presented in this work) has never been addressed with exact methods.

Contrary to the RNUP, the CND problem has been extensively addressed in different network contexts where, depending on the context, different connectivity metrics have been considered, i.e., minimizing the pairwise connectivity, maximizing the number of connected components, minimizing the number of nodes of the largest connected component size, etc (see Lalou et al., 2018, for a recent survey). Many of these variants assume a given number c of critical nodes, which, in our problem, represents the worst-case number of nodes that can simultaneously fail. Moreover, alternative formulations for the CND problem have been proposed with different objective functions and constraints as, for example, the beta-vertex disruptor (Dinh et al., 2010) and the component-cardinality-constrained (Lalou et al., 2016) CND variants. As already mentioned, we focus on the CND variant where, for a given number c of critical nodes, the aim is to compute a set of c nodes that minimizes the pairwise connectivity of the network (Arulselvan et al., 2009; Purevsuren and Cui, 2019; Di Summa et al., 2011; Di Summa et al., 2012; Ventresca, 2012; Veremyev et al., 2014), a variant that has been used in the vulnerability evaluation of telecommunication networks to multiple node failures (de Sousa and Santos, 2020; Santos et al., 2018).

Recently, the CND problem itself has been modeled with multi-objective formulations. In Ventresca et al. (2018), the CND problem is formulated as a bi-objective problem: maximizing the number of connected components in a graph while simultaneously minimizing the variance of their cardinalities by removing a subset of critical nodes. Six known multi-objective evolutionary algorithms are tested and compared. In Li et al. (2019), a bi-objective variant of the CND problem is studied, where the two conflicting objectives are the minimization of the pairwise connectivity of the induced graph and the cost of removing the critical nodes simultaneously. Two decomposition based multi-objective evolutionary algorithms are modified and improved. In Faramondi et al. (2019), a multi-objective formulation is proposed to obtain a Pareto frontier that considers different trade-offs between conflicting objectives of the attacker. Then, using the information from the Pareto front two indices are proposed to assess the robustness of a network and to identify the critical nodes. Case studies are reported using as objectives the minimization of the network connectivity, measured in terms of pairwise connectivity, and minimization of the attack total cost. The full defender/attacker approach, where the decision maker perspective is also taken into account, is left for future research.

On one hand, contrary to our work, in Faramondi et al. (2019) and Li et al. (2019), the minimization of the cost is considered as one of the objectives but from the point of view of the attacker (i.e. the cost of removing the critical nodes simultaneously). On the other hand, in our work, the CND problem is only considered as a tool to evaluate the network robustness to multiple node failures. Thus, we consider a standard single-objective formulation to the CND problem, and the second objective of the proposed RNUP is related to the cost of the edges added to the topology to increase its robustness to multiple node failures.

We model the RNUP as a bi-objective problem and provide a path formulation. Based on the proposed path formulation for the RNUP problem, we propose a general procedure to generate all the points belonging to the Pareto frontier that solves two subproblems (an edge selection problem and the CND problem) alternately. To enhance the procedure, we propose an approach that models the selection of edges as a set covering problem. Usually, path formulations are used to ensure connectivity. However, in our case, the computational results will show that the set covering constraints are much more effective in solving the edge selection problem. Two variants are proposed, one based on a row generation approach, where cover inequalities are added on the fly, and another approach where a characterization of the relevant cover inequalities is used to select all the inequalities.

The main computational experiments are conducted on 4 well-known network topologies commonly used in telecommunications (Orlowski et al., 2010). The results show that using the general procedure can only solve very small size instances. However, using the enhanced procedures, all the tested instances are solved considering sets of $c \in \{2, 3, 4, 5, 6\}$ critical nodes and the complete Pareto frontier is obtained in instances up to $c = 4$ critical nodes. Additional computational tests are conducted on different topologies generated using three well-known graph algorithms: Erdos-Renyi model (Erdös and Rényi, 1959), Watts-Strogatz small-world model (Watts and Strogatz, 1998) and Barabasi-Albert scale-free model (Barabási et al., 1999). These tests aim at evaluating the impact of the size of the problem instances on the proposed solution procedures.

1.1. Main contributions

The original contributions of this work are summarized as follows:

- the RNUP is introduced and modeled as a bi-objective optimization problem using a path formulation;
- an upgrade algorithm is introduced to determine the complete Pareto frontier;
- a cover model is developed for the edge selection subproblem;
- two alternative algorithms based on the cover model are proposed;

- extensive computational results on different topologies and on different sizes are reported, showing the applicability of the solution approaches.

1.2. Paper outline

The paper outline is as follows. The RNUP is modeled as a bi-objective problem in Section 2. Then, a general approach to obtain the Pareto frontier is introduced in Section 3. In Section 4, we present two alternative approaches based on enhancements of the general algorithm. Computational results are reported and discussed in Section 5. In Section 6, we present the main conclusions of the conducted work.

2. Robust network upgrade problem

We consider a network represented by a connected undirected graph $G_n = (N, E_n)$ where $N = \{1, \ldots, n\}$ is the set of nodes and $E_n = \{\{i,j\} \in N \times N : i < j\}$ is the set of edges representing all possible edges. Additionally, we denote by $E_0 \subset E_n$ the subset of edges corresponding to the existing edges. For each candidate edge $\{i,j\} \in E_n \backslash E_0$, parameter l_{ij} represents the cost of installing an edge between the two nodes i and j in an upgraded solution.

The robust network upgrade problem (RNUP) consists of installing new edges to increase the robustness of a given network while minimizing the cost with the additional installed edges. We model this problem as a general bi-objective optimization problem in Section 2.1 and provide a mixed integer linear programming (MILP) path based formulation in Section 2.2.

2.1. Bi-objective optimization model

We model the RNUP as a bi-objective problem. The decisions are the new edges to add to the existing network (N, E_0). The objectives are the minimization of the cost of the added edges and the maximization of a robustness metric of the upgraded network.

Initially, we consider, for each candidate edge $\{i,j\} \in E_n \backslash E_0$, the binary decision variable y_{ij} that is 1 if edge $\{i,j\}$ is selected, and 0 otherwise. The proposed RNUP can be modeled as the following bi-objective problem:

$$min \quad L := \sum_{\{i,j\} \in E_n \backslash E_0} l_{ij} y_{ij} \tag{1}$$

$$max \quad z := f\left(E_0 \cup \left\{\{i,j\} \in E_n \backslash E_0 : y_{ij} = 1\right\}\right) \tag{2}$$

$$s.t. \quad y_{ij} \in \{0,1\}, \ \{i,j\} \in E_n \backslash E_0. \tag{3}$$

where $f(E)$ is a robustness metric of the network $G^U = (N, E)$. The first objective (1) is to minimize the total cost of the new edges.

In this work, the robustness metric is given by the objective function of the CND problem, that will be denoted by $\mathbf{CND}(E)$, i.e., $f(E) = \mathbf{CND}(E)$. The CND problem identifies a set of c nodes whose removal from G^U minimizes the pairwise connectivity on the remaining graph. So, the second objective (2) of the RNUP is to maximize the connectivity of the remaining graph assuming that the c critical nodes are removed.

2.2. Bi-objective mixed integer linear formulation

Next, we provide a MILP formulation for the bi-objective problem. Following the classical models to ensure connectivity between pair of nodes (see for instance Shen et al., 2012), we propose a path based formulation where a path links each pair of nodes that have connectivity.

Consider the set $\mathscr{K}$ of all combinations of c nodes from N (i.e., set of all failure scenarios of c nodes). For each $K \in \mathscr{K}$, the binary parameter α_i^K is 1 if and only if node $i \in N$ belongs to the node set K. For each edge $\{i,j\} \in E_n$, we consider two arcs (i,j) and (j,i) obtained from the two possible orientations of the edge. The set of all arcs will be denoted by A.

In addition to variables y_{ij} introduced before, we consider the following two sets of binary decision variables. For each node pair $s,t \in N, s < t$, for each failure scenario $K \in \mathscr{K}$ and for each arc $(i,j) \in A$, variable x_{ij}^{stK} is 1 if arc (i,j) belongs to the path between s and t in the failure scenario K, and 0 otherwise. Additionally, for each node pair $s, t \in N, s < t$ and for each failure scenario $K \in \mathscr{K}$, variable u_{st}^K is 1 if nodes s and t remain connected in the failure scenario K, and 0 otherwise.

For ease of notation, we define variables y_{ij} also for the edge set E_0 which are set to 1. Then, the bi-level problem (1)–(3) can be defined by the following MILP formulation:

$$min \quad L := \sum_{\{i,j\} \in E_n \backslash E_0} l_{ij} y_{ij} \tag{4}$$

$$max \quad z \tag{5}$$

$$s.t. \quad z \leqslant \sum_{s \in N} \sum_{t \in N, s<t} u_{st}^K, \quad K \in \mathscr{K}, \tag{6}$$

$$\sum_{j \in N:(i,j) \in A} x_{ij}^{stK} + \alpha_i^K \leqslant 1, \quad s,t \in N, s < t, i \in N, K \in \mathscr{K}, \tag{7}$$

$$\sum_{j \in N:(j,i) \in A} x_{ji}^{stK} + \alpha_i^K \leqslant 1, \quad s,t \in N, s < t, i \in N, K \in \mathscr{K}, \tag{8}$$

$$\sum_{j \in N:(s,j) \in A} x_{sj}^{stK} = u_{st}^K, \quad s,t \in N, s < t, K \in \mathscr{K}, \tag{9}$$

$$\sum_{j \in N:(i,j) \in A} x_{ij}^{stK} = \sum_{j \in N:(j,i) \in A} x_{ji}^{stK}, \quad s,t \in N, s < t, i \in N \backslash \left\{s,t\right\}, K \in \mathscr{K}, \tag{10}$$

$$\sum_{j \in N:(j,t) \in A} x_{jt}^{stK} = u_{st}^K, \quad s,t \in N, s < t, K \in \mathscr{K}, \tag{11}$$

$$\sum_{j \in N:(j,s) \in A} x_{js}^{stK} = 0, \quad s,t \in N, s < t, K \in \mathscr{K}, \tag{12}$$

$$x_{ij}^{stK} \leqslant y_{\{ij\}}, \quad s,t \in N, s < t, \left(i,j\right) \in A, K \in \mathscr{K}, \tag{13}$$

$$y_{ij} = 1, \quad \left\{i,j\right\} \in E_0, \tag{14}$$

$$y_{ij} \in \left\{0,1\right\}, \quad \left\{i,j\right\} \in E_n \backslash E_0, \tag{15}$$

$$x_{ij}^{stK} \in \left\{0,1\right\}, \quad s,t \in N, s < t, \left(i,j\right) \in A, K \in \mathscr{K}, \tag{16}$$

$$u_{st}^K \in \left\{0,1\right\}, \quad s,t \in N, s < t, K \in \mathscr{K}, \tag{17}$$

$$z \geqslant 0. \tag{18}$$

Again, the objective (4) is to minimize the total cost L of the new edges. Then, the objective (5) is to maximize the robustness z of the upgraded topology. Constraints (6) guarantee that z is at most the pairwise connectivity of the remaining graph after each set K of critical nodes is removed. Thus, they ensure that variable z cannot exceed the robustness value of any failure scenario $K \in \mathscr{K}$. Combined with objective (5), which maximizes z, in an optimal solution, this upper bound is attained.

Constraints (7)–(8) ensure that arc (i,j) does not belong to any path if either nodes i or j are critical in each failure scenario $K \in \mathscr{K}$. Constraints (9)–(11) represent the flow conservation constraints if nodes s and t remain connected in the failure scenario $K \in \mathscr{K}$. Additionally, constraints (12) ensure that there will be no flow entering the source node s in the flow conservation constraints, i.e., removing the possibility of cycles.

Notation $y_{\{ij\}}$ represents decision variable y_{ij} if $i < j$, and variable y_{ji} otherwise, and then, constraints (13) guarantee that if either arc (i,j) or (j,i), with $i < j$, is used for any path in any failure scenario, then the edge $\{i,j\}$ must exist in the upgraded topology (i.e., $y_{ij} = 1$). Finally, constraints (14)–(18) are the variable domain constraints. Notice that this formulation assumes that edges from the original topology belong to the upgraded topology (i.e., $y_{ij} = 1$ for all $\{i,j\} \in E_0$).

We can observe that this formulation depends on the cardinality of $\mathscr{K}$ which increases exponentially as a function of the number of nodes if c is large. However, for small values of c, the size of $\mathscr{K}$ is small. Notice additionally that for each $K \in \mathscr{K}$, the resulting formulation is compact.

