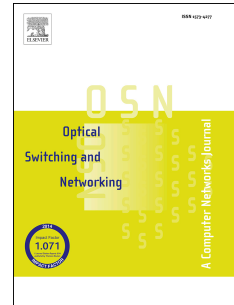# Accepted Manuscript

Minimization of the network availability upgrade cost with geodiverse routing for disaster resilience

Amaro de Sousa, Teresa Gomes, Rita Girão-Silva, Lúcia Martins

Minimization of the network availability upgrade cost with geodiverse routing for disaster resilience

Amaro de Sousa[a,b,*], Teresa Gomes[c,d], Rita Girão-Silva[c,d], Lúcia Martins[c,d]

[a]*Departamento de Eletrónica, Telecomunicações e Informática,*
Universidade de Aveiro, 3810-193 Aveiro, Portugal
[b]*Instituto de Telecomunicações, 3810-193 Aveiro, Portugal*
[c]*Department of Electrical and Computer Engineering,*
University of Coimbra, 3030-290 Coimbra, Portugal
[d]*INESC Coimbra, 3030-290 Coimbra, Portugal*

Abstract

Telecommunication networks are a critical infrastructure of our society. Wide area backbone communication networks are based on optical networks, where each fiber has a very large capacity. These networks must offer high end-to-end availability and a high resilience to large-scale disasters. Routing with geodiversity can mitigate the impact of disasters but will result in longer paths, making it difficult to achieve the availability levels required by critical services. In this paper, we consider a given core optical network such that the current availability and the cost of upgrading it to a higher value are known for each network link (or edge). Then, the problem of selecting a set of edges to be upgraded at a minimum cost, while guaranteeing desired values of end-to-end availability and geodiversity, is considered and formulated as an arc based integer non-linear programming model. The non-linear constraints of the model are approximated and linearized, resulting in a new ILP based heuristic. A filtering procedure is proposed for decreasing (if possible) the cost and the number of upgraded edges of the solutions obtained by previously proposed greedy heuristics and also by the ILP based heuristic. The relative performance of the heuristics is evaluated using different geodiverse distances and end-to-end availability values in two reference core optical networks.

Keywords: availability, geodiverse routing, resilience, disaster

*Corresponding author
Email addresses: `asou@ua.pt` (Amaro de Sousa), `teresa@deec.uc.pt` (Teresa Gomes), `rita@deec.uc.pt` (Rita Girão-Silva), `lucia@deec.uc.pt` (Lúcia Martins)

1. Introduction

Telecommunication networks are currently a key infrastructure upon which our society depends and critical services are expected to be provided with high availability. On the other hand, large-scale disasters are becoming more frequent in time and wider in scope, drastically degrading the services supported by telecommunication networks. To cope with such trends, telecommunication networks must evolve to become both highly available and resilient to disasters. This is of particular importance in core networks, based on optical technologies, which typically cover wide areas with large geographical distances between switching nodes. In these cases, end-to-end availability is harder to achieve (availability is lower for larger link lengths) and disasters have more probability of affecting a significant part of the network but still leaving many surviving nodes without service.

In this work, a topology of a given core optical network is considered such that all geographical distances between any two network elements (nodes or fiber links) are known. In this context, the nodes represent Dense Wavelength Division Multiplexing (DWDM) access nodes with reconfigurable optical add-drop multiplexers or optical cross-connects and the links represent optical transmission systems which consist of long haul fibers and optical amplifiers. Different measures may be used for assessing the geographical diversity of routes [1]. We consider the path geodiversity strategy proposed in [2, 3], which assumes that a disjoint pair of routing paths is defined for each source-destination pair of nodes, where the two routing paths must be geographically separated by a minimum distance of D. The aim is that a disaster with a geographical coverage of diameter lower than D may affect intermediate elements of one path without affecting any element of the other path. For larger values of D, the paths are more distant from each other and therefore, the network tends to be more resilient to wider area disasters. However, this leads to longer paths and to a decrease in the end-to-end availability. These two aspects are conflicting, which increases the difficulty in handling the problem of jointly providing a high end-to-end availability and wide-area disaster resilience.

Assuming that the current availability and the cost of upgrading it to a higher value are known for each network link (or edge), the problem addressed in this paper is the selection of edges whose availability should be upgraded, so that a desired value of availability and geodiversity (leading to a wide-area disaster resilient network) can be achieved for a set of source-destination pairs. A recent work dealing with the selection of edges to be upgraded taking into consideration the impact of

2

geographically correlated failures is [4] but no availability requirements are considered. The joint consideration of availability and disaster resilience turns the problem into a non-linear one which is much harder to solve than the previous problem variants.

Requirements for end-to-end availability and geodiversity are defined for each source-destination pair in the upgraded network: (i) at least one path pair with a minimum target availability of Λ must be provided to each node pair; (ii) that path pair must satisfy a target geodiversity value D in order to guarantee a certain disaster area resilience. Note however that the locations of the nodes and the edges along each geographical path may preclude the fulfillment of the target geodiversity value. Therefore, defining D_{st}^{Max} as the maximum geodiversity value that can be provided by any path pair to a given (s,t) node pair, we will in fact require the fulfilment of the geodiversity value $D_{st} = \min(D, D_{st}^{Max})$ for each (s,t) node pair.

In a related previous work [5], the problem addressed here was defined by an integer non-linear programming model (with a path based formulation), a solving algorithm based on different greedy strategies was proposed and the relative performance of the different strategies was evaluated on a set of problem instances. Here, that work is extended in the following ways. First, starting by the greedy strategies shown to be more efficient in [5], we propose improvements to such algorithms making them more effective. The path based formulation proposed in [5], although useful to reach a clear mathematical description of the problem, is hard to be used in practice since it has an exponential number of variables. Hence, next we describe the problem as an integer non-linear programming model using an arc based formulation, which although more complex, has a polynomial number of variables. Then, we redefine the problem as an integer linear programming (ILP) model with the replacement of the non-linear constraints by linear constraints which are an approximation of the first ones. Based on the obtained ILP model, we propose an ILP based heuristic for the problem. To evaluate the proposed models and resolution approaches, two test networks were used. The first belongs to the set of realistic network design test instances in [6] and the other corresponds to a hypothetical fiber-optic backbone network which was made available for research in large-scale DWDM networks. Finally, we present a set of computational results illustrating the relative performance between the non-ILP based heuristic and the ILP based heuristic.

The paper is organized as follows. Section 2 reviews the related work. In section 3, the network upgrade problem is first illustrated with an example and then described and formulated as an

3

integer non-linear programming model. In section 4, the greedy based heuristics are described, with the corresponding algorithms. Section 5 describes how the non-linear programming model is linearized and the ILP based heuristic is put forward. Computational results comparing the efficiency of the different heuristics are presented in section 6. The paper ends with section 7, where the main conclusions are presented along with some further work.

2. Related Work

The problem of the impact of geographically correlated failures and large scale disasters in optical networks is quite relevant and has been the focus of many recent publications. The option of considering a simple path protection scheme with a node-disjoint path pair for each node pair is not sufficient in the case of disasters (either natural or deliberate attacks), as a large area may be affected disrupting both the active and the backup paths. A survey on disaster survivability in optical networks is presented in [7].

In [8], the authors model the vulnerability of an optical WDM network when it is affected by events for which a certain probability is known. The possibility of multiple simultaneous attacks is considered. The authors propose a 1+1 or a 1:1 protection scheme to guarantee an appropriate service at the network, even in situations of large-scale failures or disasters. Some additional protection is provided in specific vulnerable points identified in the network. The problem of the survivability against disasters for the particular case of an optical datacenter network is tackled in [9]. For the specific case of datacenter networks, issues such as content placement, routing, and protection of paths and content are of paramount importance. The authors propose an ILP model for network design, to find out the appropriate number of data centers and where replicas of the data should be kept, in order to guarantee the survivability of the network. The network survivability in the case of geographic correlated failures is evaluated in [10]. Some vulnerable elements in the network are identified by determining the cut in the network with the largest impact for different measures that quantify the effects of failures.

Other problems related to survivability in optical networks are tackled in [11, 12]. The authors in [11] propose the use of multiple backup routes to improve the connection acceptance rate and the backup resource sharing. A comparison between their approach and conventional path protection strategies is provided. In [12], the authors consider different physical topology models and discuss their influence in the availability of a network. The models are studied in different scenarios,

4

with or without Shared Risk Link Groups (SRLG), which are groups of links (or edges) sharing a common physical resource (in this case, a single link failure turns into a multiple link failure as it affects all the links in the same SRLG). The authors in [13] tackle a problem of survivability in elastic optical networks, which allow for an adaptive allocation of spectral resources considering the traffic demands. A new Routing, Modulation and Spectrum Assignment (RMSA) algorithm is put forward, allowing for a good network performance.

Disaster-based failures have a major impact on the quality of critical network services and the fact that large-scale disasters are becoming more frequent in time and wider in scope only adds to the dimension of the problem. In this context, the end-to-end availability and the resilience of telecommunication networks to large-scale disasters is of paramount importance [14]. The requirements for critical services (eg. emergency calls, smart grid communications, financial transactions) are especially relevant, as a high end-to-end availability [15, 16, 17] must be guaranteed. The network must also be resilient so that when a disaster-based failure occurs, the impact outside the disaster area is minimal and the recovery of fully functional services is quickly achieved. A survey of strategies to protect networks against large-scale natural disasters is presented in [18].

Path (geo)diversity may be used to improve the availability of services [19] and the resilience to disasters [20] in core optical networks. In [19], the concept of path diversification at the physical layer is introduced and some metrics are discussed for the evaluation of path, node pair and graph diversity. Also the increase of flow robustness in the presence of path diversification in case of link and node failures is underlined. In [20], real optical networks were used to test, at different vulnerability scales, a new mechanism for the identification of network vulnerabilities. Furthermore, the concept of path diversification introduced in [19] was extended for path geographical diversity to quantify the geographic separation of links and nodes for resilience analysis of geographically correlated failures. Also an adequate routing algorithm is proposed to find multiple geodiverse paths to circumvent critical network regions. An improved model for the identification of vulnerable regions in case of regional correlated failures was presented in [21] and it was tested for real optical backbone networks. In this work, node centrality metrics on unweighted and weighted graphs were used to prioritize protection (weights assigned to nodes accordingly to the population of cities). The main idea of these works is that the geographical diversity of the network topology should be taken into account when making routing decisions. In other words, assuming that each pair of
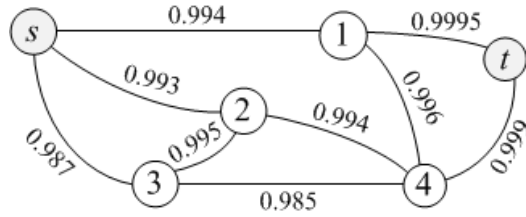
5

Figure 1: Illustrative example

network nodes is provided with multiple routing paths which are geographically separated, then the occurrence of disasters in the geographical area of one path, should not greatly affect the geographical area of the other paths.

3. Problem Motivation and Definition

125   For illustrative purposes of the tackled problem, consider the network example shown in Fig. 1 representing the geographical location of all nodes and the geographical routes of all fiber links together with their availability values. Assume in this example that node pair s–t must be provided with a pair of routing paths with a minimal availability of 0.9999 and a minimal geodiversity of 120 km.

130   In Fig. 2, the pair of routing paths in Solution 1 (highlighted in bold) has an availability of 0.99991 but a geodiversity of 74 km (highlighted in gray), which is below the required geodiversity value (the exact definition of the geodiversity value of a pair of paths is provided later in this section). On the other hand, the pair of routing paths in Solution 2 has a geodiversity of 146 km but an availability of 0.99981, which is below the required availability value.

135   In this example, to reach the required availability and geodiversity values, some links must be upgraded. By upgrading the availability of link $\{3, 4\}$ from 0.985 to 0.9998 (with a given associated cost), the network can now provide the pair of routing paths shown in the Upgraded Solution of Fig. 2 with an availability of 0.99991 and a geodiversity of 146 km. So, the aim of our problem is to determine a minimum cost set of links to be upgraded such that all node pairs of interest can 140 be provided with a pair of routing paths with required availability and geodiversity values.