3. Solution approach: finding the Pareto frontier

When dealing with bi-objective problems, the most relevant information from the decision maker's point of view is to know the Pareto frontier allowing to compare the cost of a network upgrade with the gains in the given robustness metric. The proposed algorithms are designed to derive all the non-dominated solution pairs (L,z) in the Pareto frontier, where L is the cost of the added edges and z is the robustness value.

First, in Section 3.1, we present a general algorithm to obtain the Pareto frontier which uses two optimization problems, one for computing the robustness value of a topology (solving a CND problem) and another for selecting a set of edges for the upgraded topology, denoted as the Edge Selection Problem (ESP). An Integer Linear Programming (ILP) formulation for the CND problem is given in Section 3.2 and a path formulation for the ESP is given in Section 3.3.

3.1. A general algorithm for the RNUP

Here, we present a general algorithm that generates a set of pairs (L_s, z_s) that includes all the Pareto optimal solutions of the RNUP. The iterative algorithm considers the graph $G = (N, E_0)$ and starts with the trivial pair $(L_1, z_1) = (0, \mathbf{CND}(E_0))$ belonging to the Pareto frontier. In each iteration, a new pair (L_s, z_s) is obtained that strictly increases the value of z. The iterative step stops when the CND value of the upgraded topology z_s is maximal, i.e., when $z_s = \binom{n-c}{2}$. The full description of the algorithm is given in Algorithm 1.

Algorithm 1: General upgrade algorithm

1: **Input**: $G = (N, E_0)$, $c \in \{1, \ldots, |N|\}$
2: $s \leftarrow 1$
3: $(L_s, z_s) \leftarrow (0, \mathbf{CND}(E_0))$
4: **while** $z_s < \binom{n-c}{2}$ **do**
5: $s \leftarrow s + 1$
6: Compute set of edges $E' \subseteq E_n \backslash E_0$ such that $L_s \leftarrow \sum_{\{i,j\} \in E'} l_{ij}$ is minimized and $\mathbf{CND}(E_0 \cup E') > z_{s-1}$
7: $z_s \leftarrow \mathbf{CND}(E_0 \cup E')$
8: **end while**

This algorithm computes the CND value in Steps 3 and 7 and solves the ESP in Step 6. These two problems are solved using ILP formulations which are described in the following sections.

3.2. Critical node detection MILP formulation

Several MILP formulations have been proposed for the CND problem (see Arulselvan et al., 2009 and the more recent papers discussing formulations enhancements Pavlikov, 2018; Santos et al., 2018). Here, we consider the MILP formulation introduced in Santos et al. (2018).

Aiming to minimize the pairwise connectivity of the graph without c critical nodes,consider the following two sets of decision variables: for each node $t \in N$, variable v_t is 1 if t is a critical node, and 0 otherwise; and for each node pair $s, t \in N, s < t$, variable u_{st} is 1 if nodes s and t remain connected (i.e., if exists a feasible path connecting end-nodes s and t) in graph $G^U = (N, E)$ after the removal of the critical nodes, and 0 otherwise. Additionally, for all $s, t \in N$ (with $s \neq t$), the notation $u_{\{st\}}$ represents the decision variable u_{st} if $s < t$, and variable u_{ts} otherwise.

Moreover, for each node pair $s, t \in N, s < t$, consider that the set $N_E^{st} \subseteq N$ represents the set of adjacent nodes to s, on graph $G^U = (N, E)$, if the node degree of s is not higher than the node degree of t, and the set of adjacent nodes to t otherwise.

Then, for a given number $c \in \{1, \ldots, |N|\}$ of critical nodes, a compact formulation for the CND problem is given by the following MILP formulation:

$$min \quad z := \sum_{s,t \in N, s<t} u_{st} \tag{19}$$

$$s.t. \quad \sum_{t \in N} v_t = c, \tag{20}$$

$$u_{st} + v_s + v_t \geq 1, \quad s, t \in N, s < t, \{s,t\} \in E, \tag{21}$$

$$u_{st} \geq u_{\{sk\}} + u_{\{tk\}} - 1 + v_k, \quad s, t \in N, s < t, \{s,t\} \notin E, k \in N_E^{st}, \tag{22}$$

$$v_t \in \{0,1\}, \quad t \in N, \tag{23}$$

$$u_{st} \in \{0,1\}, \quad s, t \in N, s < t. \tag{24}$$

The objective (19) is to minimize the pairwise connectivity, i.e., the total number of node pairs that have connectivity in the remaining graph $(N \backslash K, E^K)$, with $E^K = \{\{i,j\} \in E : i,j \notin K\}$ and where $K = \{i \in N : v_i^* = 1\}$ is the set of critical nodes. Constraint (20) ensures that exactly c nodes of N are selected as critical nodes.

Constraints (21) guarantee that a pair of adjacent nodes in graph $G^U = (N, E)$ is connected if none of the end-nodes is a critical node. Constraints (22) represent a generalization of constraints (21) for each pair of nodes s and t that are not adjacent, and guarantee that those nodes are connected if there is a non-critical node $k \in N_E^{st}$ connected to both s and t.

Constraints (23)-(24) are the variable domain constraints. As noted in Santos et al. (2018), constraints (24) can be replaced by $u_{st} \geq 0$, reducing the number of binary variables. Henceforward, the optimal value z^* of this MILP problem will be represented by $\mathbf{CND}(E)$.

3.3. A path formulation for the ESP

Next, we consider the optimization problem defined in Step 6 of Algorithm 1. This optimization problem seeks a set of edges with the minimum cost that provides a CND value greater than a given threshold r (where, in Step 6, $r = z_{s-1}$).

We denote this problem by $\mathbf{ESP}(r)$ which can be modeled as an ILP path based formulation as follows:

$$min \quad \sum_{\{i,j\} \in E_n \backslash E_0} l_{ij} y_{ij} \tag{25}$$

$$s.t. \quad \sum_{\{s,t\} \in E_n} u_{st}^K \geq r + 1, K \in \mathscr{K}, \quad (y, x, u) \text{ satisfies} (7) - (17). \tag{26}$$

Notice that, as the CND value must be integer and it must be greater than the threshold r, the right-hand side of (26) is $r + 1$. The set of edges $E' = \{\{i,j\} \in E_n \backslash E_0 : y_{ij}^* = 1\}$ is the optimal solution of $\mathbf{ESP}(r)$ and $G^U = (N, E_0 \cup E')$ denotes the upgraded topology.

Theorem 1. *The pairs $(L_1, z_1), \ldots, (L_S, z_S)$ generated by Algorithm 1 include all the Pareto optimal solutions to the bi-objective optimization problem (1)–(3).*

Proof. Let (L, z) be a Pareto optimal solution not generated by

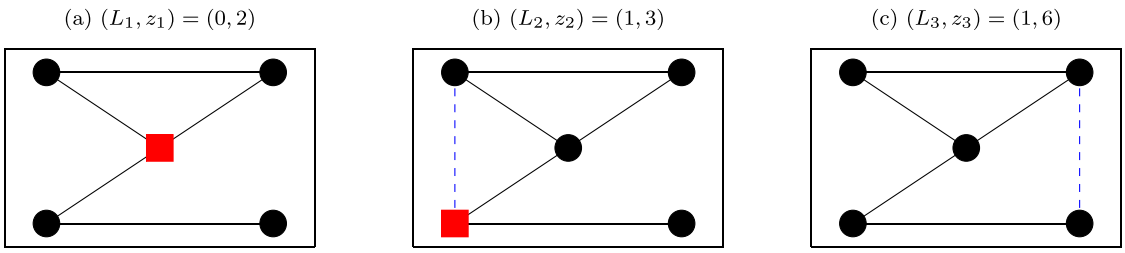

Fig. 1. Example where solution (b) is dominated by solution (c) for $c = 1$ critical node and with $l_{ij} = 1$ in both selected new edges (critical node in a red square and selected edge in dashed blue).

Algorithm 1.

Suppose that $L_s < L < L_{s+1}$, for some $s \in \{1,\ldots,S\}$. From Step 6, L_{s+1} is the optimal value of the subproblem **ESP**(z_s). Then, for each $E \subseteq E_n$, with $E_0 \subseteq E$, such that $\sum_{\{i,j\} \in E \setminus E_0} l_{ij} \leqslant L < L_{s+1}$, we have $z \leqslant z_s$. Thus, (L, z) is dominated by (L_s, z_s), which contradicts the assumption that (L, z) is a Pareto optimal solution.

Suppose $L = L_s$ for some $s \in \{1, \ldots, S\}$. If $z < z_s$, then (L, z) is dominated by (z_s, L_s) which contradict the assumption that (L, z) is a Pareto optimal solution. Consider now the case $z > z_s$. We may assume $L_{s+1} > L$, otherwise we could iteratively replace s by $s+1$ until the desirable condition $L_{s+1} > L$ is verified. Thus, at Step 6, L_{s+1} is the optimal value of the subproblem **ESP**(z_s) which contradicts $z \geqslant z_s + 1$ and $L < L_{s+1}$. Hence, the unique possible case is $z = z_s$. Then, (L, z) must coincide with (L_s, z_s) for some $s \in \{1, \ldots, S\}$, which shows that all the Pareto optimal solutions are generated by Algorithm 1. □

Although all the Pareto optimal solutions are generated with Algorithm 1, some of the solutions obtained with the algorithm may be dominated. Fig. 1 presents such an example where solutions (b) and (c) are alternative optimal solutions to **ESP**(2) assuming that the cost parameters l_{ij} correspond to the Euclidean distance between nodes i and j. However, solution (b) is dominated by solution (c). Hence, if Algorithm 1 obtains (b) before (c), a dominated solution is generated.

The next result allows to identify all the dominated solutions generated by Algorithm 1.

Proposition 1. *A pair (L_s, z_s) generated by Algorithm 1 is dominated if and only if $z_s < z_{s+1}$ and $L_s = L_{s+1}$.*

Proof. Suppose that (L_s, z_s) is not a Pareto optimal solution. Since the robustness value is strictly increasing from each iteration to the next, the condition $z_s < z_{s+1}$ is straightforward.

Given that L_s and L_{s+1} are both the minimal objective function values of (25) for the thresholds z_s and z_{s+1}, respectively, we have that $L_s \leqslant L_{s+1}$. Suppose that $L_s < L_{s+1}$. Then, $L_{s-1} \leqslant L_s < L_{s+1}$ and $z_{s-1} < z_s < z_{s+1}$, which contradicts the assumption of (L_s, z_s) not being a Pareto optimal solution. Thus, $L_s = L_{s+1}$.

The converse implication is straightforward by definition of a Pareto optimal solution. □

Remark 1. Although it is theoretically possible to obtain a solution that is not a Pareto optimal solution, in all instances where the cost l_{ij} of new links is related with the distance between the nodes $i,j \in N$, none of the tested algorithms computed such a dominated solution. This shows how rare those dominated solutions are in real-world topologies when compared to academic scenarios like the one presented in Fig. 1. Moreover, as checking if a solution is dominated can easily be done in linear time, we omit the step of verifying if a solution is dominated in all the algorithms presented in this paper.

4. Covering approach to the robust network upgrade problem

The path formulation for the **ESP**(r) presented in the previous section has the advantage of being a compact model for each set of critical nodes K. Nevertheless, it includes many variables, which implies to solve large size models in each iteration of Algorithm 1. As a consequence, the computational results will show (in Section 5.1) that it is only able to compute optimal Pareto solutions for a very limited range of instances. In order to use Algorithm 1 to obtain Pareto frontiers for larger instances, a different approach must be considered to optimally solve the RNUP.

In this section, we introduce an alternative ILP model to **ESP**(r). This model results from transforming the ESP into a set covering problem (see for instance Chvatal et al., 1979) and will be called henceforward the *Cover* model. As will be described next, the Cover model has the advantage of having a small number of variables, but it has the disadvantage of including much more constraints (the cover inequalities). In order to use the Cover model efficiently, we need to address the issue of managing these constraints efficiently.

Next, we first define the Cover model (Section 4.1). Then, we introduce the general algorithm based on the Cover model (Section 4.2). Finally, we present two algorithms based on two different strategies to generate the cover inequalities: a row generation algorithm (Section 4.3) and an algorithm based on the partition of the network into components (Section 4.4). The advantages of using this alternative model will be discussed later.

4.1. Cover model for network upgrade

In order to define the Cover model, first we must introduce some notation. Given a set of edges E and a set of critical nodes $K \subset N$, consider the remaining graph $G_K^E = (N \setminus K, E^K)$, with $E^K = \{\{i,j\} \in E : i,j \notin K\}$. Let z_K^E be the robustness value of this graph, i.e., the total number of node pairs that have connectivity in graph G_K^E.