Consider a directed bi-connected graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{A}$ is the set of arcs such that when $(i, j) \in \mathcal{A}$ exists, $(j, i) \in \mathcal{A}$ also exists. Each pair of arcs $(i, j), (j, i) \in \mathcal{A}$ represents the two directions of an existing fiber link (or edge) $\{i, j\} \in \mathcal{E}$ between node $i \in \mathcal{N}$ and
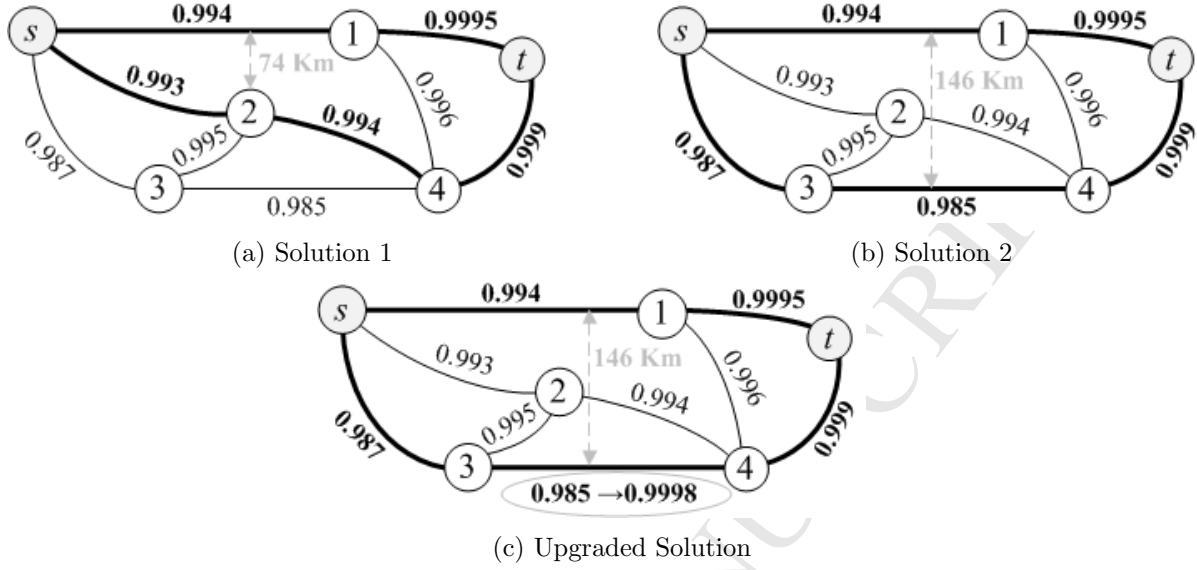
(a) Solution 1

(b) Solution 2

(c) Upgraded Solution

Figure 2: Different pairs of routing paths in the example of Fig. 1

node $j \in \mathcal{N}$ ($\{i, j\}$ is a unordered pair). Such edge is characterized by its current availability $a_{\{i,j\}}$,

145 its upgraded availability $\hat{a}_{\{i,j\}}$ and the cost $c_{\{i,j\}}$ required to upgrade its availability from $a_{\{i,j\}}$ to $\hat{a}_{\{i,j\}}$. The aim is to determine a set of edges to be upgraded at a minimal cost. To model an upgraded solution, we consider the following binary variables:

$x_{\{i,j\}}$ – this variable is equal to 1 if the edge $\{i, j\}$ is upgraded, i.e., the availability of both arcs (i, j) and (j, i) are upgraded, or 0, otherwise;

150 and, with these variables, we define the optimization function of our problem as the minimization of the upgrade cost:

$$\sum_{\{i,j\}\in\mathcal{E}} \left(c_{\{i,j\}} x_{\{i,j\}} \right) \tag{1}$$

The upgraded network must guarantee that, for a given set of node pairs K, each node pair $k \in K$ is provided with at least one pair of routing paths fulfilling two requirements: a minimum availability value defined by parameter Λ and a minimal geodiversity value defined by parameter

155 D. Each node pair $k \in K$ is defined by a source node $s_k \in \mathcal{N}$ and a target (or destination) node $t_k \in \mathcal{N}$. To model the solution of a pair of routing paths for each node pair $k \in K$, we consider the following arc variables:

y_{ij}^{ku} – this variable with index $u = 0$ is equal to 1 if arc $(i, j) \in \mathcal{A}$ is in the first path of $k \in K$ and edge $\{i, j\} \in \mathcal{E}$ is not upgraded; equal to 0, otherwise;

7

160　　　　– this variable with index $u = 1$ is equal to 1 if arc $(i, j) \in \mathcal{A}$ is in the first path of $k \in K$ and edge $\{i, j\} \in \mathcal{E}$ is upgraded; equal to 0, otherwise.

z_{ij}^{ku} – this variable with index $u = 0$ is equal to 1 if arc $(i, j) \in \mathcal{A}$ is in the second path of $k \in K$ and edge $\{i, j\} \in \mathcal{E}$ is not upgraded; equal to 0, otherwise;

　　　　– this variable with index $u = 1$ is equal to 1 if arc $(i, j) \in \mathcal{A}$ is in the second path of $k \in K$

165　　　　and edge $\{i, j\} \in \mathcal{E}$ is upgraded; equal to 0, otherwise.

The proper linking of these arc variables with the previous $x_{\{i,j\}}$ variables and definition of all variables is done with the following constraints:

$$y_{ij}^{k0} + y_{ji}^{k0} + z_{ij}^{k0} + z_{ji}^{k0} \leq 1 - x_{\{i,j\}} \qquad k \in K, \{i, j\} \in \mathcal{E} \tag{2}$$

$$y_{ij}^{k1} + y_{ji}^{k1} + z_{ij}^{k1} + z_{ji}^{k1} \leq x_{\{i,j\}} \qquad k \in K, \{i, j\} \in \mathcal{E} \tag{3}$$

$$x_{\{i,j\}} \in \{0, 1\} \qquad \{i, j\} \in \mathcal{E} \tag{4}$$

$$y_{ij}^{ku}, y_{ji}^{ku}, z_{ij}^{ku}, z_{ji}^{ku} \in \{0, 1\} \qquad k \in K, \{i, j\} \in \mathcal{E}, u = 0, 1 \tag{5}$$

When $x_{\{i,j\}} = 1$ (the edge is upgraded), constraints (2) guarantee that the arc variables y_{ij}^{ku}, y_{ji}^{ku}, z_{ij}^{ku} and z_{ji}^{ku} with index $u = 0$ are set to 0 on upgraded edges while constraints (3) guarantee that
170　at most one of the arc variables y_{ij}^{ku}, y_{ji}^{ku}, z_{ij}^{ku} and z_{ji}^{ku} with index $u = 1$ is set to 1. A similar analysis stands when $x_{\{i,j\}} = 0$ (the edge is not upgraded). Constraints (4)–(5) are the variable domain constraints of all variables.

Let $N(i)$ be the set of neighboring nodes of $i \in \mathcal{N}$ in graph G. In the next constraints, we consider for each node pair $k \in K$ an auxiliary parameter b_i^k associated to each node $i \in \mathcal{N}$ that is
175　equal to 1 when $i = s_k$, equal to –1 when $i = t_k$, and equal to 0 for all other nodes. With the arc variables, we guarantee the two desired routing paths with the following constraints:

$$\sum_{j \in N(i)} \sum_{u=0}^{1} \left(y_{ij}^{ku} - y_{ji}^{ku} \right) = b_i^k \qquad k \in K, i \in \mathcal{N} \tag{6}$$

$$\sum_{j \in N(i)} \sum_{u=0}^{1} \left(z_{ij}^{ku} - z_{ji}^{ku} \right) = b_i^k \qquad k \in K, i \in \mathcal{N} \tag{7}$$

$$\sum_{j \in N(i)} \sum_{u=0}^{1} \left(y_{ji}^{ku} + z_{ji}^{ku} \right) \leq 1 \qquad k \in K, i \in \mathcal{N} \setminus \{s_k, t_k\} \tag{8}$$

$$\sum_{u=0}^{1} \left(y_{s_k t_k}^{ku} + z_{s_k t_k}^{ku} \right) \leq 1 \qquad k \in K, \text{ if } (s_k, t_k) \in \mathcal{A} \tag{9}$$

8

Constraints (6) are path conservation constraints guaranteeing that variables y_{ij}^{ku} define the first path for node pair $k \in K$ and constraints (7) are similar constraints guaranteeing that variables z_{ij}^{ku} define the second path for node pair $k \in K$ (recall that constraints (2)–(3) guarantee that at most one of the eight variables $y_{ij}^{k0}, y_{ij}^{k1}, y_{ji}^{k0}, y_{ji}^{k1}, z_{ij}^{k0}, z_{ij}^{k1}, z_{ji}^{k0}, z_{ji}^{k1}$ is 1 for each edge $\{i,j\}$ and each node pair k). Constraints (8) guarantee that the paths are node disjoint (they cannot share an intermediate node). If arc (s_k, t_k) exists (i.e., if there is a direct edge between the nodes of pair $k \in K$), constraints (9) guarantee that the path composed by this arc alone can only be in one of the two paths. So, constraints (8)–(9) guarantee that each pair of routing paths is node and arc disjoint.

Let the availability values of each arc $(i,j) \in \mathcal{A}$ be equal to the availability values of the edge $\{i,j\}$ it belongs to, i.e., $a_{ij} = a_{ji} = a_{\{i,j\}}$ and $\hat{a}_{ij} = \hat{a}_{ji} = \hat{a}_{\{i,j\}}$. For each node pair $k \in K$, the minimal availability Λ is guaranteed by the following non-linear constraints:

$$1 - \left(1 - \prod_{\substack{(i,j)\in\mathcal{A}: \\ y_{ij}^{k0}=1}} a_{ij} \prod_{\substack{(i,j)\in\mathcal{A}: \\ y_{ij}^{k1}=1}} \hat{a}_{ij} \right) \times \left(1 - \prod_{\substack{(i,j)\in\mathcal{A}: \\ z_{ij}^{k0}=1}} a_{ij} \prod_{\substack{(i,j)\in\mathcal{A}: \\ z_{ij}^{k1}=1}} \hat{a}_{ij} \right) \geq \Lambda, k \in K \qquad (10)$$

In general terms, the geodiversity of a pair of routing paths is defined as the minimal distance between the two paths. In [2], the geodiversity value is defined as the minimal distance between any intermediate node of one path and any intermediate node of the other path. Motivated by the fact that disasters might shutdown links without affecting their end nodes, a stronger definition is used in [3]. In this case, the geodiversity value is the minimal distance between any intermediate element (node or arc) of one path and any element (again, node or arc) of the other path. Since the minimum distance between the outgoing arcs from the source node (and also between the incoming arcs into the target node) is zero (they share a common node), these arc pairs are excluded from the geodiversity definition. In [3], it is also shown how, by proper definition of distances between arc pairs (described next), the geodiversity of a pair of paths can actually be modeled based only on the distances between any arc of one path and any arc of the other path and this work follows such approach.

For each node pair $k \in K$, with source node s_k and target node t_k, the geographical distance between two arcs $a = (i_a, j_a)$ and $b = (i_b, j_b)$ is defined as (i) the minimal distance between any point in the geographical path of a and any point in the geographical path of b if the arcs do not
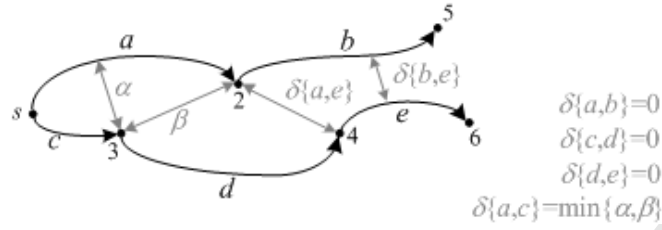
9

Figure 3: Geographical distances between arcs (adapted from [3])

share neither s_k nor t_k or (ii) the minimal distance between one of the arcs and the non-common
205 end node of the other arc if they share either s_k or t_k.

As an example, consider Fig. 3 which shows part of a network with the source node s of a particular node pair, five other nodes (2 to 6) and five arcs (a to e). The geographical distances between different arc pairs are illustrated in the figure where $\delta(a,b)$ represents the geographical distance between arcs a and b. Examples of case (i) are the distances $\delta(a,e)$ and $\delta(b,e)$. Note
210 that the zero distances $\delta(a,b)$, $\delta(c,d)$ and $\delta(d,e)$ also fall into this case, as these arc pairs share a node which is neither s nor t. As for case (ii), take arcs a and c, which share the source node s. The distance $\delta(a,c)$ is the minimum between α (the distance between arc a and node 3, which is the non-common end node of c), and β (the distance between arc c and node 2, which is the non-common end node of a). In the computational results (section 6), the geographical path of
215 each arc is assumed to be the shortest path between its end nodes over a sphere representing Earth.

Note that the geodiversity value of a pair of paths only depends on the location of each arc along each geographical path. Moreover, for each node pair $k \in K$, there is a maximal geographical distance, which we represent by D_k^{Max}, above which a pair of routing paths is infeasible (in the example of Fig. 1, the maximal geographical distance is 146 km for node pair s–t, as highlighted in
220 Fig. 2). These values can be computed in advance using [3]. Therefore, we consider the geodiversity requirement imposed to node pair $k \in K$ using $D_k = \min(D, D_k^{Max})$.

For each node pair $k \in K$, consider P_k defined as the set of all pairs of edges whose arcs cannot be simultaneously in the two paths because their geographical distance is below D_k (these sets are calculated beforehand). Each pair $p \in P_k$ is defined by edge $\{i_p, j_p\}$ whose arcs are (i_p, j_p) and
225 (j_p, i_p), and edge $\{m_p, n_p\}$ whose arcs are (m_p, n_p) and (n_p, m_p). Following [3], for each node pair

$k \in K$, the minimum geodiversity D_k is guaranteed by the following constraints:

$$\sum_{u=0}^{1} \left(y_{i_p j_p}^{ku} + y_{j_p i_p}^{ku} + z_{m_p n_p}^{ku} + z_{n_p m_p}^{ku} \right) \leq 1 \qquad k \in K, p \in P_k \tag{11}$$

$$\sum_{u=0}^{1} \left(y_{m_p n_p}^{ku} + y_{n_p m_p}^{ku} + z_{i_p j_p}^{ku} + z_{j_p i_p}^{ku} \right) \leq 1 \qquad k \in K, p \in P_k \tag{12}$$

Constraints (11)–(12) are the geodiversity constraints: for each pair of edges $p \in P_k$ if one arc of one edge is in one of the paths, none of the two arcs of the other edge can be on the other path (recall that constraints (8)–(9) already guarantee that the arcs of each edge cannot be on both paths simultaneously).

In conclusion, the network upgrade problem is defined by the following integer non-linear programming model:

Minimize   (1)

Subject to: (2)–(12)

where the non-linearity of the model is due to constraints (10).

4. Greedy based Algorithm

In this section, an heuristic approach, named Minimum Upgrade Cost with Availability and Geodiversity (MUCAG), is proposed based on a greedy strategy. For clarity, in this section each node pair of K is denoted by (s,t). The values of minimal availability Λ, the minimal geodiversity D_{st} for each node pair $(s,t) \in K$, and the sets of edge pairs $P_{(s,t)}$ $((s,t) \in K)$ are given, along with the initial availability, upgraded availability and upgrade cost of each edge. The algorithm selects one edge to be upgraded in each iteration according to a specific criterion, until the resulting network configuration provides the desired availability and geodiversity. Then, a filtering procedure is applied for decreasing (if possible) the cost and the number of upgraded edges of the previous network configuration.

The path pairs with a minimal geodiversity D_{st} for a node pair $(s,t) \in K$ constitute a feasible pair of geodiverse paths for that node pair. Therefore, a central task of the algorithm is to find a feasible pair of geodiverse paths $r \in R_{st}$ with availability $\Lambda_r \geq \Lambda$ for each node pair $(s,t) \in K$. Note that the availability Λ_r depends on the set of upgraded edges (i.e. the set of edges with an upgraded availability). This task is performed by the Guaranteed Available Pair of Geodiverse Paths (GAPGP) algorithm, an adaptation of the algorithm in [22] for calculating the most reliable

11

pair of link (or edge) disjoint paths. Note that the GAPGP algorithm calculates the pair of geodiverse paths with the highest availability value, even if it is lower than Λ.

In the following subsections, we describe separately the GAPGP algorithm, the MUCAG algorithm, the different MUCAG greedy strategies tested in practice and, finally, the strategies for filtering the set of upgraded edges at the end of MUCAG. For ease of notation, a link (or edge) will be simply noted as $e \in \mathcal{E}$ hereinafter.

4.1. GAPGP algorithm description

The pseudo-code description of the GAPGP algorithm is provided in Algorithm 1. For a given node pair $(s, t) \in K$, minimum availability value Λ, minimum geodiversity value D_{st} and the current network configuration in terms of edge availability (depending on the values assigned to the binary variables x_e), GAPGP computes a feasible pair of geodiverse paths $r \in R_{st}$ with an availability value Λ_r. This value may be $\Lambda_r \geq \Lambda$ if such path pair exists (in which case a solution providing the desired availability and geodiversity has been found for that node pair) or $\max_{r \in R_{st}} \Lambda_r < \Lambda$ (in which case we know that the desired availability is not provided to that node pair yet).

The availability of a path p is calculated by $\hat{\Lambda}(p) = \prod_{e \in p} ((1 - x_e) a_e + x_e \hat{a}_e)$. The linearization of this expression leads to $\lambda(p) = -\sum_{e \in p} ((1 - x_e) \log(a_e) + x_e \log(\hat{a}_e))$. Let $\mathcal{C}'_e = -((1 - x_e) \log(a_e) + x_e \log(\hat{a}_e)), \forall e \in \mathcal{E}$ be a positive edge "cost" related to the current availability of the edge. Using this "cost", the enumeration of the k least-cost paths corresponds to the enumeration of the k paths with the highest availability. The GAPGP algorithm starts by calculating these "costs" in lines 1–3, which will be subsequently used in the function *next-shortest-path* (line 7). This function corresponds to the iterative use of Yen's [23] k-shortest path algorithm or of the loopless version [24] of the MPS algorithm [25].

In the main while cycle (lines 6–27), the algorithm iteratively generates a new first path p with the mentioned function *next-shortest-path* by non-increasing order of availability value. A test to determine if an optimal solution has already been found is performed next. This test is based on the condition in line 11, which will be further explained later on.

For each first path p, a second path q is obtained by function *path-geo-distance* (line 16) as the edge-disjoint path with the highest availability and a geodiversity of D_{st} with p. This function runs a shortest path algorithm in an auxiliary graph given by G where the edges of p and the edges with a distance from any edge of p below D_{st} have been removed. If the second path q exists ($q \neq \emptyset$

in line 17), the availability of the feasible path pair $r' = (p, q)$ is evaluated to check if the current best solution r must be updated (lines 18–21).

The algorithm stops (line 6) when either the availability of the current best path pair r is at least Λ or the variable *opt* becomes true. In this latter case, we know that the path pair with the highest availability has been reached, even if it is lower than the desired Λ. Variable *opt* becomes true in one of two cases: i) the first case (line 25) is when function *next-shortest-path* (line 7) returns no path, which is to say that all possible paths have already been enumerated; ii) the second case (line 12) is when the availability of the current best path pair cannot be further improved (line 11).

Line 11 requires some additional explanation. Given the availability of a path p, $\hat{\Lambda}(p) = e^{-\lambda(p)}$, the availability of a path pair (p, q) is $\hat{\Lambda}(p) + (1 - \hat{\Lambda}(p))\hat{\Lambda}(q)$. Let the current best path pair be $r = (p_w, q_w)$ with availability Λ_r, where w is the order of generation of the first element of the pair (path p obtained in line 7) and q_w is the second path (obtained in line 16). Let a new first path generated by *next-shortest-path* in a subsequent iteration be p_i (with $i > w$). If $\hat{\Lambda}(p_i) + (1 - \hat{\Lambda}(p_i))\hat{\Lambda}(p_i) \leq \Lambda_r$ (line 11), then $r = (p_w, q_w)$ is optimal.

The verification of this optimality is straightforward [22]. Note that $\hat{\Lambda}(p_i) \leq \hat{\Lambda}(p_w)$, as the first paths are enumerated in non-increasing order of availability. Let q_i be the path with the highest availability which guarantees a geodiversity of D_{st} with p_i. We always have $\hat{\Lambda}(q_i) \leq \hat{\Lambda}(p_i)$. Indeed, if q_i had an availability higher than $\hat{\Lambda}(p_i)$ this would mean that q_i had already been found in an earlier iteration and was actually a first path p_j with $j \leq i$. Having proved that $\hat{\Lambda}(q_i) \leq \hat{\Lambda}(p_i)$, then the availability of this path pair (p_i, q_i) is lower than Λ_r, as $\hat{\Lambda}(p_i) + (1 - \hat{\Lambda}(p_i))\hat{\Lambda}(q_i) \leq \hat{\Lambda}(p_i) + (1 - \hat{\Lambda}(p_i)))\hat{\Lambda}(p_i) \leq \Lambda_r$, as previously mentioned. Therefore, any path pair obtained from this point onwards has an availability that does not exceed Λ_r, leading to the confirmation of the optimality of the solution $r = (p_w, q_w)$.

4.2. MUCAG algorithm description

The pseudo-code description of the MUCAG algorithm is provided in Algorithm 2. The inputs are the network topology, the edges availability, the upgraded edges availability and cost, the set of node pairs for which an availability of Λ is desired, the sets $P_{(s,t)}$ $((s,t) \in K)$, and the filtering option h (to be explained later).

At the beginning of this algorithm, set K includes the set of node pairs (s, t) for which we want to guarantee at least one path pair with a minimal availability of Λ and a minimal geodiversity of

13

Algorithm 1 GAPGP

Require: G, s, t, Λ, D_{st}, $(a_e, \hat{a}_e, x_e) : \forall e \in \mathcal{E}$, $P_{(s,t)}$
Ensure: (r, Λ_r)
 1: **for all** $e \in \mathcal{E}$ **do**
 2:     $\mathcal{C}'_e = -\log(a_e)(1 - x_e) - \log(\hat{a}_e)x_e$ $\triangleright$ New edge cost
 3: **end for**
 4: $r \leftarrow (\emptyset, \emptyset);\ \Lambda_r \leftarrow 0$
 5: $opt \leftarrow false$ $\triangleright$ r is not yet the most available path pair
 6: **while** $\Lambda > \Lambda_r \wedge \neg opt$ **do**
 7:     $p \leftarrow$ next-shortest-path$(s, t, G, \mathcal{C}')$
 8:     **if** $p \neq \emptyset$ **then**
 9:         **if** $\Lambda_r \neq 0$ **then**
10:             $g \leftarrow e^{-\lambda(p)}$ $\triangleright$ g is $\hat{\Lambda}(p)$
11:             **if** $g(2 - g) \leq \Lambda_r$ **then** $\triangleright$ $\hat{\Lambda}(p) \triangleq \hat{\Lambda}(q)$
12:                 $opt \leftarrow true$ $\triangleright$ r is the optimal solution
13:             **end if**
14:         **end if**
15:         **if** $\neg opt$ **then**
16:             $q \leftarrow$ path-geo-distance$(P_{(s,t)}, D_{st}, p, G, \mathcal{C}')$
17:             **if** $q \neq \emptyset$ **then**
18:                 $r' \leftarrow (p, q)$
19:                 **if** $\Lambda_{r'} > \Lambda_r$ **then**
20:                     $r \leftarrow r'$ $\triangleright$ Updates solution
21:                 **end if**
22:             **end if**
23:         **end if**
24:     **else**
25:         $opt \leftarrow true$ $\triangleright$ No more improvement possible
26:     **end if**
27: **end while**

D_{st}. This algorithm starts by setting all the variables x_e to 0 (lines 1–3), as the network has no upgraded edges yet. Set K is saved in $\bar{K}$ (line 4) so the obtained solution may be processed later.

The algorithm proceeds by running a while cycle (lines 5–23). At the beginning of this cycle, set

315 K contains the node pairs (s, t) for which a pair of geodiverse paths with the required availability Λ has not been found yet. The while cycle runs until this set K is empty (line 5).

In the first step of each cycle (line 6), the auxiliary sets K' and $\mathcal{R}$ are first initialized empty. Set K' will be used to keep track of the node pairs (s, t) for which a pair of geodiverse paths with the required availability Λ has not been found even after a run of the GAPGP algorithm (lines 8–

320 10). Set $\mathcal{R}$ includes all the best pairs of geodiverse paths, which still do not fulfil the availability

requirement Λ.

With the current network configuration of upgraded edges, a verification is performed to see if some of the node pairs $(s, t) \in K$ now have a path pair satisfying both the availability and the geodiversity requirements. For each $(s, t) \in K$ (lines 7-13), a run of the GAPGP algorithm (line 8) computes a pair of geodiverse paths r with an availability Λ_r. If the availability of that path pair is lower than the required value ($\Lambda_r < \Lambda$ – line 9), then the node pair (s, t) is added to set K' (line 10) and the path pair r (obtained with the GAPGP algorithm) is added to $\mathcal{R}$ (line 11). As we shall see, the edges in these path pairs appear to be appropriate for upgrading, in order to try and achieve the desired availability for the path pairs of all the node pairs in the forthcoming iterations of this cycle. Note that if all the edges in set $\mathcal{R}$ have already been upgraded, the algorithm ends without achieving the desired end-to-end availability for the node pairs currently in set K'. This situation is tested in line 20.

Afterwards, K is set with K' (line 14), i.e., K has the new (smaller) set of node pairs (s, t) for which there is still no pair satisfying both the availability and the geodiversity requirements. If set K is not empty (lines 15-19), the algorithm proceeds by selecting one edge e (among the non-upgraded ones belonging to path pairs in $\mathcal{R}$) with function *select-edge* (line 16).

The function *select-edge* also returns set K'', which includes all the node pairs (s, t) for which the availability requirement is now satisfied, while using the paths in $\mathcal{R}$, when the selected edge is upgraded. Obviously, $K'' \subseteq K$ and K'' may be an empty set. The edge e selected for upgrading is signalled by the value of variable x_e (line 17) and finally, K'' is removed from set K (line 18).

MUCAG is a greedy heuristic, as it selects the best edge (under the circumstances described for each variant – see the next subsection) in each iteration. As the heuristic is solved and more edges are being upgraded, we anticipate that some of the edges upgraded earlier may not be actually necessary for the purpose of guaranteeing an availability not lower than Λ for the node pairs which currently have the availability requirement satisfied. For this reason at the end of the algorithm the function *filter-solution* is called to remove redundant edges if parameter h is not zero – further details are in subsection 4.4. If $h = 0$, none of the upgraded edges has its availability downgraded to the original value. This corresponds to the algorithm proposed in [5] and may be considered the basic heuristic.

Algorithm 2 MUCAG

Require: G, K, Λ, $(a_e, \hat{a}_e, c_e) : \forall e \in E$, $D_{st} : \forall (s,t) \in K$, h (filtering option),
Ensure: $x_e : \forall e \in E$

 1: **for all** $e \in E$ **do**
 2:     $x_e \leftarrow 0$
 3: **end for**
 4: $\bar{K} \leftarrow K$
 5: **while** $K \neq \emptyset$ **do**
 6:     $K' \leftarrow \emptyset, \mathcal{R} \leftarrow \emptyset$
 7:     **for all** $(s,t) \in K$ **do**
 8:         $(r, \Lambda_r) \longleftarrow$ GAPGP$(G, s, t, \Lambda, D_{st}, (a_e, \hat{a}_e, x_e) : e \in E, P_{(s,t)})$
 9:         **if** $\Lambda_r < \Lambda$ **then**
10:             $K' \leftarrow K' \cup \{(s,t)\}$
11:             $\mathcal{R} \leftarrow \mathcal{R} \cup \{r\}$
12:         **end if**
13:     **end for**
14:     $K \leftarrow K'$
15:     **if** $|K| > 0$ **then**
16:         $(e, K'') \leftarrow$ select-edge$(\mathcal{R}, K, x_e : e \in E)$
17:         $x_e \leftarrow 1$
18:         $K \leftarrow K \setminus K''$
19:     **end if**
20:     **if** $\nexists e : x_e = 0$ **then**
21:         Exit the While cycle
22:     **end if**
23: **end while**
24: **if** $h \neq 0$ **then**
25:     filter-solution$(h, G, \bar{K}, \Lambda, (a_e, \hat{a}_e, c_e, x_e) : \forall e \in E, P_{(s,t)} : \forall (s,t) \in \bar{K})$
26: **end if**

---

350 *4.3. Variants for the greedy algorithm*

As explained in the description of the MUCAG algorithm, the edge to be upgraded in each iteration is selected according to a certain criterion (line 16 of Algorithm 2). Let $\mathcal{E}(\mathcal{R})$ be the set of edges, which have not been upgraded yet, belonging to at least one of the path pairs in $\mathcal{R}$.