Moreover, consider the set of edges of an auxiliary graph $(N \setminus K, \overline{E}^K)$ where two nodes are adjacent if and only if they belong to the same connected component in graph G_K^E, i.e., $\overline{E}^K = \{\{i,j\} \in E_n : i,j \in N \setminus K \text{ have connectivity in } G_K^E\}$.

We are interested in those edges that link different connected components in G_K^E, in order to increase the robustness of the upgraded network. Given the introduced notation, we now consider the set:

$$\gamma_K^E := \left\{ \{i,j\} \in E_n : i,j \notin K \text{ and } \{i,j\} \notin \overline{E}^K \right\} \tag{27}$$

which corresponds to the set of candidate edges $\{i,j\} \in E_n$ such that nodes i and j have no connectivity in the remaining graph G_K^E.

Additionally, for a given threshold r for the robustness value, we define the family of sets of candidate edges $\Gamma(r)$ as follows:

$$\Gamma(r) := \left\{ \gamma_K^E : E \subset E_n, \text{ with } E_0 \subseteq E, \text{ and } K \in \mathscr{K} \text{ such that } z_K^E \leqslant r \right\}. \tag{28}$$

The family $\Gamma(r)$ considers all the topologies G_K^E resulting from a simultaneous failure of the nodes in K, whose robustness value does not exceed the threshold r. Thus, in order to increase the robustness of this topology, at least one additional edge from each of these γ_K^E sets must be added.

Using this observation, we define a set covering problem, denoted by **Cover**$(\Gamma(r))$, which includes one cover constraint for each set of candi-

date edges $\gamma_K^E \in \Gamma(r)$ such that $z_K^E \leqslant r$:

$$min \quad \sum_{\{i,j\} \in E_n \setminus E_0} l_{ij} y_{ij} \tag{29}$$

$$s.t. \quad \sum_{\{i,j\} \in \gamma_K^E} y_{ij} \geqslant 1, \quad \gamma_K^E \in \Gamma\left(r\right), \tag{30}$$

$$y_{ij} \in \{0,1\}, \qquad \{i,j\} \in E_n \setminus E_0. \tag{31}$$

The objective (29) is to minimize the total cost of the selected candidate edges to add to the current topology (N, E_0). For a given threshold r, constraints (30) are the cover inequalities for each set of edges $E \subset E_n$, with $E_0 \subseteq E$ (i.e., for each upgraded graph), and for each critical node set $K \in \mathscr{K}$ such that $z_K^E \leqslant r$. These cover inequalities cut off all infeasible solutions based on the remaining edge set E^K, i.e., a solution with $y_{ij} = 1$, if $(i,j) \in \gamma_K^E$, and $y_{ij} = 0$ otherwise (see Proposition 2 below). Finally, constraints (31) are the variable domain constraints.

Proposition 2. *Given $E \subset E_n$ and $K \in \mathscr{K}$, the incidence vector $\bar{y}$ of the remaining graph G_K^E, i.e., $\bar{y}_{ij} = 1$, for $\{i,j\} \in E^K$ and $\bar{y}_{ij} = 0$, for $\{i,j\} \in E_n \setminus E^K$, is violated by the cover inequality (30) defined for γ_E^K.*

Proof. If $\{i,j\} \in E^K$, then $\{i,j\} \notin \gamma_K^E$, since $E^K \subseteq \overline{E}^K$. Thus, $\bar{y}_{ij} = 0$ for all $\{i,j\} \in \gamma_K^E$, which implies that constraint (30) is violated by $\bar{y}$. □

The next result illustrates how the Cover ILP model (29)–(31) can be used to optimally solve Step 6 of Algorithm 1.

Theorem 2. *Let y^* be the optimal solution of* Cover$(\Gamma(z_{s-1}))$ *for a robustness value z_{s-1}. Then $z_s > z_{s-1}$, where $z_s = \mathbf{CND}(E_0 \cup \{\{i,j\} \in E_n \setminus E_0 : y_{ij}^* = 1\})$.*

Proof. Suppose that $z_s \leqslant z_{s-1}$. Given $K = \{i \in N : v_i^* = 1\}$, then as $z_s \leqslant z_{s-1}$, it follows that $\gamma_K^{E_0 \cup E'} \in \Gamma(z_{s-1})$, where $E' = \{\{i,j\} \in E_n \setminus E_0 : y_{ij}^* = 1\}$. Setting $E = E_0 \cup E'$, and considering $\bar{y}_{ij} = 1$ for $\{i,j\} \in E_0$ and $\bar{y}_{ij} = y_{ij}^*$ for $\{i,j\} \in E_n \setminus E_0$, then by Proposition 2, the cover inequality $\sum_{\{i,j\} \in \gamma_K^E} y_{ij} \geqslant 1$ is violated. This contradicts the assumption that y^* is a feasible solution of Cover$(\Gamma(z_{s-1}))$. □

4.2. Cover-based upgrade algorithm

Here, we propose Algorithm 2 that uses the proposed Cover model to obtain the Pareto optimal solutions, i.e., in each iteration of the algorithm, a candidate solution (L_s, z_s) is computed.

Algorithm 2: Cover-based upgrade algorithm

1: **Input:** $G = (N, E_0)$, $c \in \{1, \ldots, |N|\}$
2: $s \leftarrow 1$
3: $(L_s, z_s) \leftarrow (0, \mathbf{CND}(E_0))$
4: **while** $z_s < \binom{n-c}{2}$ **do**
5: $s \leftarrow s + 1$
6: $y^* \leftarrow$ optimal solution to **Cover**$(\Gamma(z_{s-1}))$
7: $L_s \leftarrow \sum_{\{i,j\} \in E_n \setminus E_0} l_{ij} y_{ij}^*$
8: $z_s \leftarrow \mathbf{CND}(E_0 \cup \{\{i,j\} \in E_n \setminus E_0 : y_{ij}^* = 1\})$
9: **end while**

As input to the algorithm, we are given the network topology $G = (N, E_0)$ to be upgraded and the number of critical nodes $c \in \{1, \ldots, |N|\}$. The algorithm starts by solving the CND problem for the original topology (Line 3) and by assigning the first (trivial) Pareto solution.

The main loop (Lines 4–9) stops when z_s reaches the upper bound of the problem, i.e., when the CND value of the upgraded topology is maximal. First, we increase the current index solution s. Then, the Cover model is optimized for the family of cover constraints $\Gamma(z_{s-1})$, i.e., with the threshold of the previous candidate solution z_{s-1}. Finally, the CND of the upgraded topology $(N, E_0 \cup E')$ is computed, where E' is the minimal cost of the corresponding upgraded topology.

Theorem 3. *The pairs $(L_1, z_1), \ldots, (L_S, z_S)$ generated by Algorithm 2 include all the Pareto optimal solutions of the bi-objective optimization problem (1)–(3).*

Proof. In order to prove that Algorithm 2 includes all the Pareto optimal solutions, using Theorem 1, it suffices to show that this algorithm generates the same points of Algorithm 1.

Notice that Algorithm 2 is obtained by replacing Line 6 of Algorithm 1 with Lines 6–7. In each iteration of Algorithm 2, the Cover problem is solved allowing to obtain the minimum cost set of candidate edges with value L_s such that, by Theorem 2, $z_s = \mathbf{CND}(E_0 \cup \{\{i,j\} \in E_n \setminus E_0 : y_{ij}^* = 1\}) > z_{s-1}$. Therefore, both algorithms generate the same solutions. □

In contrast to the path formulation for the ESP, which is compact for a set K, the number of cover inequalities (30) increases exponentially with the size of the graph and leads to large size models that hardly can be solved to optimality even for relatively small instances. The main challenge is to devise approaches that use a small number of cover constraints to obtain the Pareto frontier. We address this challenge in two distinct ways: by using a row generation technique and by splitting the set of different topologies into equivalence classes and generate a cover constraint for each class.

4.3. Row generation approach

Here, we propose a row generation algorithm, where the family of inequalities (30) is initially ignored. In each iteration, the relaxed model is solved and an upgraded solution with edge set E is obtained. For a given threshold r, if $z_K^E \leqslant r$ for some $K \in \mathscr{K}$, then a new cut for the edge set γ_K^E is added. This procedure is described in Algorithm 3.

Algorithm 3: Row generation

1: **Input:** $G = (N, E_0)$, $c \in \{1, \ldots, |N|\}$
2: $s \leftarrow 1$
3: $(L_s, z_s) \leftarrow (0, \mathbf{CND}(E_0))$
4: $v^* \leftarrow$ optimal solution to **CND**(E_0)
5: $\Gamma \leftarrow \{\gamma_K^{E_0}\}$, where $K = \{i \in N : v_i^* = 1\}$
6: **while** $z_s < \binom{n-c}{2}$ **do**
7: $s \leftarrow s + 1$
8: **repeat**
9: $y^* \leftarrow$ optimal solution to **Cover**(Γ)
10: $L_s \leftarrow \sum_{\{i,j\} \in E_n \setminus E_0} l_{ij} y_{ij}^*$
11: $E' \leftarrow \{\{i,j\} \in E_n \setminus E_0 : y_{ij}^* = 1\}$
12: $z^* \leftarrow \mathbf{CND}(E_0 \cup E')$
13: $v^* \leftarrow$ optimal solution to **CND**$(E_0 \cup E')$
14: $\Gamma \leftarrow \Gamma \cup \{\gamma_K^{E_0 \cup E'}\}$, where $K = \{i \in N : v_i^* = 1\}$
15: **until** $z^* > z_{s-1}$
16: $(L_s, z_s) \leftarrow (L^*, z^*)$
17: **end while**

Similarly to Algorithm 2, the input is the original network topology $G = (N, E_0)$ and the number of critical nodes c. The algorithm starts by solving the CND problem for this topology and by assigning the initial Pareto optimal solution (Lines 3–4). Additionally, the family Γ of edge sets corresponding to the active cover constraints (30) is initialized (Line 5) with the set of candidate edges $\gamma_K^{E_0}$, where K is the set of critical nodes of the input graph.

The main loop (Lines 6–17) ensures that the algorithm stops when z_s reaches the upper bound. In each loop iteration, a new candidate solution (L_s, z_s) is generated.

The row generation phase is considered in the loop defined by Lines 8–15. The Cover model is optimized for the family of cover constraints Γ (Lines 9–10) and the CND of the current upgraded topology $(N, E_0 \cup E')$

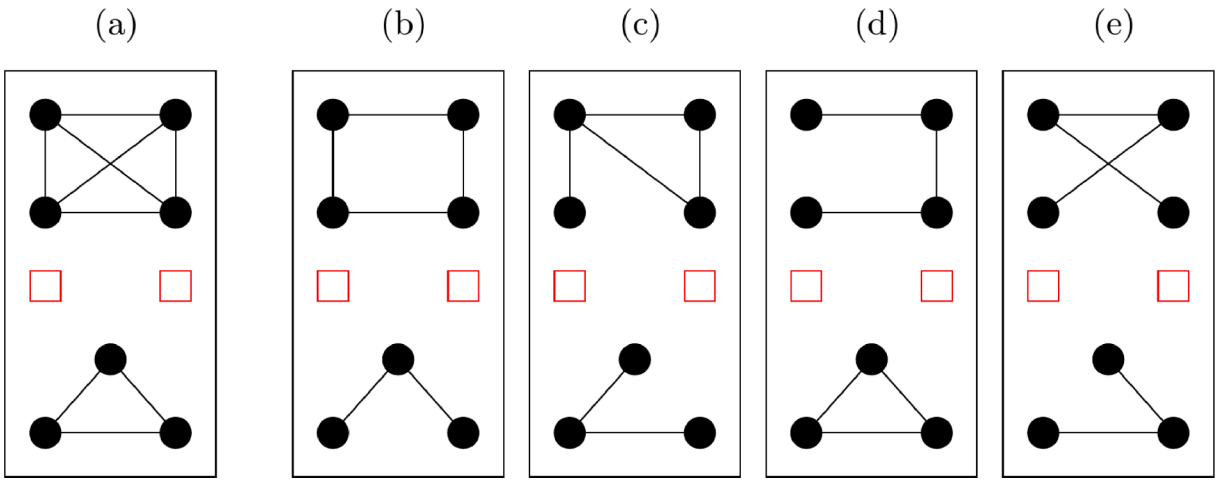

Fig. 2. All the edge sets of the remaining graphs represented in the figure belong to the same class. This class is represented by the set of edges from the graph represented in (a).

is computed (Lines 12–13). Then, based on the optimal solutions of these two optimization problems, the cover cut set $\gamma_K^{E_0 \cup E'}$ is added to family Γ (Line 14). This process is repeated until the robustness value z^* of upgraded topology is higher than the robustness value of the previous solution z_{s-1}. When this happens, the next candidate solution (L_s, z_s) is assigned to the current solution (Line 16).