Let $C_E(e), e \in \mathcal{E}(\mathcal{R})$ be the number of times that edge $e \in \mathcal{E}(\mathcal{R})$ is in the path pairs of $\mathcal{R}$. 355 Given these values, we define the subset $\mathcal{E}_M(\mathcal{R})$ which contains the edges in $\mathcal{E}(\mathcal{R})$ with maximal value of $C_E(e)$, i.e. $\mathcal{E}_M(\mathcal{R})$ contains the most frequent non-upgraded edges in path pairs of $\mathcal{R}$.

Let $A_E(e), e \in \mathcal{E}(\mathcal{R})$ be the number of path pairs of $\mathcal{R}$ for which the availability becomes at least Λ if edge $e \in \mathcal{E}(\mathcal{R})$ is upgraded. Given these values, we define the subset $\mathcal{E}_O(\mathcal{R})$ which contains the edges in $\mathcal{E}(\mathcal{R})$ with maximal value of $A_E(e)$, assuming $A_E(e) > 0$, i.e. $\mathcal{E}_O(\mathcal{R})$ contains

16

the non-upgraded edges in path pairs of $\mathcal{R}$ which lead to more path pairs achieving at least the desired availability. If $\max_{e \in \mathcal{E}(\mathcal{R})} A(e) = 0$ then $\mathcal{E}_O(\mathcal{R})$ is an empty set (this means that upgrading any single edge from $\mathcal{E}(\mathcal{R})$ will not change the number of node pairs with the required availability).

The considered greedy strategies for selecting an edge for upgrade are as follows:

- **MinCost-MaxCount:** Select the edge $e \in \mathcal{E}_M(\mathcal{R})$ with minimum upgrade cost edge c_e, i.e. the edge with minimal upgrade cost is selected among the most frequent ones in path pairs of $\mathcal{R}$.

- **MinCost-MaxOn:** Select the edge $e \in \mathcal{E}_O(\mathcal{R})$ with minimum upgrade cost edge c_e, i.e. the edge with minimal upgrade cost is selected among the ones leading to more path pairs achieving at least the desired availability. If $\mathcal{E}_O(\mathcal{R})$ is an empty set in an iteration, then it is replaced by $\mathcal{E}_M(\mathcal{R})$ (leading to a MinCost-MaxCount strategy) in that iteration.

- **MaxOn-MaxCount:** Select the edge $e \in \mathcal{E}_M(\mathcal{R})$ such that, when upgraded, will lead to the greatest number of path pairs achieving at least the desired availability, i.e. the edge leading to more node pairs with the required availability is selected among the most frequent ones in path pairs of $\mathcal{R}$. If in a certain iteration none of the edges in $\mathcal{E}_M(\mathcal{R})$ leads to node pairs attaining the required availability, then any edge in $\mathcal{E}_M(\mathcal{R})$ is selected – resulting in the selection of the most frequent edge among the ones in path pairs of $\mathcal{R}$ in this iteration.

- **MaxCount-MaxOn:** Select the most frequent edge $e \in \mathcal{E}_O(\mathcal{R})$, i.e. the edge appearing in more path pairs is selected among the ones leading to the greatest number of path pairs achieving at least the desired availability. If $\mathcal{E}_O(\mathcal{R})$ is an empty set in an iteration, then it is replaced by $\mathcal{E}_M(\mathcal{R})$ in that iteration.

4.4. Strategies for filtering the set of upgraded edges

As mentioned in the description of the MUCAG algorithm, it is possible that some of the edges upgraded in earlier iterations may not be actually necessary for the purpose of guaranteeing an availability not lower than Λ for the node pairs which have the availability requirement satisfied.

If parameter h is not 0 function *filter-solution* is called and an attempt is made to downgrade some of the upgraded edges (without compromising the desired availability). This function has two variants, depending on the value of parameter h ($h = 1, 11$). If $h = 1$ an exhaustive search is made among the upgraded edges to determine the subset of edges (by increasing size of such sets),

17

that if downgraded to the their initial availability value, would result in the largest cost decrease

390 of the obtained solution, while still ensuring the required end-to-end availability for all pairs in $\bar{K}$. If $h = 11$ a greedy approach is used, and the more expensive edge, that can be downgraded to its initial availability value, is successively selected, until no more edges can be downgraded while ensuring Λ availability for all node pairs in $\bar{K}$.

We also evaluated using function *filter-solution* after each edge was selected in line 16 (discour-

395 aging the re-selection of the edges identified for downgrade) but this approach resulted in a very high increase of CPU time, and not necessarily in better solutions. Hence, this possibility was discarded.

5. ILP based Heuristics

To be able to use standard Integer Linear Programming (ILP) solving techniques, constraints (10)

400 of the model presented in section 3 must be redefined. The aim is to reach a set of linear constraints that approximate as much as possible constraints (10). A possible approach is the one proposed in [26]. However, that approach does not allow to control how close the linear approximation is to the exact non-linear problem. Instead, we propose a different set of linear constraints that can be made as close as required to constraints (10), as described next.

405 For each node pair $k \in K$, consider a lower bound of the first routing path availability given by the real variable Λ_1^k and a lower bound of the second path availability given by the real variable Λ_2^k. Consider also arc "cost" values given by $a_{ij}^0 = a_{ji}^0 = -\log\left(a_{\{i,j\}}\right)$, when edge $\{i,j\}$ is not upgraded, and $a_{ij}^1 = a_{ji}^1 = -\log\left(\hat{a}_{\{i,j\}}\right)$, when edge $\{i,j\}$ is upgraded. Constraints (10) can be redefined by the following still non-linear constraints:

$$\sum_{(i,j)\in\mathcal{A}} \sum_{u=0}^{1} \left(a_{ij}^u y_{ij}^{ku}\right) \leq -\log\left(\Lambda_1^k\right) \qquad k \in K \tag{13}$$

$$\sum_{(i,j)\in\mathcal{A}} \sum_{u=0}^{1} \left(a_{ij}^u z_{ij}^{ku}\right) \leq -\log\left(\Lambda_2^k\right) \qquad k \in K \tag{14}$$

$$1 - \left(1 - \Lambda_1^k\right) \times \left(1 - \Lambda_2^k\right) = \Lambda \qquad k \in K \tag{15}$$

$$\Lambda_1^k > 0, \Lambda_2^k > 0 \qquad k \in K \tag{16}$$

410 In constraints (13), guaranteeing that the total "cost" of the arcs included in the first path (defined by variables y_{ij}^{ku}) is lower or equal than $-\log\left(\Lambda_1^k\right)$, we are guaranteeing that the availability

18

of the first path is higher or equal than Λ_1^k. Constraints (14) do the same for the second routing path defined by variables z_{ij}^{ku}. Constraints (15) guarantee that the availability of the pair of routing paths is at least Λ when the availability of the first path is at least Λ_1^k and the availability of the second path is at least Λ_2^k. Finally constraints (16) are the variable domain constraints.

To reach a linear set of constraints, we discretize the pairs of values Λ_1^k and Λ_2^k in W different pairs and use the discrete values as parameters related to new binary variables. Consider the different pairs of values represented by $\left(\Lambda_y^w, \Lambda_z^w\right)$, with $w = 1, 2, \cdots, W$, and the "cost" parameter pairs $\left(a_y^w = -\log \Lambda_y^w, a_z^w = -\log \Lambda_z^w\right)$, with $w = 1, 2, \cdots, W$. Consider also the following additional binary variables:

v_w^k – this variable is equal to 1 if the first path of node pair $k \in K$ has an availability not lower than Λ_y^w and the second path of node pair $k \in K$ has an availability not lower than Λ_z^w; or 0, otherwise.

With the discretized values and new variables, the previous constraints (13)–(16) are reformulated as:

$$\sum_{(ij)\in\mathcal{A}} \sum_{u=0}^{1} \left(a_{ij}^u y_{ij}^{ku}\right) \leq \sum_{w=1}^{W} \left(a_y^w v_w^k\right) \qquad k \in K \tag{17}$$

$$\sum_{(ij)\in\mathcal{A}} \sum_{u=0}^{1} \left(a_{ij}^u z_{ij}^{ku}\right) \leq \sum_{w=1}^{W} \left(a_z^w v_w^k\right) \qquad k \in K \tag{18}$$

$$\sum_{w=1}^{W} v_w^k = 1 \qquad k \in K \tag{19}$$

$$v_w^k \in \{0,1\} \qquad k \in K, w = 1, 2, \cdots, W \tag{20}$$

Since all pairs of values $\left(\Lambda_y^w, \Lambda_z^w\right)$ are computed guaranteeing the equality defined in constraints (15), i.e. $\Lambda_z^w = \frac{\Lambda - \Lambda_y^w}{1 - \Lambda_y^w}$, these constraints are no longer needed. Instead, constraints (19) force one of the pairs of values to be selected for each node pair $k \in K$. Constraints (17)–(18) guarantee that the selected pair of availability values is fulfilled by the pair of routing paths of the same node pair $k \in K$. Finally, constraints (20) are the variable domain constraints of the previously introduced variables.

By replacing constraints (10) with constraints (17)–(20) in the model proposed in section 3, we obtain an ILP model that can be dealt with by standard ILP solving techniques. Note though that the optimal value of the ILP model is an upper bound of the optimal value of the original

19

Table 1: Availability values for the first and the second paths.

w	$\Lambda = 0.9999$		$\Lambda = 0.99998$		$\Lambda = 0.99999$	
	Λ_y^w	Λ_z^w	Λ_y^w	Λ_z^w	Λ_y^w	Λ_z^w
1	0.990000000000000	0.990000000000000	0.995527864044998	0.995527864044998	0.996837722339839	0.996837722339839
2	0.990732392875567	0.989209728179311	0.995990354537884	0.995012027824160	0.997272074065046	0.996334211324501
3	0.991410874309088	0.988357371448668	0.996404908557973	0.994436859166857	0.997646671097441	0.995750700214882
4	0.992039412566081	0.987438112974690	0.996776494610024	0.993795574202477	0.997969733980343	0.995074537078819
5	0.992621683807115	0.986446772219328	0.997109566219942	0.993080623352105	0.998248352338584	0.994291089343922
6	0.993161093588738	0.985377779138005	0.997408115311091	0.992283607335773	0.998488640460545	0.993383440710896
7	0.993660796781712	0.984225146827364	0.997675720048216	0.991395184566873	0.998695871476404	0.992332044105300
8	0.994123716023055	0.982982442578961	0.997915587722395	0.990404969201679	0.998874593077133	0.991114325141631
9	0.994552558809797	0.981642757304138	0.998130593190999	0.989301419089882	0.999028727313720	0.989704230190752
10	0.994949833334445	0.980198673306755	0.998323313333380	0.988071712861931	0.999161656666694	0.988071712861931

435 non-linear problem. The reason for this is that any solution compliant with constraints (17)–(20) is also compliant with constraints (10), meaning that all feasible solutions of the ILP model are also valid solutions of the original problem. However, there might exist solutions compliant with constraints (10) that are not compliant with constraints (17)–(20), meaning that the optimal solution of the ILP model might not be the optimal solution of the original problem.

440 The number of availability value pairs W and the values selected for each pair influence how close the ILP model is to the original non-linear problem. On the one hand, W should be as large as possible, so that the linear approximation given by constraints (17)–(20) becomes closer to constraints (10), but on the other hand it cannot be so large that the number of resulting variables makes the resulting model too complex to be solved.

445 In the computational results, after some preliminary tests, we have considered $W = 10$. The ten different pairs of values were considered as follows. First, without any lack of generality, we have assumed that, for any node pair $k \in K$, the first routing path has always an availability not lower than the availability of the second routing path, i.e. $\Lambda_y^k \geq \Lambda_z^k$. In this case, the lowest value of Λ_y^k is when $\Lambda_y^k = \Lambda_z^k$ which, doing the calculations, results in $\Lambda_y^k = 1 - \sqrt{1 - \Lambda}$. So, the first discretized
450 pair of values is $\left(\Lambda_y^1 = 1 - \sqrt{1 - \Lambda}, \Lambda_z^2 = 1 - \sqrt{1 - \Lambda}\right)$. For the other pairs $w = 2, 3, \cdots, 10$, we have considered the availability values that are shown in Table 1 for $\Lambda = 0.9999$, $\Lambda = 0.99998$ and $\Lambda = 0.99999$. For illustration purposes, these values are also shown in the plots of Fig. 4. The lines (solid for the first path and dotted for the second path) on each plot highlight the non-linear relation between Λ_y^k and Λ_z^k to reach each value of Λ and the tick marks are the different pairs of
455 values $\left(\Lambda_y^w, \Lambda_z^w\right)$, with $w = 1, ..., 10$, which are a discretization of the lines.