4.4. Cover inequalities from partitions of the set of nodes

Given a set of critical nodes $K \in \mathscr{K}$, the family of edge sets forms an equivalence class where two sets $E_1^K, E_2^K \subset E_n$ belong to the same class if and only if $\overline{E}_1^K = \overline{E}_2^K$, i.e., graphs $G_{E_1}^K = (N \backslash K, E_1^K)$ and $G_{E_2}^K = (N \backslash K, E_2^K)$ have the same connected components. Each set E^K is represented by the set $\overline{E}^K$. Fig. 2 illustrates this concept.

Hence, if E_1^K, E_2^K belong to the same class, i.e., $\overline{E}_1^K = \overline{E}_2^K$, then $\gamma_K^{E_1} = \gamma_K^{E_2}$. Consequently, the cover inequality (30) is the same for all the topologies belonging to the same class and, in particular, for the topology where each component forms a clique. Thus, the inequality (30) can alternatively be defined from the node set partition corresponding to the topology.

Therefore, in order to compute the family of cover inequalities (30) for a given threshold r (and for each set of critical nodes $K \in \mathscr{K}$), we need to consider all the partitions of the node set $N \backslash K$ that take into account the existing edge set E_0 and whose robustness value is not higher than the threshold r. The next Example 1 illustrates how the robustness value of each partition is calculated.

Notice that, in general, the robustness value of any topology (and, by consequence, any components partition) has the format $z = \sum_{i=1}^{m} \binom{n_i}{2}$ with $\sum_{i=1}^{m} n_i = n - c$, where m represents the number of components in the remaining graph and each n_i represent the number of nodes of component i, with $i \in \{1, \ldots, m\}$. This is a particular property of the CND variant used in this work which considers the minimization of the pairwise connectivity.

Example 1. Given a node set $K \in \mathscr{K}$, suppose that the remaining graph $G_{E_0}^K$ has three connected components, i.e., $N \backslash K = C_1 \cup C_2 \cup C_3$, with $C_1, C_2, C_3 \subset N$ such that $C_1 \cap C_2 = C_1 \cap C_3 = C_2 \cap C_3 = \varnothing$. Additionally, $n_i = |C_i|$, for each $i \in \{1, 2, 3\}$. Thus, $n_1 + n_2 + n_3 = n - c$. The robustness values for the partitions with three and two components are given as follows:

- $z(C_1, C_2, C_3) = \binom{n_1}{2} + \binom{n_2}{2} + \binom{n_3}{2} = z_K^{E_0}$, corresponding to partition $\{C_1\}, \{C_2\}, \{C_3\}$;
- $z(C_1 \cup C_2, C_3) = \binom{n_1 + n_2}{2} + \binom{n_3}{2}$, corresponding to partition $\{C_1 \cup C_2\}, \{C_3\}$;
- $z(C_1 \cup C_3, C_2) = \binom{n_1 + n_3}{2} + \binom{n_2}{2}$, corresponding to partition $\{C_1 \cup C_3\}, \{C_2\}$;
- $z(C_1, C_2 \cup C_3) = \binom{n_1}{2} + \binom{n_2 + n_3}{2}$, corresponding to partition $\{C_1\}, \{C_2 \cup C_3\}$.

Let $\mathscr{P}_{E_0}^K$ represent the set of all partitions of $N \backslash K$ such that two nodes connected by an edge in E_0 must belong to the same set. Associated to each partition $p \in \mathscr{P}_{E_0}^K$, we consider γ_p as the set of edges connecting pairs of nodes belonging to different sets in p and z_p its corresponding robustness value. The algorithm based on the set of components partitions is described in Algorithm 4.

Algorithm 4: Components separation

1: **Input:** $G = (N, E_0)$ and $c \in \{1, \ldots, |N|\}$
2: $\mathscr{K}_0 \leftarrow \left\{ K \in \mathscr{K} : z_{E_0}^K < \binom{n-c}{2} \right\}$
3: $s \leftarrow 1$
4: $(L_s, z_s) \leftarrow (0, \min\{z_{E_0}^K : K \in \mathscr{K}_0\})$
5: **while** $z_s < \binom{n-c}{2}$ **do**
6: $\quad s \leftarrow s + 1$
7: $\quad \Gamma \leftarrow \left\{ \gamma_p : p \in \mathscr{P}_{E_0}^K \text{ and } K \in \mathscr{K}_0 \text{ such that } z_p \leqslant z_{s-1} \right\}$
8: $\quad L_s \leftarrow$ **Cover**(Γ)
9: $\quad y^* \leftarrow$ optimal solution to **Cover**(Γ)
10: $\quad z_s \leftarrow$ **CND**$(E_0 \cup \{\{i,j\} \in E_n \backslash E_0 : y_{ij}^* = 1\})$
11: **end while**

Algorithm 4 is an extension of Algorithm 2 where the family of sets of edges Γ used to define the cover inequalities is computed in Line 7. In order to obtain this family Γ, initially, we need to compute every critical node set $K \in \mathscr{K}$ such that the remaining graph $G_{E_0}^K$ has multiple components (Line 2). Then, for each set $K \in \mathscr{K}_0$, the corresponding set of partitions $\mathscr{P}_{E_0}^K$ is defined. The family Γ is computed by including each partition p such that its robustness value z_p is not higher than the current threshold z_{s-1} (Line 7).

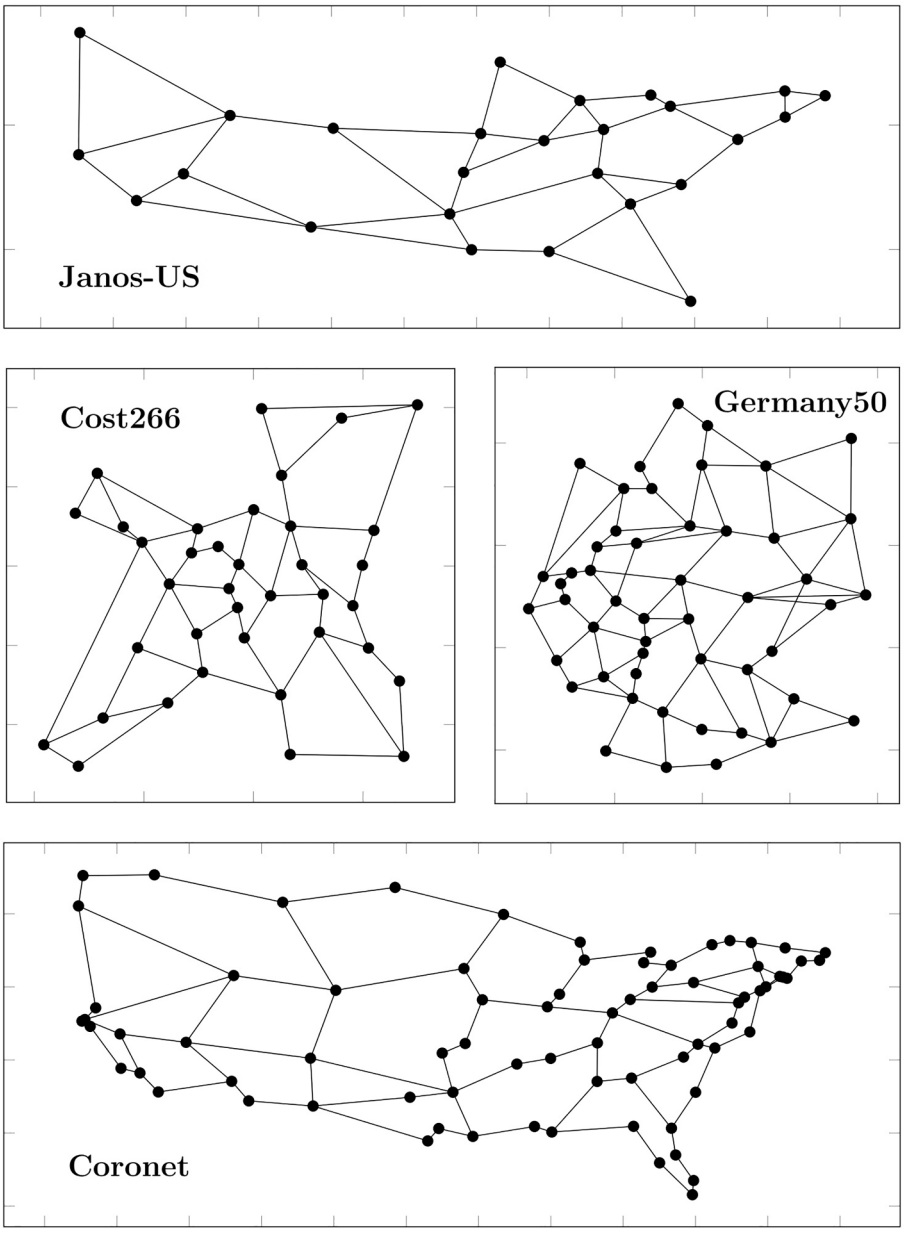

Fig. 3. Network topologies.

Table 1
Topology characteristics of each network.

Network	$\|N\|$	$\|E_0\|$	$\bar{\delta}$	L_0	$\bar{l}$	$\|E_n \setminus E_0\|$
Janos-US	26	42	3.23	25224	600.6	283
Cost266	37	57	3.08	24970	438.1	609
Germany50	50	88	3.52	8859	100.7	1137
Coronet	75	99	2.64	32642	329.7	3729

5. Computational results

All the computational tests reported in this section were obtained using the optimization software *Gurobi Optimizer* version 9.0.0, with programming language *Julia* version 1.4.1, running on a PC with an Intel Core i7-8700, 3.2 GHz and 16 GB RAM.

The main computational results are based on four telecommunication network topologies shown in Fig. 3: Janos-US, Cost266, Germany50 and Coronet. The information of their nodes (and their geographical locations) and edges is publicly available information (Orlowski et al., 2010).

In practice, the cost of a new edge between two given nodes in a telecommunication network requires the determination of the geographical route where the new edge is installed and this information is not available. However, there is a strong correlation between the distance between two network nodes and the cost of installing a new link connecting them and, thus, we assume that the cost l_{ij} of installing an edge between the two nodes i and j is given by the length (in kilometers) of the shortest path over the surface of a sphere representing Earth. Table 1 gives, for each network, the following topology characteristics: number of nodes $|N|$ and edges $|E_0|$ in the existing topologies, average node degree $\bar{\delta}$, total edge length of the original topology $L_0 = \sum_{(i,j)\in E_0} l_{ij}$, in kilometers, and average edge length $\bar{l}$. In addition, column '$|E_n \setminus E_0|$' represents the total number of candidate edges, i.e., the total number of binary variables y_{ij} in the proposed optimization models.

The remaining of this section is organized as follows. First, we report the results of the tests conducted with Algorithm 1 using formulation

Table 2
Results of Algorithm 1 using the **ESP**(.) model, considering Janos-US with $c = 2$.

L	z	$\lvert\mathscr{K}\rvert$	Rows (pre)	Rows (pos)	Total Runtime
0	181	0	–	–	0:00:01
1475	196	1	236968	172262	0:00:27
2357	213	3	710820	517198	0:02:36
2470	232	4	947746	688377	0:05:56
3940	253	8	1895450	1376043	0:26:18
4257	276	13	3080080	2235219	0:59:04

ESP(.) (Section 5.1) and show that only the smallest instance is solved with a runtime limit of 2 h. Second, we compare the two algorithms based on the Cover model and provide numerical results based on the four network topologies previously presented (Section 5.2). Then, we provide some additional insights on the Pareto frontier and on the performance of the two algorithms (Section 5.3). Finally, we test the proposed methodology on other well-known topologies in order to assess the scalability of both algorithms on larger graphs (Section 5.4), and analyze the effect of increasing the number of edges of the input topology (Section 5.5).

5.1. Testing Algorithm 1 using the path formulation **ESP**(.)

Table 2 presents the results for the Janos-US topology, with $c = 2$ critical nodes, obtained with Algorithm 1 where the selection of the edges is made by solving model **ESP**(.), as described in Section 3. This instance is the easiest one among all the instances reported in this paper.

Columns 'L' and 'z' represent the respective values of all the Pareto optimal solutions for this instance. Column '$|\mathscr{K}|$' denotes the number of critical node sets that need to be considered to obtain each Pareto optimal solution (L,z). Notice that there is no need to define constraints for scenario failures $K \in \mathscr{K}$ whose robustness value in the original graph $z_K^{E_0}$ is higher than the current threshold since these constraints are guaranteed by the original topology itself. Columns 'Rows (pre)' and 'Rows (pos)' represent the total number of constraints in the path formulation **ESP**(.), before and after the preprocessing phase performed by the solver with the default options, respectively. Finally, column 'Total Runtime' gives the accumulated computational time (in the format H:MM:SS).