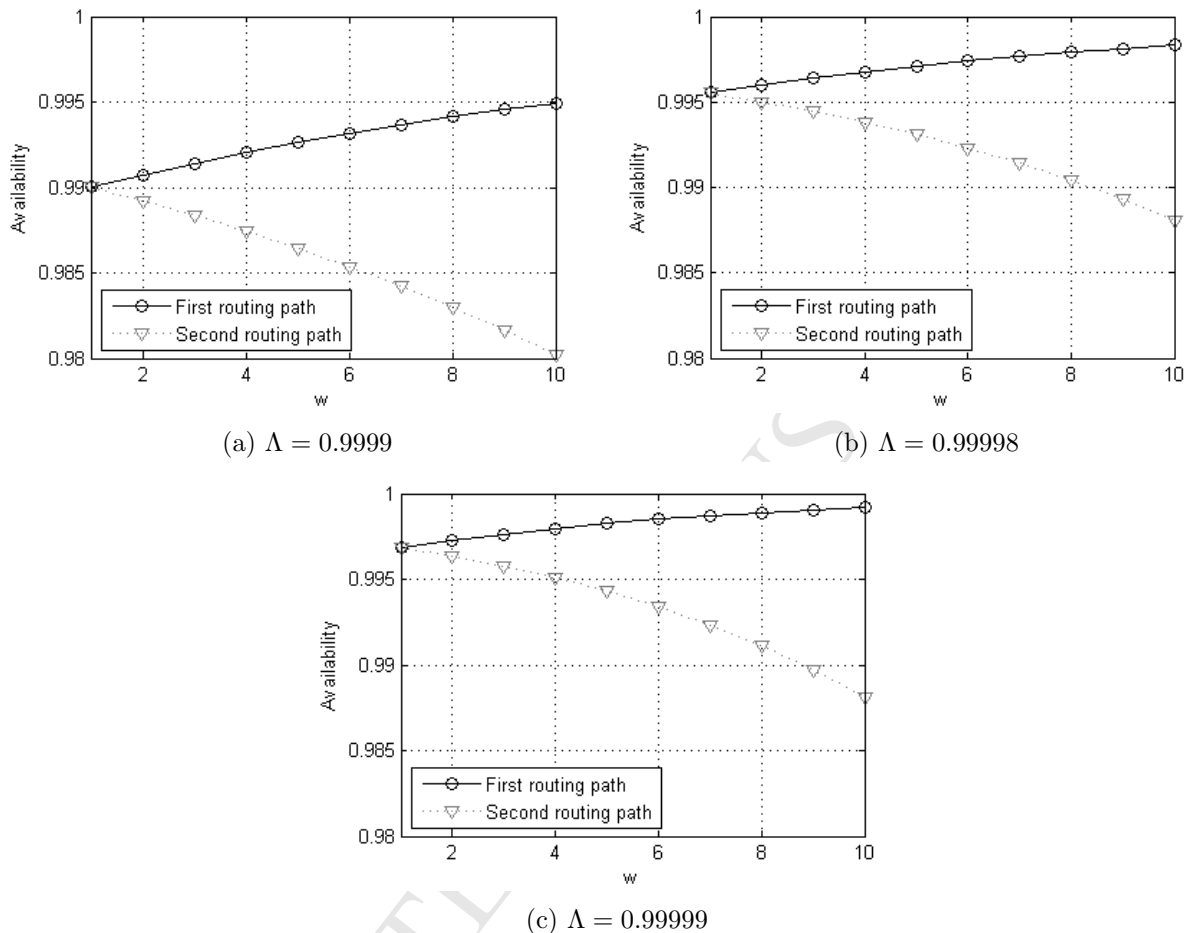Although this ILP model can be used to solve small to medium sized instances, it does not scale

(a) $\Lambda = 0.9999$

(b) $\Lambda = 0.99998$

(c) $\Lambda = 0.99999$

Figure 4: Availability pairs of values for the different required values of Λ

for bigger instances and cannot be solved for the problem instances considered in the computational results. So, we have adopted an ILP based heuristic described as follows.

The adopted heuristic follows a greedy strategy. We start by considering a small number of node pairs belonging to K and compute the minimum cost set of edges to be upgraded. Then, we fix these upgraded edges and repeat the process until all node pairs of K are provided with at least one pair of routing paths with the required availability and geodiversity values. Some preliminary tests have shown that at the beginning, the ILP model can be solved when the number of nodes pairs is small (we adopted a value of 8). Then, after fixing some edges to be upgraded, the model can be solved with a higher number of node pairs (we adopted a value of 20).

Moreover, to select at each step the node pairs to be considered, we resort to the Guaranteed Available Pair of Geodiverse Paths (GAPGP), as described in the previous section, to select the

21

node pairs whose optimal availability have the worst values for the current set of upgraded edges. This approach is of key importance in the efficiency of the proposed ILP based heuristic. When some edges are upgraded, GAPGP not only reduces significantly the number of node pairs that must be considered in the next step but, in many cases, it eliminates node pairs that, if considered in the next steps, would not be accepted by constraints (17)–(20) without the need of new upgraded edges.

The implemented ILP based heuristic is as follows:

1. Run GAPGP with the initial network configuration (i.e., with no upgraded edges) and select a set K' with up to 8 node pairs with worst availability value.

2. If $|K'| > 0$, solve the ILP model considering the node pairs in K' and update the current network configuration with the upgraded edges of the ILP solution; otherwise, exit the algorithm and the result is the current network configuration.

3. Run GAPGP with the current network configuration and select a set K' with up to 20 node pairs with worst (optimal) availability value. Return to Step 2.

When running the ILP based heuristic, we have imposed a CPU time limit of 2 hours to solve each ILP model. The reason for this decision is that usually the ILP solver (we have used CPLEX 12.6.1 [27]) does not take too much time in obtaining a good solution but it takes an exaggerated amount of time to obtain a provable optimal solution.

Finally, as in the previous section, we also consider the same parameter h at the end of the ILP based heuristic. If $h = 0$, the solution computed at the end of the heuristic is the final solution and we consider this as the basic heuristic. If $h = 1, 11$, we filter the solution given by the basic ILP based heuristic using the filtering strategies described in the previous section.

6. Computational Results

Two network topologies representative of typical core optical networks were considered in this experimental study: the Germany50 topology (the geographical information of nodes location is available at [6]) and the CORONET CONUS topology (geographical location of nodes available at http://www.monarchna.com/topology.html), hereinafter called Coronet. No public information on the geographical path of each fiber link is available. For this reason, we have considered that edges (representing fiber links) follow the shortest path over a sphere representing the Earth, as explained

22

in section 3. The information on edge lengths and geographical distance between edge-edge pairs and node-edge pairs is available at http://www.av.it.pt/asou/geodiverse.htm.

For each topology, we calculated first the maximum geodiversity value D_{st}^{Max} for all node pairs 500 (s, t) obtained by the Maximum Distance D of Geodiverse Paths (MDDGP) optimization problem in [3]. The topology characteristics of both networks are presented in Table 2.

The initial availability a_e of each edge $e \in \mathcal{E}$ is based on its length [28]:

$$a_e = 1 - \frac{MTTR}{MTBF_e} \tag{21}$$

with

$$MTBF_e \quad [\text{hrs}] = \frac{CC \times 365 \times 24}{\ell_e} \tag{22}$$

where $MTBF$ and $MTTR$ are the mean time between failures and mean time to repair in hours, 505 respectively. The parameter CC is the cable cut metric and ℓ_e is the length of edge e, both in km. Let $MTTR = 24$ h and $CC = 450$ km [16]. The upgraded availability $\hat{a}_e$ for each edge $e \in \mathcal{E}$ is obtained by considering a parallel edge of the same length, so that $\hat{a}_e = a_e (2 - a_e)$.

We consider that the cost c_e required to the availability upgrade of $e \in \mathcal{E}$ from a_e to $\hat{a}_e$ is given by its length: $c_e = \ell_e, \forall e \in \mathcal{E}$. Note that, besides the length of the upgraded edges, the number 510 of edges to be upgraded might also impact the actual cost of upgrading a network configuration. Nevertheless, there is no available information on the relative cost weight of this factor. Therefore, the analysis of the quality of results of the algorithms is performed in terms of two different parameters: (i) the number of upgraded edges $\sum_{e \in \mathcal{E}} x_e$ – referred to as $\#\hat{e}$ in the tables of results; (ii) the total upgrade cost $\sum_{e \in \mathcal{E}} c_e x_e$, – referred to as Cost in the tables of results.

515 In the function *next-shortest-path* (line 7 of the GAPGP algorithm) the loopless version [24] of the MPS algorithm [25] was used, although the code used a single core only.

A server with Intel Xeon X5660 (Six Core, 2.80 GHz, 48GB RAM) was used in the computational experiments for the non ILP based heuristics while a server with Intel Xeon E5-2650v2 (dual

Table 2: Network characteristics ($|\mathcal{N}|$, $|\mathcal{E}|$, d – average node degree, $L_M = \max_{e \in \mathcal{E}} \ell_e$, $L_a = \text{avg}_{e \in \mathcal{E}} \ell_e$, $\max_{st} D_{st}^{Max}$) – All lengths are in km.

Network	$|\mathcal{N}|$	$|\mathcal{E}|$	d	L_M	L_a	$\max_{st} D_{st}^{Max}$
Germany50	50	88	3.52	252	100.67	166
Coronet	75	99	2.64	1017	329.72	707

processor, Sixteen Core, 2.60 GHz, 64GB RAM) was used in the computational experiments for
the ILP based heuristics. While running the ILP based heuristics, CPLEX 12.6.1 was used with
its default settings to solve each ILP model limited to the use of at most eight cores.

6.1. Results for the Germany50 network

Due to its geographical coverage and the considered edge availability values, the Germany50
network already provides a four nines (0.9999) availability between all node pairs, even when the
maximal geodiversity value D_{st}^{Max} is required for all node pairs $(s, t) \in K$. Therefore, we have
assumed two minimal availability Λ values well above 0.9999 (i.e. $\Lambda = 0.99998, 0.99999$).

For the lowest value ($\Lambda = 0.99998$), we have considered one scenario where this value is required
for all node pairs of the network. For the highest value ($\Lambda = 0.99999$), we have considered two
scenarios: one where this value is required for all node pairs and another where we assumed that
critical services are concentrated on three nodes – Berlin (4), Frankfurt (17) and Munich (35) –
and the minimal availability value is required only to node pairs involving at least one of these
nodes. For all three scenarios, we have considered geodiversity values D of 40 km, 80 km, 120 km
and 160 km. Table 3 presents the number of node pairs that do not satisfy the desired availability
in the considered scenarios before the upgrade of any edge.

Tables 4–6 present the number of upgraded edges, the upgrade cost (length based) and the CPU
time for the three considered scenarios. Solutions providing the minimum number of upgraded
edges and/or the minimum upgrade cost are highlighted in bold. In Table 4 (for $\Lambda = 0.99999$) and
Table 5 (for $\Lambda = 0.99998$), the Λ value was required between all node pairs while in Table 6 the
$\Lambda = 0.99999$ was required between some of the node pairs, as explained.

One immediate observation is that a higher geodiversity value D and a higher number of
unsatisfied node pairs (see Table 3) tend to lead to solutions with both higher number of upgraded
edges and cost, and higher running times. There are a few exceptions to this expected observation,

Table 3: Number of node pairs initially with availability less than Λ for Germany50

Λ	D [km]			
	40	80	120	160
0.99999 (between all node pairs)	446	665	700	704
0.99998 (between all node pairs)	85	227	257	261
0.99999 (at least one end node is Berlin, Frankfurt or Munich)	53	86	91	92

24

Table 4: No. of upgraded edges ($\#\hat{e}$), cost and CPU time (in seconds) for achieving $\Lambda = 0.99999$ between all node pairs in Germany50

Algorithm		$D = 40$			$D = 80$			$D = 120$			$D = 160$		
		$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU
ILP based heuristic	$h = 0$	23	2181	3233	27	2509	14401	27	2617	7659	27	2671	14400
	$h = 1$	18	1584	3457	24	2338	14673	25	**2501**	7788	25	**2501**	14484
	$h = 11$	22	2017	3243	26	2433	14521	26	2549	7744	26	2549	14528
MinCost--MaxCount	$h = 0$	18	1507	13.5	25	2289	140	32	2843	447	37	3285	648
	$h = 1$	18	1507	19.4	23	**2192**	227	30	2735	708	30	2756	21875
	$h = 11$	18	1507	19.4	23	**2192**	214	31	2740	610	31	2805	885
MinCost--MaxOn	$h = 0$	15	1842	12.1	**20**	2664	154	25	3355	325	26	3405	353
	$h = 1$	15	1842	17.3	**20**	2664	182	24	3182	441	25	3319	430
	$h = 11$	15	1842	17.4	**20**	2664	182	24	3182	422	25	3319	438
MaxOn--MaxCount	$h = 0$	16	1469	12.7	22	2260	138	28	2839	428	29	2914	573
	$h = 1$	**13**	**1342**	24.2	22	2260	213	26	2723	637	27	2749	1000
	$h = 11$	14	1369	18.2	22	2260	208	27	2759	553	27	2777	838
MaxCount--MaxOn	$h = 0$	14	1688	11.1	21	2563	155	27	3389	359	25	3190	358
	$h = 1$	14	1688	15.3	**20**	2415	183	**23**	2763	608	**24**	3101	472
	$h = 11$	14	1688	15.0	**20**	2415	182	24	2878	472	**24**	3101	471

Table 5: No. of upgraded edges ($\#\hat{e}$), cost and CPU time (in seconds) for achieving $\Lambda = 0.99998$ between all node pairs in Germany50