These results show that the **ESP**(.) models solved in each iteration have a large number of active constraints. Consequently, Algorithm 1 took about 1 h to compute the Pareto frontier of this instance while the best proposed algorithm solves this instance in a second (results reported next). Moreover, this was the unique instance solved to optimality using the path formulation **ESP**(.) with the runtime limit of two hours. Henceforward, we will not report additional results using this model.

Table 3
Comparison between algorithms, considering Germany50 topology with $c = 4$ critical nodes.

		Algorithm 3			Algorithm 4		
L	z	No. ILPs	Rows	Runtime	No. ILPs	Rows	Runtime
0	640	–	–	0:00:01	–	–	0:00:05
54	650	1	1	0:00:01	1	1	0:00:01
125	675	2	3	0:00:02	1	4	0:00:01
219	702	9	12	0:00:08	1	44	0:00:02
244	731	3	15	0:00:03	1	93	0:00:02
288	762	3	18	0:00:03	1	106	0:00:02
407	795	3	21	0:00:05	1	117	0:00:03
545	830	15	36	0:00:31	1	169	0:00:03
673	864	5	41	0:00:16	1	204	0:00:04
723	867	4	45	0:00:11	1	227	0:00:04
900	904	14	59	0:00:47	1	305	0:00:06
941	906	5	64	0:00:23	1	393	0:00:07
1294	946	38	102	0:02:36	1	714	0:00:08
1442	947	23	125	0:02:07	1	910	0:00:11
2104	990	93	218	0:14:16	1	3641	0:00:43
4781	1035	500	718	1:53:01	1	15155	0:02:09

Table 4
Number of Pareto optimal solutions obtained and the total runtime used to get those solutions using Algorithm 3 (row generation).

Network	c	2	3	4	5	6
Janos-US	Pareto points	**6**	**10**	20	17	19
	Runtime	0:00:02	0:01:27	0:59:04	1:30:09	2:00:59
Cost266	Pareto points	**5**	**12**	16	25	26
	Runtime	0:00:07	0:02:57	0:19:05	3:58:25	3:57:26
Germany50	Pareto points	**3**	**7**	**16**	17	16
	Runtime	0:00:20	0:07:53	2:14:31	2:58:22	1:20:57
Coronet	Pareto points	**6**	12	32	27	41
	Runtime	0:06:04	1:32:37	1:00:47	4:39:11	2:28:08

5.2. Row generation algorithm vs components separation algorithm

In order to compare the performance of Algorithm 3 (row generation algorithm) with Algorithm 4 (components separation algorithm), Table 3 presents detailed results of running both algorithms considering the Germany50 topology with $c = 4$ critical nodes.

Once again, columns 'L' and 'z' represent the respective values of all Pareto optimal solutions for this instance. For each algorithm, column 'No. ILPs' represents the number of times that the Cover model was optimized to obtain each Pareto optimal solution, column 'Rows' gives the total number of cover constraints added to the ILP model and column 'Runtime' gives the computational time (in the format H:MM:SS) to obtain the solution of each iteration.

In this instance, both algorithms obtain the complete Pareto frontier. Algorithm 4 is much faster than Algorithm 3 in computing the complete Pareto frontier, despite the total number of cover inequalities generated by Algorithm 4 be much higher than the number generated by Algorithm 3. This is justified by the fact that Algorithm 4 needs to optimize only one ILP problem per each Pareto optimal solution.

Notice also that, in general, Algorithm 4 takes a higher running time to compute the first Pareto optimal solution. This is due to the fact that every node set $K \in \mathscr{K}$ needs to be processed at the initialization step (Line 2 of Algorithm 4), in order to compute the family of critical node sets $\mathscr{K}_0$ that divide the original topology in multiple connected components.

Next, we show the results of testing both algorithms on the four topologies and considering a number of critical nodes $c \in \{2, 3, 4, 5, 6\}$. Moreover, a runtime limit is imposed, forcing the algorithm to stop whenever an iteration takes more than 2 h to compute the next Pareto

Table 5
Number of Pareto optimal solutions obtained and the total runtime used to get those solutions using Algorithm 4 (components separation).

Network	c	2	3	4	5	6
Janos-US	Pareto points	**6**	**10**	**24**	25	26
	Runtime	0:00:01	0:00:03	0:02:40	4:22:50	1:52:21
Cost266	Pareto points	**5**	**12**	**20**	30	27
	Runtime	0:00:02	0:00:12	0:08:06	1:21:00	3:25:36
Germany50	Pareto points	**3**	**7**	**16**	21	19
	Runtime	0:00:05	0:00:48	0:03:50	0:37:28	2:34:58
Coronet	Pareto points	**6**	**13**	38	26	0
	Runtime	0:00:38	0:08:48	0:57:38	3:22:32	-

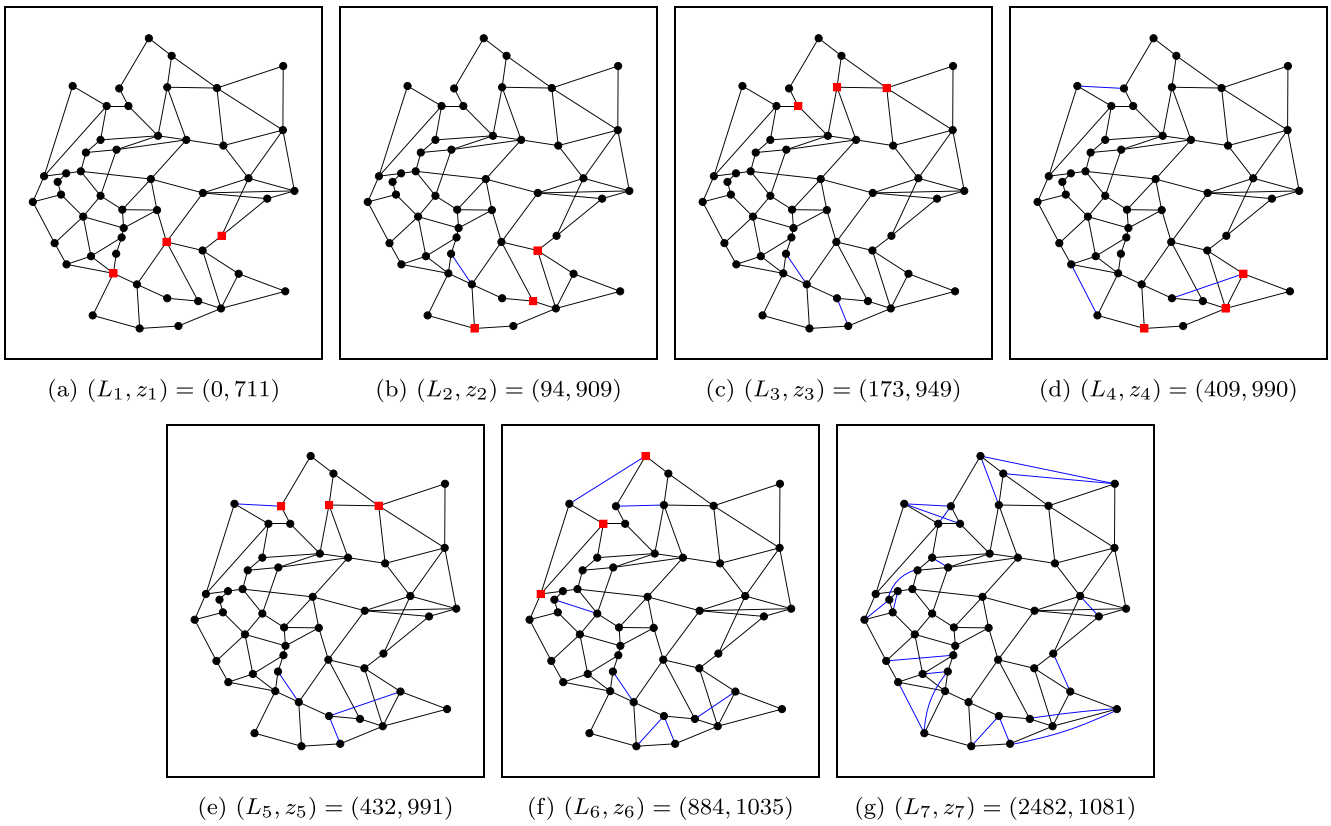

Fig. 4. Graphical illustration of the Pareto frontier for Germany50 topology and considering $c = 3$ critical nodes (optimal edges in blue and critical nodes in red squares).

optimal solution.

Tables 4 and 5 present the total number of Pareto optimal solutions obtained and the total computational time (in the format H:MM:SS) used to get those solutions using Algorithms 3 and 4, respectively. The instances for which each algorithm was able to compute the complete Pareto frontier are represented in bold. Moreover, whenever the algorithm was able to compute only a partial Pareto frontier, the real total computational time is the one reported in these tables plus 2 h.

These results show that, in general, Algorithm 4 is again much more time-efficient than Algorithm 3 when it is possible to compute the complete Pareto frontier within reasonable runtime. Additionally, there are three instances (Janos-US with $c = 4$, Cost266 with $c = 4$ and Coronet with $c = 3$) in which only Algorithm 4 was able to compute all Pareto optimal solutions.

Furthermore, when only a partial Pareto frontier was obtained, Algorithm 4 was able to obtain considerable more optimal solutions than Algorithm 3. There are two exceptions, i.e., two instances out of the 20 instances (4 topologies $\times$ 5 values of c) where Algorithm 3 performed better: Coronet topology for $c \in \{5, 6\}$. In these cases:

- for $c = 5$ critical nodes, Algorithm 4 exceeds the RAM limit when computing the 27$^{\text{th}}$ Pareto optimal solution;
- for $c = 6$ critical nodes, it is not possible to compute set $\mathscr{X}_0$ (Line 2 of Algorithm 4), i.e., processing the robustness value of each set of c nodes $K \in \mathscr{X}$ is not doable within the imposed runtime of two hours.

These two instances give us an indication of the scalability limit of Algorithm 4 (components separation). Conversely, as Algorithm 3 (row generation) does not require any prepossessing to obtain the initial Pareto optimal solutions, this algorithm is able to obtain a partial Pareto frontier in a wider range of input topologies.

5.3. Insights on the computational results

Next, we provide additional insights on the Pareto frontier of these instances.

First, recall that the robustness value of a solution is the number of node pairs that can still communicate if the critical nodes of the topology are deleted. Although this number can theoretically be any value between 0 and $\binom{n-c}{2}$, where n is the total number of nodes and c is the number of critical nodes, only a subset of these values can represent the robustness value of a Pareto optimal solution.

To illustrate this fact, Fig. 4 presents a graphical representation of all Pareto-optimal topologies for Germany50 considering $c = 3$ critical nodes. The robustness value z_s of each topology is given by:

(a) $z_1 = \binom{37}{2} + \binom{10}{2} = 666 + 45 = 711$, that corresponds to two connected components with 37 and 10 nodes, respectively;

(b) $z_2 = \binom{43}{2} + \binom{4}{2} = 903 + 6 = 909$, that corresponds to two connected components with 43 and 4 nodes, respectively;

(c) $z_3 = \binom{44}{2} + \binom{3}{2} = 946 + 3 = 949$, that corresponds to two connected components with 44 and 3 nodes, respectively;

(d) $z_4 = \binom{45}{2} + \binom{1}{2} + \binom{1}{2} = 990 + 0 + 0 = 990$, that corresponds to a connected component with 45 nodes and two isolated nodes;

(e) $z_5 = \binom{45}{2} + \binom{2}{2} = 990 + 1 = 991$, that corresponds to two connected components with 45 and 2 nodes, respectively;

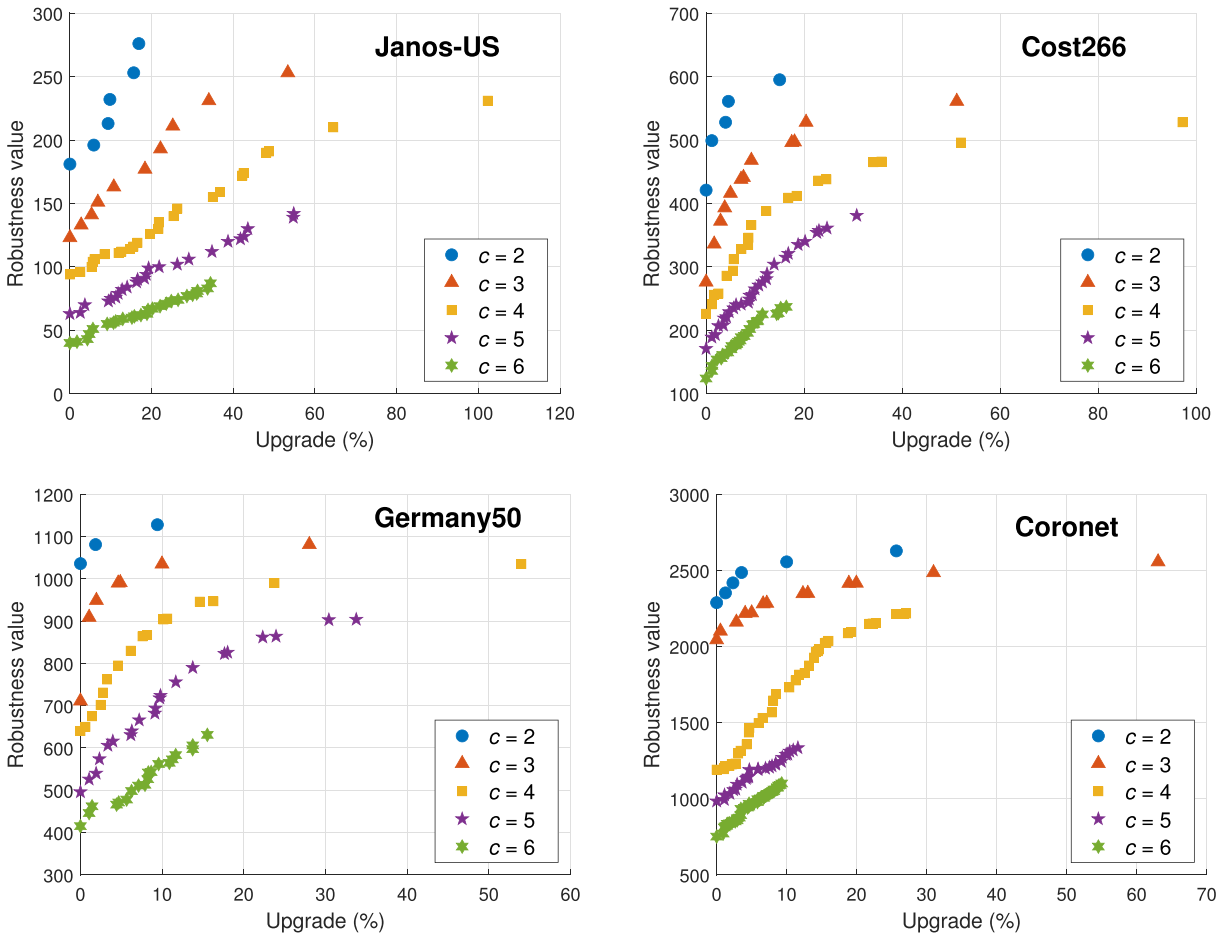

Fig. 5. Scatter plots of all obtained Pareto optimal solution values.

(f) $z_6 = \binom{46}{2} + \binom{1}{2} = 1035 + 0 = 1035$, that corresponds to a connected component with 46 nodes and an isolated node;

(g) $z_7 = \binom{n-c}{2} = \binom{47}{2} = 1081$, that corresponds to the upper bound scenario where the upgraded topology is fully robust to any failure of 3 nodes.

In general, for a topology with $|N| = 50$ nodes and considering $c = 3$ critical nodes, it is impossible to obtain an upgraded topology with a robustness value $z \geqslant 946$ that does not belong to the set $\{946, 947, 949, 990, 991, 1035, 1081\}$ since these values represent all possible alternatives of separating a maximum of 3 nodes from the main component.

Next, in Fig. 5, we represent all Pareto optimal values obtained (using Algorithm 3 or Algorithm 4, depending on which one obtained a higher number of Pareto points) with a set of scatter plots, one for each tested topology, with the robustness value (objective function value of the CND problem) as function of the edge upgrade cost percentage, i.e., $\frac{L}{L_0} \times 100(\%)$. This represents the percentage of additional edge length added to the original topology (N, E_0).

As expected, from these scatter plots we can observe that there is no cross-over between Pareto frontiers for different numbers of critical nodes c, i.e., for similar upgrade percentages, the robustness value decreases with the increase of the number of critical nodes.

For $c \in \{2, 3, 4\}$ (with the exception of Coronet with $c = 4$ instance), the complete Pareto frontier of each instance is obtained. We observe that, in general, the last points represent a much higher edge length increase with a smaller robustness value increase than the previous points. This shows that, in general, we need smaller upgrade costs to reach higher robustness gains in the first Pareto optimal solutions and, when reaching the last Pareto optimal solutions, we need higher additional cost to reach full (or near full) robust solutions.

For $c \in \{5, 6\}$, these plots show how incomplete partial Pareto frontiers obtained are. For topologies representing real-world optical networks, the proposed algorithms do not obtain the complete Pareto frontier for more than $c = 4$ critical nodes. In practice, though, a partial Pareto frontier may be enough as, in general, upgrading a topology to have a high robustness value for large values of c implies incurring in huge costs.

Finally, across all tested instances, these Pareto frontiers show that it is within the initial 20% of edge upgrade cost that occurs the highest improvement in the robustness value of each topology. To further analyze this observation, Fig. 6 represents each Pareto frontier obtained, in a stair plot format, up to 20% edge upgrade and considering the robustness value as a percentage of the upper bound $\binom{n-c}{2}$.

In these plots, we observe that the highest percentage increase in the robustness value of the topology (N, E_0) for a failure of $c \in \{4, 5, 6\}$ critical nodes occurs within the initial 20% of edge upgrade. Moreover, across the majority of tested instances, we observe that the initial 5% of edge upgrade provides the highest percentage of the robustness value to critical node failures for $c \in \{2, 3\}$.

There exists a clear exception to this trend, which is Janos-US topology with $c = 2$ critical nodes (instance presented in Table 2). In order to understand this case, Fig. 7 presents a graphical representation of all topologies, each one corresponding to a Pareto optimal solution. Analyzing these solutions, in order to upgrade the original topology (a)

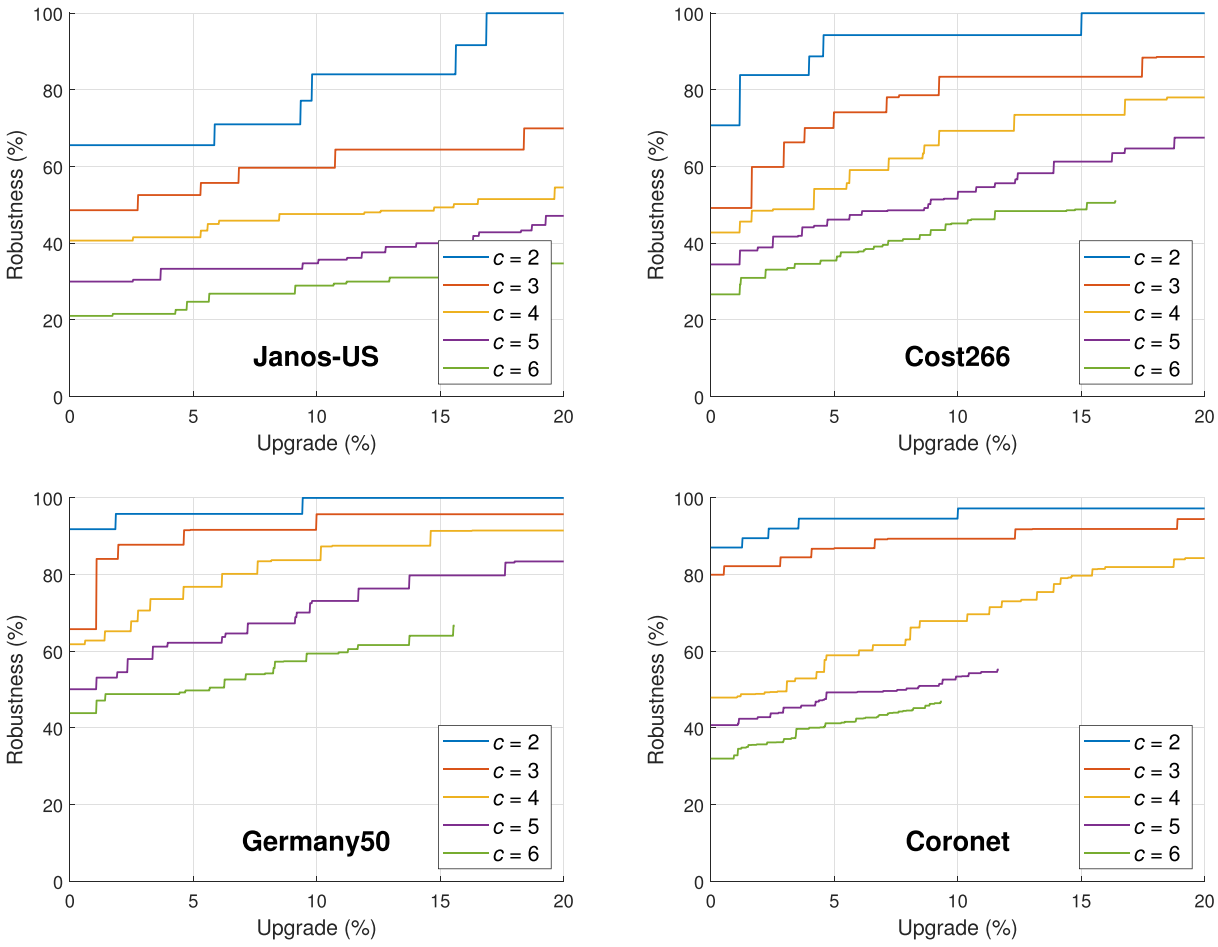

Fig. 6. Stair plots of the Pareto frontiers obtained.

to one with higher robustness for $c = 2$ node failures, the upgraded topologies require the addition of at least one long edge across the network (compared with the average edge length of the original topology). This explains why this topology has different upgrade trends when compared to the other three topologies.

5.4. Testing other topologies and larger sizes

In this section, we present the results of testing the proposed algorithms on different topologies and, in order to assess the scalability of the methods, on larger graph sizes. We generated a set of 9 distinct topologies based on three well-known graph generation algorithms: Erdos-Renyi model (Erdös and Rényi, 1959) (Fig. 8), Watts-Strogatz small-world model (Watts and Strogatz, 1998) (Fig. 9) and Barabasi-Albert scale-free model (Barabási et al., 1999) (Fig. 10). In the process of generating these topologies, whenever a topology is not connected, it is discarded and a different topology is generated. This ensures that only connected topologies are considered.

In order to test Algorithms 3 and 4 with these graphs, we considered a number of critical nodes $c \in \{2, 3, 4, 5\}$. Moreover, the previous runtime limit was again considered (the algorithm stops whenever an iteration reaches a 2 h runtime to compute the next Pareto solution). Finally, we set unitary costs of installing new edges, i.e., $l_{ij} = 1$ for each $(i, j) \in E_n$, which means that the objective function (1) corresponds to minimizing the total number of additional edges.

Tables 6 and 7 present the results summary (similar to Tables 4 and 5) of the Erdos-Renyi topologies using Algorithms 3 and 4, respectively. For these instances, the complete Pareto frontier was obtained only when $c = 2$ critical nodes were considered (using Algorithm 4). Moreover, we observe that the number of Pareto optimal solutions obtained decreases with the increase of the number of nodes $|N|$. This fact gives us an indication that the partial Pareto frontier obtained tends to be more incomplete with the increase of the number of nodes of the input topology.

Next, Tables 8 and 9 present the results summary of the Watts-Strogatz small-world topologies using Algorithms 3 and 4, respectively. The computational results obtained with the Watts-Strogatz topologies are similar to the Erdos-Renyi topologies previously presented, with the main difference that Algorithm 4 is able to compute complete Pareto frontiers for $c = 3$ critical nodes with $|N| \leqslant 100$.

Regarding both algorithms, for the largest topologies (i.e., when $|N| = 120$), the results show that: on one hand, Algorithm 3 is barely able to compute any Pareto optimal solutions (besides the trivial one); on the other hand, Algorithm 4 is only able to compute solutions for $c \in \{2, 3\}$ (due to the 2 h runtime limit constraint imposed to the preprocessing procedure of this algorithm). Therefore, although both algorithms have pros and cons when applied to topologies with $|N| \leqslant 100$, for larger topologies, the proposed methodology is not effective to compute the Pareto frontier.

Finally, Table 10 presents the results obtained using Algorithm 3 to the three topologies randomly generated using the Barabasi-Albert scale-free model. These results are quite straightforward. In all instances, Algorithm 3 is only able to compute two Pareto optimal solutions: the trivial solution (i.e. $L_1 = 0$) and the optimal Pareto pair that corresponds to adding only one new edge (i.e. $L_2 = 1$). This algorithm cannot compute the Pareto optimal solution with two (or more) additional edges within the runtime limit for any of the topologies generated with the Barabasi-Albert model.