Algorithm		$D = 40$			$D = 80$			$D = 120$			$D = 160$		
		$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU
ILP based heuristic	$h = 0$	**5**	**530**	2909	13	1222	3606	16	1331	2893	15	1390	3430
	$h = 1$	**5**	**530**	2911	11	1111	3425	15	**1268**	2880	15	**1358**	8299
	$h = 11$	**5**	**530**	2911	12	1137	3416	15	**1268**	2874	15	**1358**	8297
MinCost--MaxCount	$h = 0$	7	725	2.2	15	1276	56.7	18	1502	100.0	19	1667	121.0
	$h = 1$	6	605	4.8	15	1276	91.7	18	1502	163.9	18	1608	206.4
	$h = 11$	6	605	4.6	15	1276	88.5	18	1502	161.7	18	1608	178.6
MinCost--MaxOn	$h = 0$	6	690	2.1	10	1434	35.7	13	1638	73.2	16	2112	113.3
	$h = 1$	6	690	4.3	**9**	1358	57.6	**12**	1562	110.7	15	2023	162.1
	$h = 11$	6	690	4.3	**9**	1358	56.5	**12**	1562	106.8	15	2023	161.4
MaxOn--MaxCount	$h = 0$	6	701	2.1	12	1271	55.0	13	1547	92.0	14	1762	102.3
	$h = 1$	6	701	4.6	11	**1105**	80.3	**12**	1490	123.3	**13**	1703	153.5
	$h = 11$	6	701	4.4	11	**1105**	84.9	**12**	1490	116.1	**13**	1703	144.9
MaxCount--MaxOn	$h = 0$	6	701	2.1	11	1421	37.3	13	1659	79.4	14	1698	88.8
	$h = 1$	6	701	4.5	10	1328	59.9	13	1659	108.5	**13**	1595	124.5
	$h = 11$	6	701	4.4	10	1328	58.9	13	1659	107.1	**13**	1595	118.7

Table 6: No. of upgraded edges (#ê), cost and CPU time (in seconds) for achieving $\Lambda = 0.99999$ between some node pairs (all the pairs with source or destination Berlin (4), Frankfurt (17) or Munich (35)) in Germany50

Algorithm		$D = 40$			$D = 80$			$D = 120$			$D = 160$		
		#ê	Cost	CPU	#ê	Cost	CPU	#ê	Cost	CPU	#ê	Cost	CPU
ILP based heuristic	$h = 0$	8	864	721	17	1813	15487	15	1497	8017	15	**1497**	8586
	$h = 1$	8	864	721	13	1372	15494	15	1497	8020	15	**1497**	8590
	$h = 11$	8	864	721	16	1561	15489	15	1497	8020	15	**1497**	8590
MinCost--MaxCount	$h = 0$	10	911	0.8	15	**1259**	7.8	15	1433	9.6	22	1889	10.2
	$h = 1$	9	**857**	1.3	15	**1259**	11.6	15	1433	14.4	21	1759	18.7
	$h = 11$	9	**857**	1.3	15	**1259**	11.6	15	1433	14.2	21	1759	18.3
MinCost--MaxOn	$h = 0$	9	1229	0.7	**11**	1315	5.9	15	1853	11.6	15	1941	17.1
	$h = 1$	**7**	990	1.2	**11**	1315	8.6	**13**	1624	16.4	14	1807	24.5
	$h = 11$	**7**	990	1.2	**11**	1315	8.6	14	1696	16.6	14	1807	24.8
MaxOn--MaxCount	$h = 0$	9	911	0.8	14	1441	7.1	15	1475	10.2	17	1919	10.1
	$h = 1$	9	911	1.5	14	1441	10.4	14	**1418**	13.7	15	1614	18.9
	$h = 11$	9	911	1.5	14	1441	10.5	14	**1418**	13.7	15	1614	13.8
MaxCount--MaxOn	$h = 0$	8	1143	0.8	**11**	1335	5.9	15	1752	11.4	15	1912	13.8
	$h = 1$	**7**	990	1.2	**11**	1335	8.4	14	1667	17.4	**13**	1607	18.9
	$h = 11$	**7**	990	1.2	**11**	1335	8.5	14	1667	17.1	**13**	1607	17.9

due to the heuristic nature of the resolution approaches.

The number of redundant edges removed by function *filter-solution* is between 0 and 2 in the
545 results of Tables 5-6 and go up to 7 in the results of Table 4. Note that Table 4 has the results for
the scenario with a much larger number of node pairs with availability less than Λ (see Table 3).
The greedy strategy of all algorithms makes them less effective when more node pairs require
network availability upgrade and, so, the use of function *filter-solution* provides higher gains both
in terms of cost reduction and number of removed edges in this case.

550 In terms of the filtering strategies, the strategy of $h = 1$ is never outperformed by any of the
other two. Note that $h = 1$ is a refinement of the selection of the edges among those upgraded
with $h = 0$, as this strategy tries to perform an exhaustive search to identify the edges that
actually do not need to be upgraded. The idea behind the strategy of $h = 11$ is similar, but rather
than performing an exhaustive identification, only a greedy identification is made. Therefore, the
555 strategy of $h = 11$ allows (when possible) an improvement of the results of the basic strategy
($h = 0$), as expected, but the results are not so good as with $h = 1$. Again the drawback of the
best strategy is the running time: $h = 1$ tends to take longer, then $h = 11$ and finally $h = 0$ is the
fastest. In fact, the strategy $h = 11$ leads to a cost only slightly higher than the strategy $h = 1$. As
the CPU time of the former is less or significantly less than the latter, $h = 11$ can be considered a
560 good compromise approach.

An extreme case of runtime penalty introduced by $h = 1$ occurs for MinCost-MaxCount and

$D = 160$ km (see Table 4), where the basic heuristic ($h = 0$) selected 37 edges for upgrade, and there is a total of 7 and 6 redundant edges removed by *filter-solution* when $h = 1$ and $h = 11$, respectively, but with a very significant increase in CPU time in the case of $h = 1$ resulting only on a very slight decrease in the solution cost (from 2805 to 2756). The significant increase in CPU time is related to the large number of times GAPGP is called by *filter-solution* when $h = 1$. In the particular case of Germany50, MinCost-MaxCount, $D = 160$ km and while the exhaustive search was done, the number of evaluated candidate sets was 2445; for each set the function had to call GAPGP for all node pairs associated with that candidate set. Note that in the case of *filter-solution* when $h = 11$ the number of evaluated candidate edges is at most given by the number of upgraded edges, hence the similar CPU time with respect to $h = 1$ when the number of removed edges is 0 or 1.

Comparing the results of the ILP based heuristic with the MUCAG heuristics in terms of cost, the former has found the best solutions in 6 cases and near to best solutions in 2 cases (out of 12 cases). The results for the ILP based heuristic seem to suggest that it works particularly well for large values of D (although it also presents the minimum cost for $D = 40$ km when $\Lambda = 0.99998$). For $D = 160$ km, the number of possible solutions to explore is lower (than for smaller values of D) and it is likely that only a small number of second paths achieve one of the predefined values of Λ_z^w. The main drawback of this heuristic is the running time (at least 30 times higher than the running time of the fastest MUCAG variant, and sometimes even 1000 times higher). As mentioned in the description of this heuristic, a larger value of W should lead to better solutions (i.e. closer to the optimal solution) but it should entail an even higher resolution time. Also note that, in Table 6 the solution obtained for $D = 120, 160$ km is the same solution.

A comparison may also be performed in terms of the results with the different variants of MUCAG for selecting the edge to be upgraded, as described in subsection 4.3. By observing the results, we realize that the MinCost-MaxCount variant always leads to the highest number of upgraded edges; on the contrary, any of the three variants involving a MaxOn approach leads to a smaller number of upgraded edges. This is due to the fact that the MaxOn approach tends to select edges which lead to more path pairs achieving the desired availability and therefore, the total number of edges to be upgraded should not be very high. A comparison of the three variants involving a MaxOn approach shows that the MaxOn-MaxCount approach solutions have a slightly

27

higher number of upgraded edges.

The analysis in terms of the upgrade cost is not so clear. For smaller D, the MinCost-MaxCount variant tends to lead to solutions with smaller upgrade cost, as is noticeable in Tables 5–6 for $D = 40$ km and Tables 4 and 6 for $D = 80$ km. For lower geodiversity values, there should be many solutions for the second path with smaller total length and therefore higher availability. Therefore, the selection of shorter edges (i.e. with lower upgrade cost) for upgrade is more common.

For higher D, the variants MaxOn-MaxCount and MaxCount-MaxOn tend to present the best solutions in terms of upgrade cost: for $D = 120$ km, MaxOn-MaxCount always presents the best results; for $D = 160$ km, MaxOn-MaxCount leads to the best result in Table 4 and in Tables 5–6 it is the MaxCount-MaxOn that leads to the best results. For higher geodiversity values, the diversity of solutions is not great and therefore, the selection of more frequent edges which also leads to more path pairs achieving the desired availability seems to be the most appropriate one.

Two general conclusions are that: (i) the variants MinCost-MaxCount and MaxOn-MaxCount, that start by considering the edges in $\mathcal{E}_M(\mathcal{R})$ (i.e., the most frequent edges in the path pairs of $\mathcal{R}$), find solutions with higher number of edges to be upgraded, but a lower upgrade cost; (ii) the variants MinCost-MaxOn and MaxCount-MaxOn, that start by considering the edges in $\mathcal{E}_O(\mathcal{R})$ (i.e., edges whose upgrade makes the availability of more path pairs of $\mathcal{R}$ to become at least Λ), find solutions with a lower number of edges to be upgraded, but a higher upgrade cost.

All in all, considering only the edge selection variants in the MUCAG (i.e. excluding the ILP based heuristic), the MaxOn-MaxCount and the MaxCount-MaxOn seem to be good compromise variants for Germany50, although for some particular cases the MinCost-MaxCount allows for solutions with the smallest upgrade cost, as already mentioned. For all the distances and in all the tables, one of these variants always leads to the best solution in terms of either the number of upgraded edges or the cost of upgrade. The MaxOn-MaxCount approach solutions have a slightly higher number of upgraded edges but with lower upgrade cost, whereas the MaxCount-MaxOn solutions present a slightly higher cost of upgrade but with lower number of upgraded edges.

Fig. 5 presents the solutions found by MaxOn-MaxCount, one of the compromise variants, with the $h = 11$ strategy, for the four considered values of D. This figure illustrates that higher values of D lead to a higher number of required upgraded edges, as expected. Note that the set of upgraded edges for a given D is not a subset of the set of upgraded edges for a higher geodiversity

value. Therefore we cannot extrapolate the set of upgraded edges for a certain D, given the set of upgraded edges for another geodiversity value.

Fig. 6 presents the solutions found by MaxOn-MaxCount with $D = 40$ km, for the $h = 0$ and

625 $h = 1$ strategies. Comparing Fig. 6a ($h = 0$) and Fig. 5a ($h = 11$), we realize that the edge $\{3, 38\}$, followed by $\{7, 39\}$ did not need to be upgraded and the desired availability $\Lambda = 0.99999$ could still be provided for all the node pairs. With the $h = 1$ strategy (see Fig. 6b) we realize that again $\{7, 39\}$ and also $\{2, 35\}$ and $\{11, 15\}$ did not need to be upgraded. Note that in the $h = 1$ strategy, the aim is to find the subset of the upgraded edges with the largest cost and whose downgrade

630 still ensures the desired availability. These edges tend to be short edges with a high availability even before being upgraded. Therefore, they should have a minor contribution to the availability improvement of the different path pairs.

Fig. 7 presents the solutions found for $D = 40$ km and $h = 11$, by the ILP based heuristic and the other three MUCAG variants (the solution of MaxOn-MaxCount is in Fig. 5a). The ILP based

635 solution (see Fig. 7a) shows a set of upgraded edges which is very different from the one presented by the MUCAG variants. The MinCost-MaxCount variant (see Fig. 7b) presents a predominance of short edges being upgraded, which is expected as one of the criteria for edge selection is the minimization of the upgrade cost (related to the edge length). This leads to a solution where many short edges are upgraded, but the upgrade cost is not very high. The highest cost for the MUCAG

640 variants is obtained with the MinCost-MaxOn variant (see Fig. 7c), where a few long edges are considered. The solutions obtained with the MaxOn-MaxCount variant (see Fig. 5a) and the MaxCount-MaxOn variant (see Fig. 7d) present the same number of upgraded edges (14) but very different upgrade costs (a difference of 23.3%). Looking at the figures, it becomes apparent that the MaxOn-MaxCount approach solution has shorter and more central upgraded edges, whereas

645 the MaxCount-MaxOn solution has longer and less central upgraded edges, leading to a higher upgrade cost.