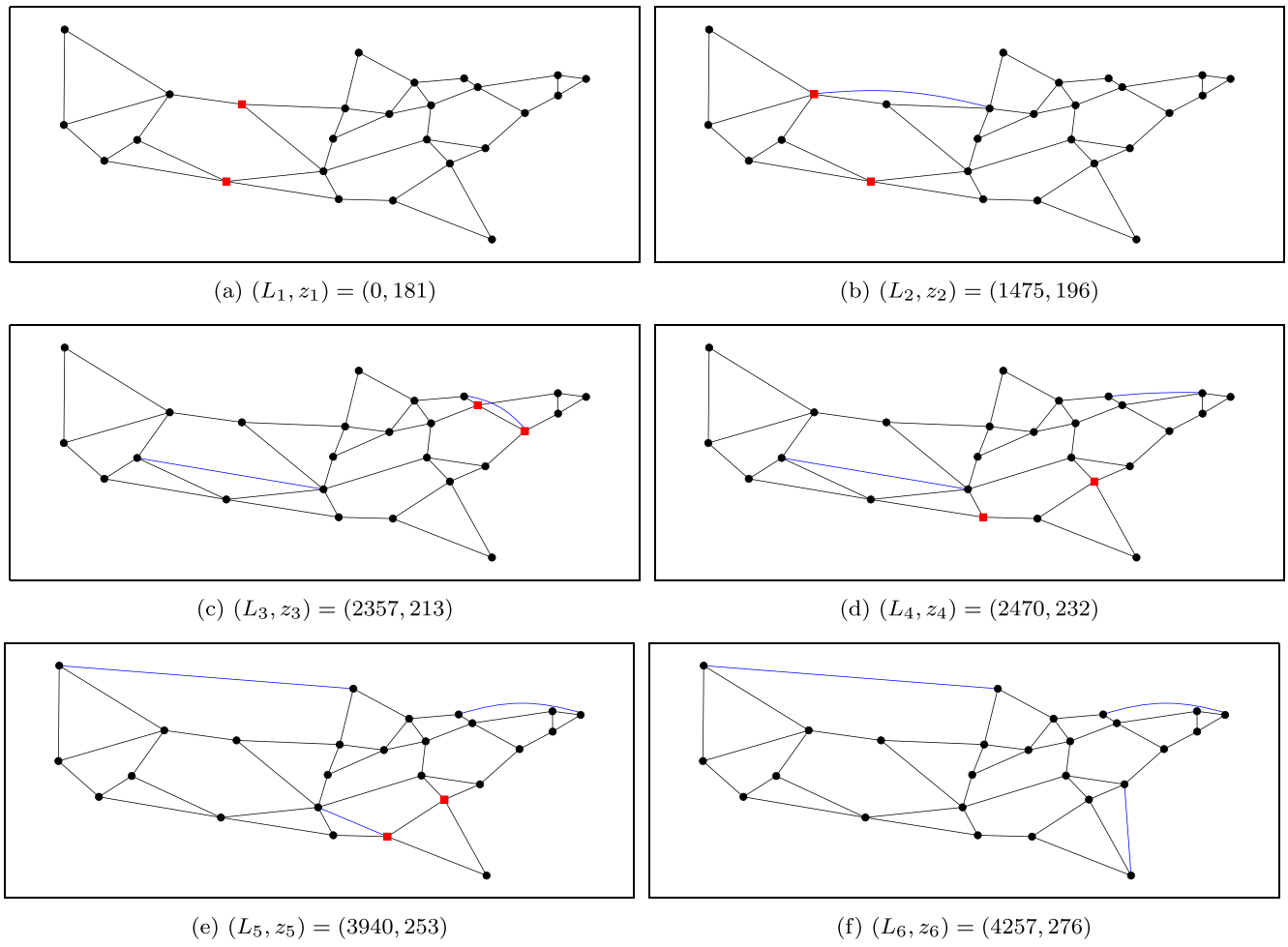

Fig. 7. Graphical representation of the Pareto frontier solutions for Janos-US topology and considering $c = 2$ critical nodes (optimal edges in blue and critical nodes in red squares).

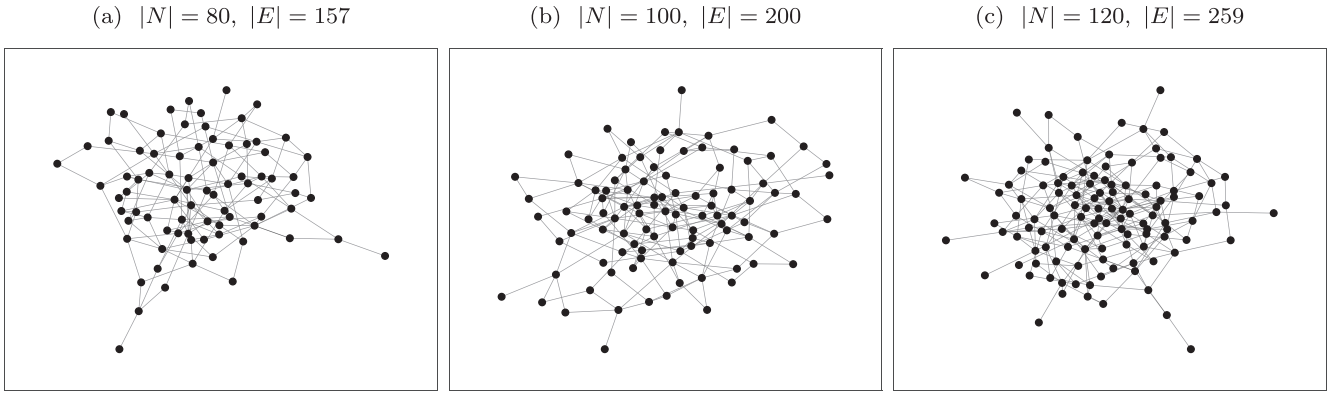

Fig. 8. Erdos-Renyi randomly generated topologies with $|N| \in \{80, 100, 120\}$, considering a probability of selection of each edge of $p = 0.04$.

Moreover, we do not present the computational results using Algorithm 4 because this algorithm is not able to compute any Pareto optimal solution for these topologies (besides the trivial one). To understand the reason for this fact, consider the simplest instance (i.e., $c = 2$ critical nodes and the Barabasi-Albert topology with $|N| = 80$ nodes). When removing the critical nodes from this topology, it results in a remaining graph with 25 distinct components. Given that this algorithm is based on computing the partition set of all possible critical node sets, for this specific CND solution only, the Bell number of 25 is, approximately, 4.6×10^{18} (represents the cardinality of the partition set). Since Algorithm 4 requires to process all components partitions, this is unworkable due to both time and memory constraints.

5.5. *Testing the effect of increasing the number of edges*

Here, for a fixed number of nodes, we present the results of testing Algorithm 4 (components separation), which from the previous results is the best procedure, on graphs with different number of edges. To

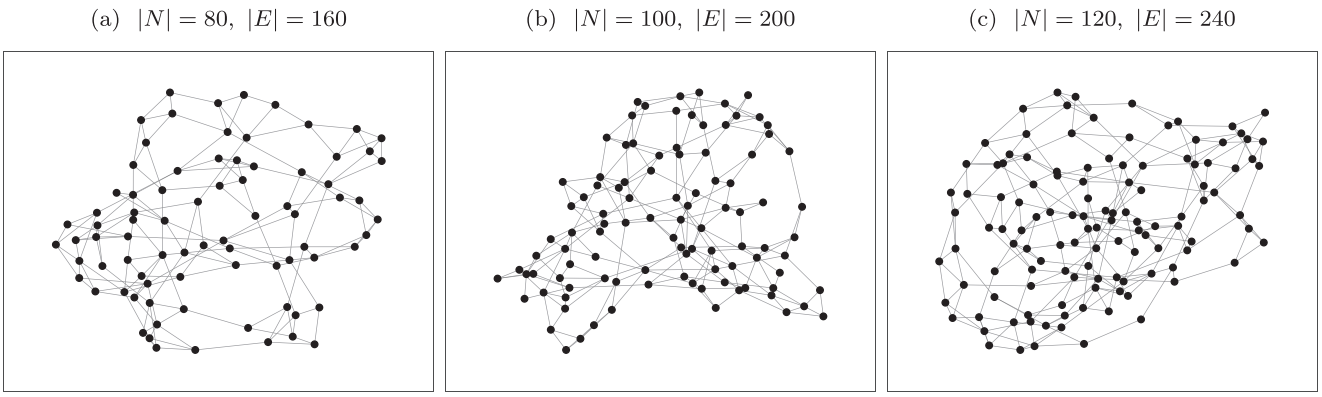

Fig. 9. Watts-Strogatz randomly generated topologies with $|N| \in \{80,100,120\}$, considering a total number of edges $|E| = 2|N|$ and a rewiring probability of $\beta = 0.2$.

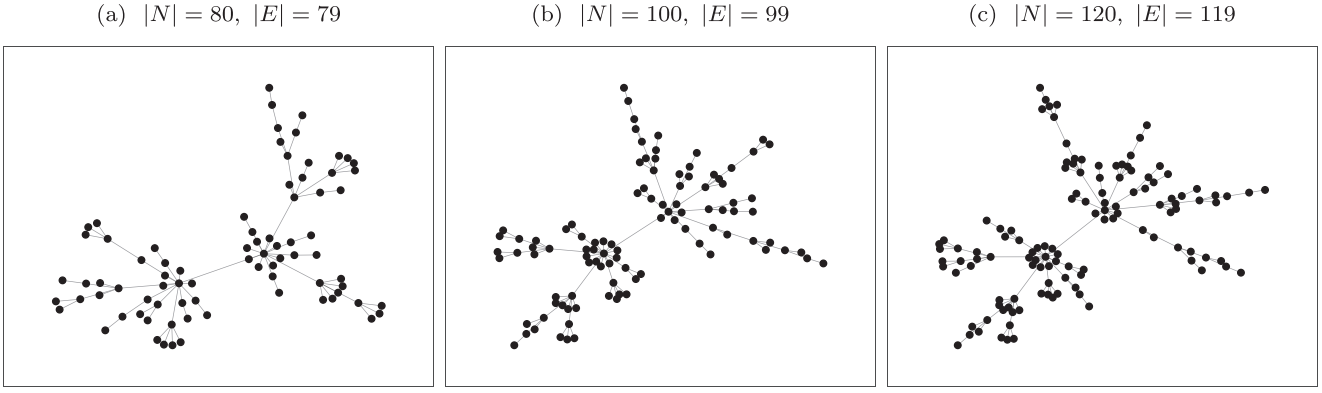

Fig. 10. Barabasi-Albert randomly generated topologies with 80, 100 and 120 nodes, respectively.

perform these tests, we considered the Erdos-Renyi model with $|N| = 50$ nodes and with a probability of selection of each edge of p that ranges from 0.05 to 0.11. Moreover, we have considered a fixed seed to this generation of topologies, in order to make each newly generated topology an upgrade of the previous one, i.e., if an edge is in the graph generated with a given p, then it is also in all the graphs generated with higher values for p. We tested different seeds until one of them ensured that the topology generated with $p = 0.05$ was connected (and by consequence, all topologies with $p > 0.05$).

In Table 11, we present the results obtained with Algorithm 4 (components separation) for each topology for $c \in \{2, 3, 4\}$ critical nodes considering, once again, unitary costs and the 2 h stopping criteria. In the first two lines of this table, we present each tested probability p and the number of edges $|E|$ of the corresponding topology. In addition to the number of Pareto optimal solutions obtained, and the total runtime required to obtain all solutions found, we present the robustness value (in percentage to the upper bound) of each initial topology z_1 and the robustness value of the last Pareto optimal point computed z_{last}. Notice that connectivity robustness of 100% means that Algorithm 4 was able to compute the complete Pareto frontier of that instance.

Contrary to the effect of increasing the number of nodes, by increasing the number of edges of the input topology, the algorithm performs better. This can be easily explained by the following two related facts. By increasing the number of edges $|E|$, there is a tendency to increase the robustness value of the input topology z_1, and to decrease the total number of different combinations of c critical nodes that split the network into disjoint components. This causes a reduction in the number of Pareto optimal solutions, and therefore, the proposed approaches require fewer iterations to obtain the Pareto frontier.

6. Conclusions

In this work, we have addressed the robust network upgrade problem (RNUP) that aims to identify a set of new edges to add to the original topology in order to increase its robustness to simultaneous node failures. This problem is formulated as a bi-objective MILP problem with two distinct objectives: minimizing the total cost of the new edges, and

Table 6
Erdos-Renyi results summary, i.e., number of Pareto optimal solutions obtained and the total runtime used to get those solutions using Algorithm 3 (row generation).

$\|N\|$	c	2	3	4	5
80	No. Pareto points	**5**	7	7	5
	Runtime	1:17:51	2:17:58	2:56:10	4:45:46
100	No. Pareto points	4	5	4	3
	Runtime	0:09:49	1:03:29	2:38:42	4:56:10
120	No. Pareto points	3	2	1	3
	Runtime	1:18:10	0:41:31	0:16:41	3:00:39

Table 7
Erdos-Renyi results summary, i.e., number of Pareto optimal solutions obtained and the total runtime used to get those solutions using Algorithm 4 (components separation).

$\|N\|$	c	2	3	4	5
80	No. Pareto points	**5**	7	9	0
	Runtime	0:01:13	0:03:20	0:18:35	–
100	No. Pareto points	**5**	7	7	0
	Runtime	0:07:19	0:24:56	0:44:20	–
120	No. Pareto points	**4**	5	0	0
	Runtime	0:21:26	0:26:33	–	–

Table 8
Watts-Strogatz results summary, i.e., number of Pareto optimal solutions obtained and the total runtime used to get those solutions using Algorithm 3 (row generation).

$\|N\|$	c	2	3	4	5
80	No. Pareto points	**2**	2	5	3
	Runtime	0:01:27	0:10:22	0:59:09	0:28:01
100	No. Pareto points	**2**	3	4	4
	Runtime	0:18:02	1:27:28	2:58:45	2:55:03
120	No. Pareto points	**2**	2	2	1
	Runtime	0:45:46	1:55:36	0:54:06	0:54:33

Table 9
Watts-Strogatz results summary, i.e., number of Pareto optimal solutions obtained and the total runtime used to get those solutions using Algorithm 4 (components separation).

$\|N\|$	c	2	3	4	5
80	No. Pareto points	**2**	**3**	7	0
	Runtime	0:00:50	0:16:56	2:25:38	–
100	No. Pareto points	**2**	**4**	7	0
	Runtime	0:04:36	1:47:55	3:38:19	–
120	No. Pareto points	**2**	2	0	0
	Runtime	0:19:43	0:40:28	–	–

maximizing the robustness of the resulting upgraded topology. As robustness metric, we have considered the objective function value of the CND problem which measures the pairwise connectivity between nodes when a set of c critical nodes are removed from the graph.

A general algorithm was presented to obtain the Pareto frontier where the robustness value is obtained by solving an ILP problem and the selection of the new edges is obtained solving a path formulation adapted from the bi-objective MILP problem. Since using this approach can only solve the smallest instances, we also presented an alternative formulation by modeling the selection of edges as a set covering problem. As the number of cover inequalities increases exponentially with the size of the instance, we proposed two algorithms to select the cover inequalities. One is a row generation algorithm that iteratively selects cover inequalities, and the other is a components separation algorithm that selects simultaneously all the cover inequalities that force the connection of different components in order to obtain a desire robustness value.

The computational tests have shown that the components separation algorithm is much more time-efficient than the row generation algorithm when it is possible to compute the complete Pareto frontier within a reasonable running time. In the telecommunication topologies, it was possible to obtain the complete Pareto frontier for all four tested topologies with $c \in \{2, 3\}$ critical nodes and for the Janos-US, Cost266 and Germany50 topologies with $c = 4$ critical nodes. Nevertheless, the components separation algorithm has scalability issues when considering a higher number of critical nodes. In contrast, the row generation algorithm is able to obtain a partial Pareto frontier for a wider range of instances. For example, it was able to compute 41 Pareto optimal solutions considering the Coronet topology with $c = 6$ critical nodes.

Finally, although both algorithms have advantages and disadvantages, when considering input topologies with larger sizes (more than 100 nodes), both algorithms present scalability issues. In this work, we have addressed the RNUP with exact procedures. For larger graphs, heuristic approaches have to be considered, aiming to obtain an approximation of the Pareto frontier.

CRediT authorship contribution statement

Fábio Barbosa: Conceptualization, Methodology, Software, Formal analysis, Visualization, Writing - review & editing. **Agostinho Agra:** Supervision, Conceptualization, Methodology, Writing - review & editing. **Amaro Sousa:** Supervision, Conceptualization, Methodology, Writing - review & editing.

Table 10
Barabasi-Albert results summary, i.e., number of Pareto optimal solutions obtained and the total runtime used to get those solutions using Algorithm 3 (row generation).

$\|N\|$	c	2	3	4	5
80	No. Pareto points	2	2	2	2
	Runtime	0:04:26	0:04:17	0:15:12	0:21:50
100	No. Pareto points	2	2	2	2
	Runtime	0:08:25	0:42:38	0:40:23	1:37:39
120	No. Pareto points	2	2	2	2
	Runtime	0:12:45	1:36:02	3:55:45	5:18:05

Table 11
Results summary of increasing the number of edges using the Erdos-Renyi generation model (with probability $p \in \{0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.11\}$) and Algorithm 4 (components separation).

c	p	0.05	0.06	0.07	0.08	0.09	0.10	0.11
	$\|E\|$	65	74	90	103	113	124	141
2	No. Pareto sol.	**8**	**6**	**5**	**3**	**3**	**2**	**2**
	Runtime	0:00:12	0:00:09	0:00:08	0:00:06	0:00:05	0:00:04	0:00:03
	z_1 (%)	80.2	87.8	87.8	91.8	91.8	95.8	95.8
	z_{last} (%)	100	100	100	100	100	100	100
3	No. Pareto sol.	9	7	6	**5**	**5**	**3**	**2**
	Runtime	0:57:25	0:00:25	0:00:31	0:00:31	0:00:25	0:00:19	0:00:12
	z_1 (%)	59.3	68.4	79.7	87.5	87.5	91.6	95.7
	z_{last} (%)	87.8	91.7	95.7	100	100	100	100
4	No. Pareto sol.	6	8	8	6	5	4	**3**
	Runtime	2:02:24	2:01:26	0:04:41	0:00:54	0:00:30	0:00:22	0:01:46
	z_1 (%)	48.0	62.8	75.5	83.3	87.2	87.2	91.4
	z_{last} (%)	73.0	87.3	91.5	95.7	95.7	95.7	100

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This work was supported by ERDF Funds through the Centre's Regional Operational Program and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia, I.P. under the project ResNeD CENTRO-01-0145-FEDER-029312. First author was also supported by FCT through PhD grant SFRH/BD/132650/2017.

References

Arulselvan, A., Commander, C.W., Elefteriadou, L., Pardalos, P.M., 2009. Detecting critical nodes in sparse graphs. Comput. Oper. Res. 36 (7), 2193–2200.

Barabási, A., Albert, R., 1999. Emergence of scaling in random networks. Science 286 (5439), 509–512.

Barbosa, F., de Sousa, A., Agra, A., 2018. Topology design of transparent optical networks resilient to multiple node failures. In: 2018 10th International Workshop on Resilient Networks Design and Modeling (RNDM). pp. 1–8.

Barbosa, F., de Sousa, A., Agra, A., 2020. Design/upgrade of a transparent optical network topology resilient to the simultaneous failure of its critical nodes. Networks 75 (4), 356–373.

Caballero, R., Gómez, T., Luque, M., Miguel, F., Ruiz, F., 2002. Hierarchical generation of pareto optimal solutions in large-scale multiobjective systems. Comput. Oper. Res. 29 (11), 1537–1558.

Chvatal, V., 1979. A greedy heuristic for the set-covering problem. Math. Oper. Res. 4 (3), 233–235.

de Sousa, A., Santos, D., 2020. In: Rak, Jacek, Hutchison, David (Eds.), Vulnerability evaluation of networks to multiple failures based on critical nodes and links, Guide to Disaster-Resilient Communication Networks. Springer International Publishing, Cham, pp. 63–86.

de Sousa, A., Mehta, D., Santos, D., 2017. The robust node selection problem aiming to minimize the connectivity impact of any set of p node failures. In: DRCN 2017 – Design of Reliable Communication Networks; 13th International Conference. pp. 1–8.

de Sousa, A., Piccini, J., Robledo, F., Romero, P., 2019. An interplay between critical node detection and epidemic models. In: 2019 11th International Workshop on Resilient Networks Design and Modeling (RNDM), pp. 1–7.

Dinh, T.N., Xuan, Y., Thai, M.T., Park, E.K., Znati, T., 2010. On approximation of new optimization methods for assessing network vulnerability. 2010 Proceedings IEEE INFOCOM 1–9.

Di Summa, M., Grosso, A., Locatelli, M., 2011. Complexity of the critical node problem over trees. Comput. Oper. Res. 38 (12), 1766–1774.

Di Summa, M., Grosso, A., Locatelli, M., 2012. Branch and cut algorithms for detecting critical nodes in undirected graphs. Comput. Optim. Appl. 53 (3), 649–680.

Erdös, P., Rényi, A., 1959. On random graphs i. Publicationes Mathematicae Debrecen 6, 290–297.

Faramondi, L., Oliva, G., Panzieri, S., Pascucci, F., Schlueter, M., Munetomo, M., Setola, R., 2019. Network structural vulnerability: a multiobjective attacker perspective. IEEE Trans. Syst. Man Cybern.: Syst. 49 (10), 2036–2049.

Furdek, M., Wosinska, L., Goscien, R., Manousakis, K., Aibin, M., Walkowiak, K., Ristov, S., Gushev, M., Marzo, J.L., 2016. An overview of security challenges in communication networks. In: 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM), pp. 43–50.

Gomes, T., Tapolcai, J., Esposito, C., Hutchison, D., Kuipers, F., Rak, J., de Sousa, A., Iossifides, A., Travanca, R., André, J., Jorge, L., Martins, L., Ugalde, P.O., Pašic, A., Pezaros, D., Jouet, S., Secci, S., Tornatore, M., 2016. A survey of strategies for communication networks to protect against large-scale natural disasters. In: 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM). pp. 11–22.

Hadian, A., Bagherian, M., 2021. A Pareto frontier for node survivable computer network design problem. Telecommun. Syst. 76, 371–389.

Lalou, M., Amin Tahraoui, M., Kheddouci, H., 2016. Component-cardinality-constrained critical node problem in graphs. Discrete Appl. Math. 210, 150–163.

Lalou, M., Amin Tahraoui, M., Kheddouci, H., 2018. The critical node detection problem in networks: a survey. Comput. Sci. Rev. 28, 92–117.

Li, J., Pardalos, P.M., Xin, B., Chen, J., 2019. The bi-objective critical node detection problem with minimum pairwise connectivity and cost: theory and algorithms. Soft Comput. 23, 12729–12744.

Natalino, C., Yayimli, A., Wosinska, L., Furdek, M., 2019. Infrastructure upgrade framework for content delivery networks robust to targeted attacks. Opt. Switching Networking 31, 202–210.

Orlowski, S., Wessäly, R., Pióro, M., Tomaszewski, A., 2010. SNDlib 1.0 – survivable network design library. Networks 55 (3), 276–286.

Pavlikov, K., 2018. Improved formulations for minimum connectivity network interdiction problems, Computers <texmath type="inline">&</texmath>. Oper. Res. 97, 48–57.

Purevsuren, D., Cui, G., 2019. Efficient heuristic algorithm for identifying critical nodes in planar networks. Comput. Oper. Res. 106, 143–153.

Rak, J., Hutchison, D., July 2020. Guide to disaster-resilient communication networks. Computer Communications and Networks, Springer International Publishing, Cham.

Santos, D., de Sousa, A., Monteiro, P., 2018. Compact models for critical node detection in telecommunication networks. Electron. Notes Discrete Math. 64, 325–334.

Shen, S., Smith, J.C., Goli, R., 2012. Exact interdiction models and algorithms for disconnecting networks via node deletions. Discrete Optim. 9 (3), 172–188.

Shukla, P.K., Deb, K., 2007. On finding multiple pareto-optimal solutions using classical and evolutionary generating methods. Eur. J. Oper. Res. 181 (3), 1630–1652.

Ventresca, M., 2012. Global search algorithms using a combinatorial unranking-based problem representation for the critical node detection problem. Comput. Oper. Res. 39 (11), 2763–2775.

Ventresca, M., Robert Harrison, K., Ombuki-Berman, B.M., 2018. The bi-objective critical node detection problem. Eur. J. Oper. Res. 265 (3), 895–908.

Veremyev, A., Boginski, V., Pasiliao, E.L., 2014. Exact identification of critical nodes in sparse networks via new compact formulations. Optim. Lett. 8 (4), 1245–1259.

Watts, D., Strogatz, S., 1998. Collective dynamics of 'small-world' networks. Nature 393, 440–442.