6.2. Results for the Coronet network

The Coronet network presents a much wider geographical coverage and our calculations show that four nines (0.9999) availability cannot be provided to many node pairs without upgraded

650 edges. In this case, we have considered scenarios for the geodiversity values D of 100 km, 200 km, 400 km and 600 km and for availability values of $\Lambda = 0.9999$ and $\Lambda = 0.99999$. The number of node
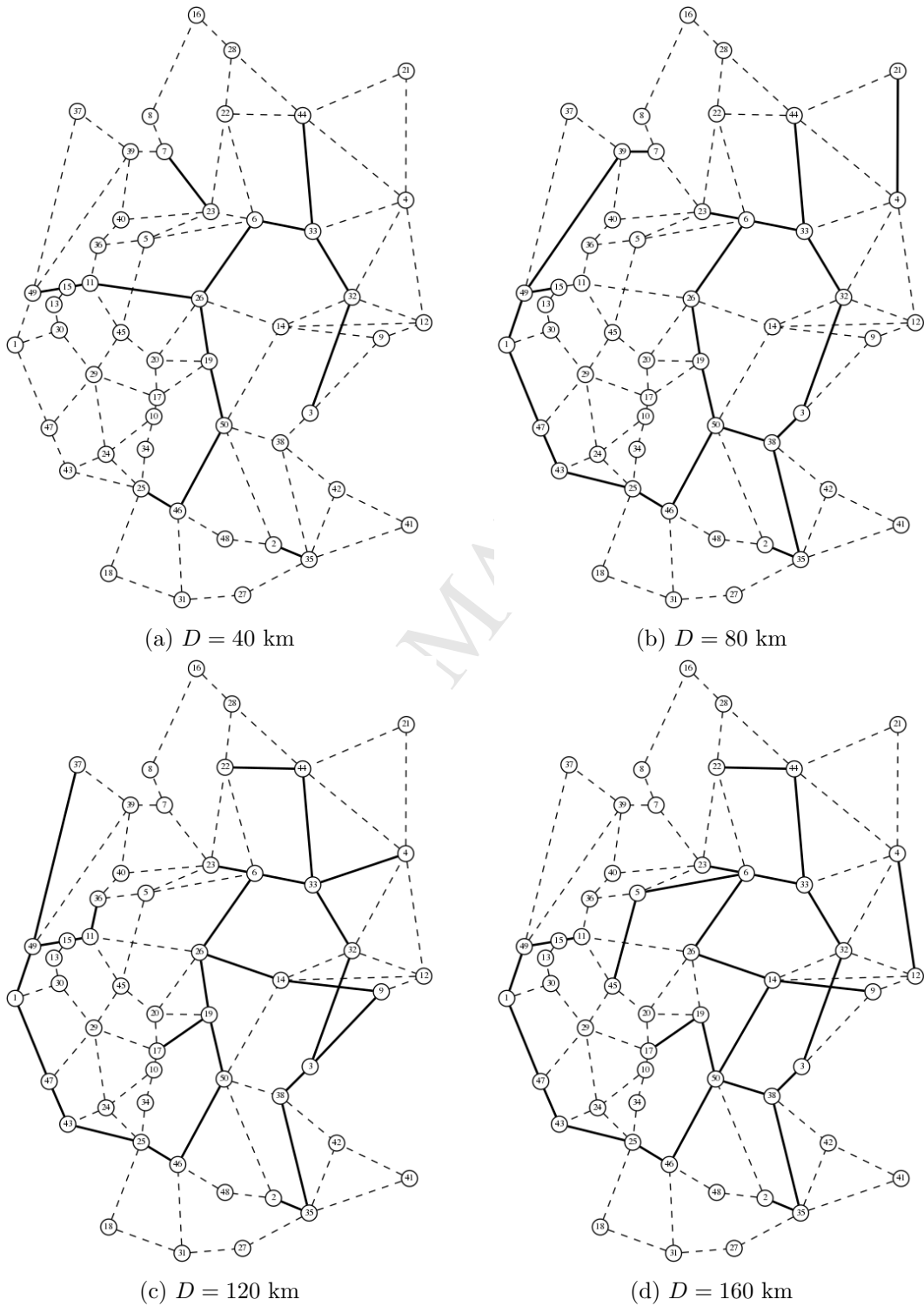
29

(a) $D = 40$ km

(b) $D = 80$ km

(c) $D = 120$ km

(d) $D = 160$ km

Figure 5: MaxOn-MaxCount, $h = 11$: upgraded edges for achieving availability $\Lambda = 0.99999$ between all node pairs in Germany50

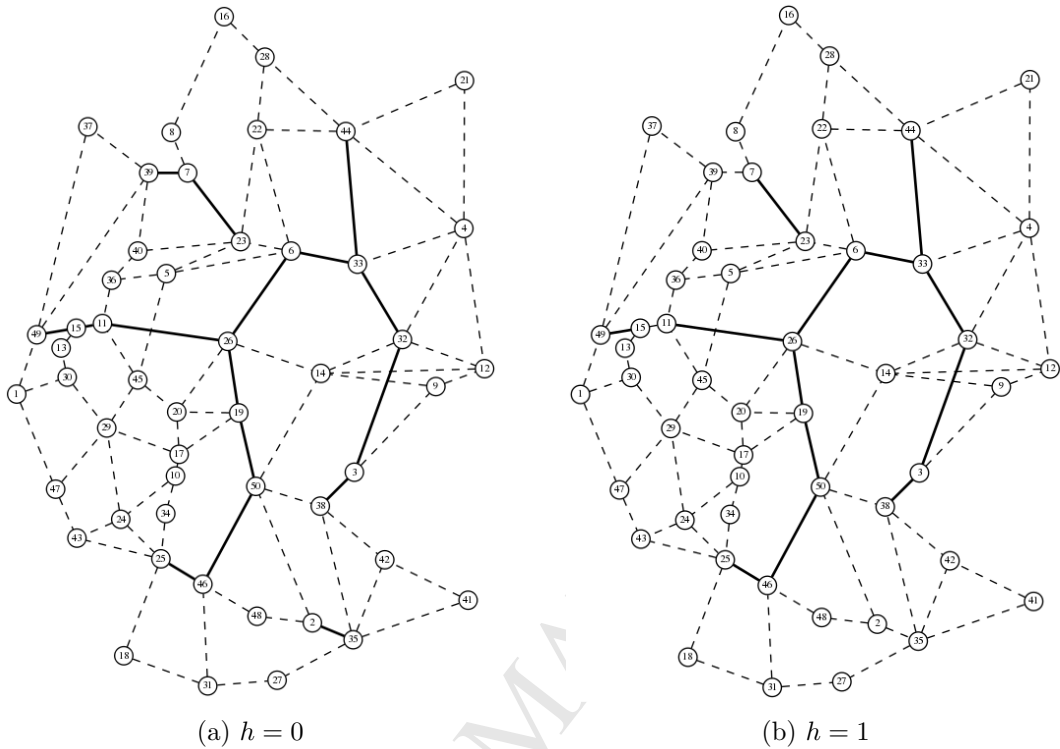(a) $h = 0$ (b) $h = 1$

Figure 6: MaxOn-MaxCount, $D = 40$ km: upgraded edges for achieving availability $\Lambda = 0.99999$ between all node pairs in Germany50

pairs that do not satisfy the required availability, for each value of D, before the upgrade of any edge is shown in Table 7. In Table 8 (for $\Lambda = 0.9999$) and Table 9 (for $\Lambda = 0.99999$), the solution values in terms of the number of upgraded edges, the upgrade cost and the CPU time are displayed.

655 Again, solutions providing the minimum number of upgraded edges and/or the minimum upgrade cost are highlighted in bold.

Note that in Table 9, no results are displayed for the ILP based heuristic. In fact, this heuristic is not effective at all in these cases as it runs a significant number of steps and the final solutions, even when filtered by function *filter-solution* (either with $h = 1$ or with $h = 11$) are much worse

660 than the ones found by the MUCAG heuristics. No results are also presented for the MinCost-MaxCount variant with $h = 1$ – after 24 h of running time, the *filter-solution* function was still trying to find the set of edges of minimum cost for possible downgrade.

It should be noted first that, although for $\Lambda = 0.99999$ the ILP based heuristic performs very poorly, for $\Lambda = 0.9999$ this heuristic, regardless of the value of h, attains the best solution in terms

665 of upgrade cost for all considered D values, as can be seen in Table 8. The best values with the

31

(a) ILP based heuristic

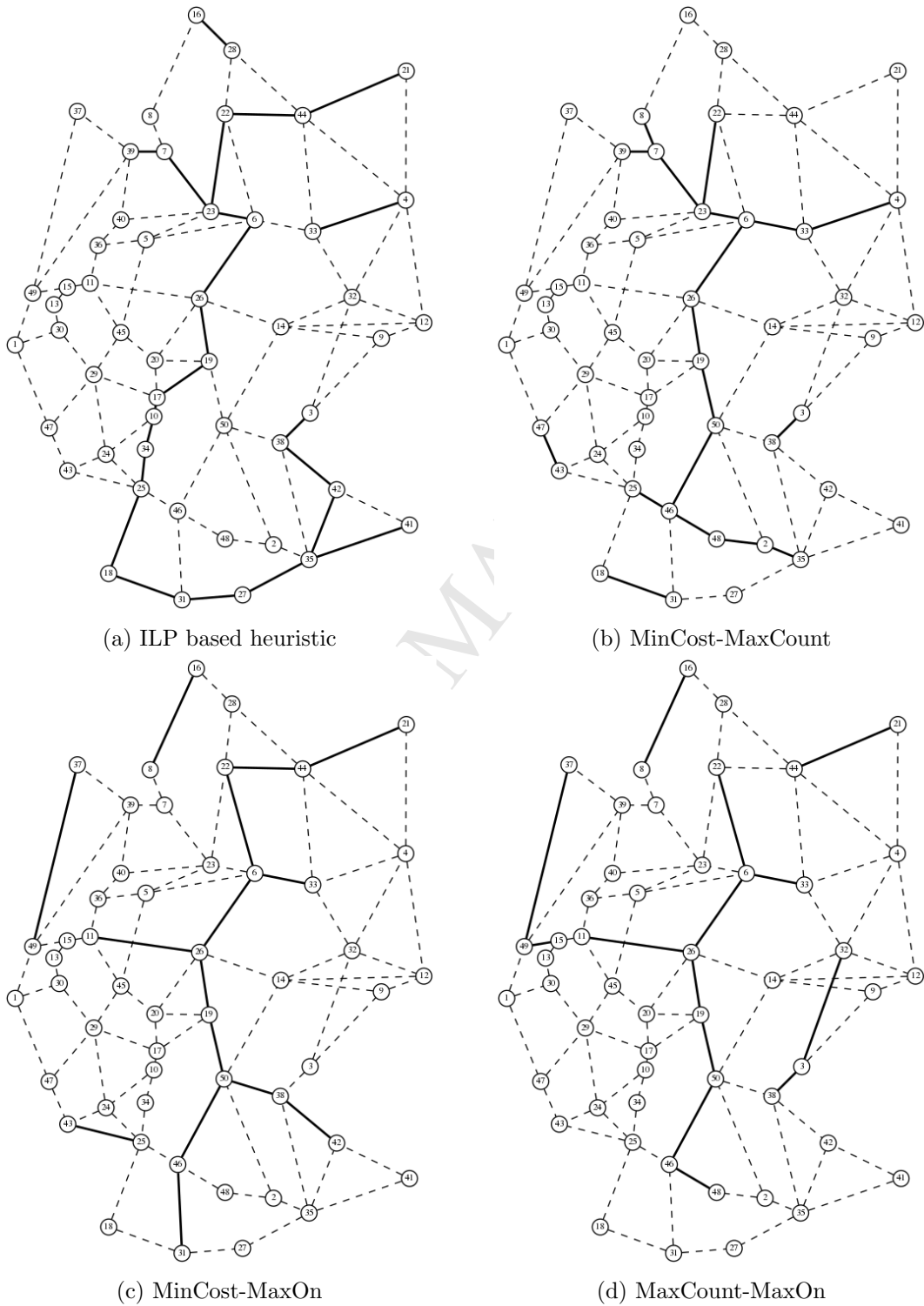(b) MinCost-MaxCount

(c) MinCost-MaxOn

(d) MaxCount-MaxOn

Figure 7: Upgraded edges for achieving availability $\Lambda = 0.99999$ between all node pairs in Germany50, when $D = 40$ km and $h = 11$

MUCAG variants are obtained with the MinCost-MaxCount, but with an increase up to 5.4% and 10.7% for $h = 11$ and $h = 0$, respectively. Note that the solution obtained for $D = 400, 600$ km is the same solution (as in the case of the results of Table 6 for the higher D). Moreover, concerning running times, the ILP based heuristic becomes more efficient for higher values of D while the

670 MUCAG variants become less efficient (in fact, for $D = 400, 600$ km, the running times are of the same magnitude for both methods).

The main observations are similar to the ones already mentioned for the Germany50 network. For higher D, solutions have both a higher number of upgraded edges and cost, and a higher running time, except in a few cases. Obviously, as the problem dimension is bigger than the one

675 for the Germany50 network, the running times are noticeably higher for the Coronet. In Table 8 for MinCost-MaxCount, $D = 400$ km, $h = 1$, we find a very high running time, which occurs because of a high size of the set of edges that may be downgraded, as already explained for a similar case in Table 4.

The resolution strategy with $h = 1$ leads to better results, but with a higher running time.

680 Again, the strategy with $h = 11$ seems to be a good compromise strategy: it takes less CPU time to produce a solution very close to the best solution (obtained with $h = 1$). However, for Coronet when $\Lambda = 0.99999$ the number of downgraded edges, resulting from applying *filter-solution* on the basic solution, varies between 2 and 10 (higher than in the case of Germany50 network), and the CPU time can be high even for strategy $h = 11$ due to the large number of upgraded edges of the

685 basic solution (see Table 9 for results of MinCost-MaxCount and $D = 600$ km).

In terms of the selection of edges for upgrade with the MUCAG variants: (i) when we analyze the number of upgraded edges, the variant with the best performance is MaxCount-MaxOn, for both values of Λ and the variant with the worst results is the MinCost-MaxCount, followed by the MaxOn-MaxCount; (ii) when we analyze the cost of upgraded edges, the MinCost-MaxCount

690 presents the best results, except for $D = 100$ km and $\Lambda = 0.9999$ where MaxCount-MaxOn

Table 7: Number of node pairs initially with availability less than Λ for Coronet

	D [km]			
Λ	100	200	400	600
0.9999	2061	2149	2184	2196
0.99999	2734	2737	2737	2737

33

Table 8: No. of upgraded edges ($\#\hat{e}$), cost and CPU time (in seconds) for achieving $\Lambda = 0.9999$ between all node pairs in Coronet

Algorithm		$D = 100$			$D = 200$			$D = 400$			$D = 600$		
		$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU
ILP based heuristic	$h = 0$	44	12648	7854	43	14198	3251	49	16519	1116	49	16519	947
	$h = 1$	42	**12420**	7968	40	**13940**	3423	44	**15989**	1876	44	**15989**	1700
	$h = 11$	43	12487	7919	42	14072	3396	48	16287	1315	48	16287	1176
MinCost--MaxCount	$h = 0$	45	13439	425	53	15493	617	56	18194	1127	56	18482	1149
	$h = 1$	44	13197	528	50	14521	1121	50	16448	14988	51	16953	4788
	$h = 11$	44	13197	507	50	14521	964	52	16586	1625	52	16978	1638
MinCost--MaxOn	$h = 0$	34	14593	297	39	17237	600	45	19610	1002	47	20584	1025
	$h = 1$	33	14120	352	**38**	16824	771.4	45	19610	1425	41	17893	2616
	$h = 11$	33	14120	348	**38**	16824	755	45	19610	1397	41	17893	1305
MaxOn--MaxCount	$h = 0$	38	13598	386	44	15130	581	46	17860	1037	47	17865	1083
	$h = 1$	37	13440	480	42	14650	1047	44	16846	3191	42	17012	4731
	$h = 11$	37	13440	475	42	14658	858	44	16846	1356	44	17315	1388
MaxCount--MaxOn	$h = 0$	33	13494	472	**38**	17031	603	44	19376	956	46	21066	985
	$h = 1$	**31**	12620	575	**38**	17031	770	**40**	18221	1433	**40**	17611	5807
	$h = 11$	**31**	12620	571	**38**	17031	755	**40**	18221	1326	**40**	17611	1347

Table 9: No. of upgraded edges ($\#\hat{e}$), cost and CPU time (in seconds) for achieving $\Lambda = 0.99999$ between all node pairs in Coronet

Algorithm		$D = 100$			$D = 200$			$D = 400$			$D = 600$		
		$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU	$\#\hat{e}$	Cost	CPU
ILP based heuristic		–	–	–	–	–	–	–	–	–	–	–	–
MinCost--MaxCount	$h = 0$	82	24534	1482	86	25966	2125	88	27617	6223	89	28350	7070
	$h = 1$	–	–	–	–	–	–	–	–	–	–	–	–
	$h = 11$	72	**21952**	2397	79	**23539**	5332	80	**25857**	23024	82	**26309**	34105
MinCost--MaxOn	$h = 0$	66	26499	1775	69	26939	13087	72	28274	16830	71	28786	13719
	$h = 1$	**62**	24003	7097	66	25582	15027	70	27771	23327	69	28120	23097
	$h = 11$	**62**	24139	2076	66	25582	14535	70	27771	22292	69	28120	20393
MaxOn--MaxCount	$h = 0$	68	23083	1140	72	24495	1650	79	28045	4219	79	28542	4689
	$h = 1$	66	23007	1562	66	23826	11829	76	27300	17756	73	27033	65898
	$h = 11$	66	23007	1530	70	24073	2998	76	27300	15296	76	27288	15783
MaxCount--MaxOn	$h = 0$	66	26499	1813	67	26949	4102	72	28696	15630	72	29247	17911
	$h = 1$	**62**	24003	7173	**63**	25207	5363	**68**	26926	19984	**67**	27064	27886
	$h = 11$	**62**	24139	2069	65	25833	5358	**68**	26926	18139	**67**	27064	20996

presents the min-cost solution after using *filter-solution*. These two conclusions reinforce/validate the observations already made for the Germany50 network, as we realized that the variants starting with the MaxCount approach seem to lead to solutions with a high number of upgraded edges, but a lower upgrade cost.

Hence in the case of Coronet network, the combination of MinCost-MaxCount *filter-solution* with $h = 1$ seems to be the best approach among the MUCAG variants for obtaining a solution with minimal upgrade cost, albeit upgrading a larger number of edges. Nevertheless, given all the results, MaxOn-MaxCount is a good compromise solution having a slightly higher upgrade cost (less than 3.4%) than MinCost-MaxCount solutions, but presenting a smaller number of upgraded edges, and requiring less CPU time. MaxCount-MaxOn may also be considered a good compromise solution, as it leads to the smallest number of edges for upgrade in all cases but with a higher upgrade cost (less than 17.3%) than MinCost-MaxCount solutions.

Due to the larger dimension of this network (when compared to the dimension of the Germany50 network), figures with the obtained solutions are harder to analyze. Fig. 8 presents the solutions found for $\Lambda = 0.9999$, $D = 400$ km and $h = 11$, for the ILP based heuristic (it has the best upgrade cost), the MinCost-MaxCount (it has the best upgrade cost among the MUCAG variants) and the compromise variants, MaxOn-MaxCount and MaxCount-MaxOn.

In this figure, it is possible to check that the MinCost-MaxCount strategy (see Fig. 8b) has a much higher number of upgraded edges, many of them short, leading to an upgrade cost which is not very high. On the contrary, the MaxCount-MaxOn strategy (see Fig. 8d) has a lower number of upgraded edges, many of them long, leading to a high upgrade cost.

7. Conclusions and Further Work

Telecommunication networks, being one of the underlying infrastructures supporting critical services of our society, are expected to provide high end-to-end availability services and to ensure the resilience of end-users applications even in the presence of large-scale natural disasters.

Several protection mechanisms may be considered, namely path protection, to enhance network robustness. However, although path protection improves end-to-end availability, it may not suffice to achieve the level of availability required by critical services. Geodiverse routing contributes to enhance wide area disaster resilience of networks as it ensures two routing paths with some geographic separation, reducing the probability of both being affected by a single disruptive event.

35

(a) ILP based heuristic

(b) MinCost-MaxCount

(c) MaxOn-MaxCount
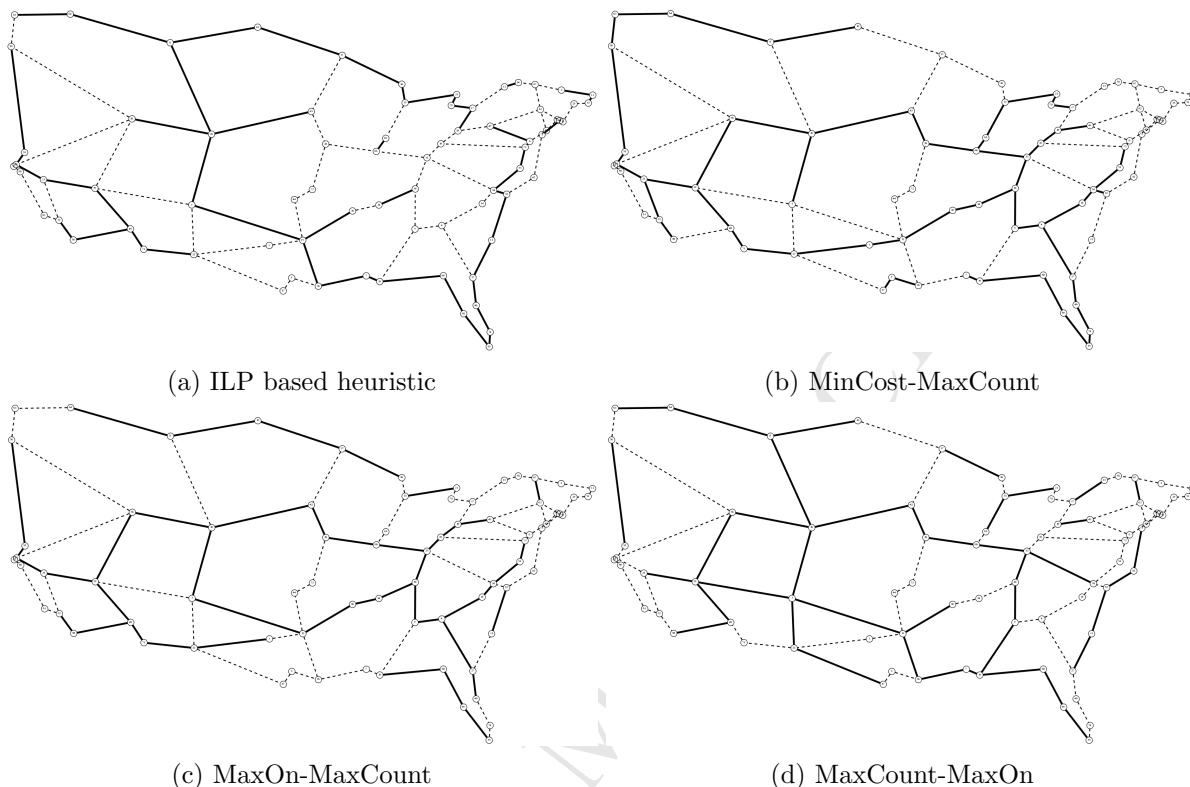
(d) MaxCount-MaxOn

Figure 8: Upgraded edges for achieving availability $\Lambda = 0.9999$ between all node pairs in Coronet, when $D = 400$ km and $h = 11$

Jointly considering geodiverse routing and availability constraints, to guarantee high end-to-end availability services is a challenging problem.

The selection of edges to be upgraded at a minimum cost to ensure a required level of availability and geodiversity was addressed in this work. In [5] a path based formulation was proposed for this problem but it is not practical due to the exponential number of variables. Thus, an integer non-linear programming model is introduced, which uses an arc based formulation; this is next reformulated as an ILP model, using approximate linear constraints to replace the original non-linear ones of the former formulation. This resulted in a ILP based heuristic.

Hence, two different resolution strategies were put forward: one based on a greedy approach (for which some variants were already proposed in [5]) and a novel ILP based approach. Both resolution methods add links iteratively. Once a set of links for upgrade is devised, it was noticed that the required end-to-end availability and geodiverse routing distance could still be provided without the upgrade of some of those links. Therefore the solutions are analyzed and redundant

36

upgraded arcs have their availability downgraded to its original value, reducing the total cost of the solution without compromising neither the desired level of geodiversity nor of the availability. Although an exhaustive search procedure was put forward, as it may become computationally prohibitive, a more computationally efficient procedure was also proposed resulting in only slightly worse solutions.

The computational results have shown that both the greedy approach heuristics and the ILP based heuristic can be applied to real optical core networks. The ILP based heuristic provides better cost solutions when the required geodiversity is more stringent, but it requires longer runtime and does not cope well with larger number of node pairs requiring higher availability values.

Regarding future work, note that due to the non-linear nature of the problem, an exact method is hard to be derived. Nevertheless, mathematical methods able to obtain lower bounds for the cost of the solutions will be pursued. It is expected that the knowledge of those bounds will allow to develop more effective heuristics. Moreover, the proposed methods are to be extended for more general cases of upgrade costs and upgrade alternatives (i.e., having more than one upgrading possibility for each existing edge).

Acknowledgment

References

[1] A. Csoma, A. Gulyas, L. Toka, On measuring the geographic diversity of Internet routes, IEEE Communications Magazine 55 (5) (2017) 192–197.

[2] S. Trajanovski, F. A. Kuipers, A. Ilić, J. Crowcroft, P. V. Mieghem, Finding critical regions and region-disjoint paths in a network, IEEE/ACM Transactions on Networking 23 (3) (2015) 908–921. doi:10.1109/TNET.2014.2309253.

[3] A. de Sousa, D. Santos, P. Monteiro, Determination of the minimum cost pair of D-geodiverse paths, in: The 2017 International Conference on Design of Reliable Communication Networks (DRCN 2017), Munich, 2017.

[4] J. Zhang, E. Modiano, D. Hay, Enhancing network robustness via shielding, IEEE/ACM Transactions on Networking 25 (4) (2017) 2209–2222. doi:10.1109/TNET.2017.2689019.

[5] A. de Sousa, T. Gomes, R. Girão-Silva, L. Martins, Minimizing the network availability upgrade cost with geodiversity guarantees, in: 9th International Workshop on Resilient Networks Design and Modeling (RNDM), Alghero, Italy, 2017.

[6] S. Orlowski, R. Wessäly, M. Pióro, A. Tomaszewski, SNDlib 1.0–Survivable Network Design library, Networks 55 (3) (2010) 276–286. doi:10.1002/net.20371.

[7] M. F. Habib, M. Tornatore, F. Dikbiyik, B. Mukherjee, Disaster survivability in optical communication networks, Computer Communications 36 (2013) 630–644.

[8] P. K. Agarwal, A. Efrat, S. K. Ganjugunte, D. Hay, S. Sankararaman, G. Zussman, The resilience of WDM networks to probabilistic geographical failures, IEEE/ACM Transactions on Networking 21 (5) (2013) 1525–1538.

[9] M. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, B. Mukherjee, Design of disaster-resilient optical data-center networks, Journal of Lightwave Technology 30 (16) (2012) 2563–2573. doi:10.1109/JLT.2012.2201696.

[10] X. Long, D. Tipper, T. Gomes, Measuring the survivability of networks to geographic correlated failures, Optical Switching and Networking 14 (Part 2) (2014) 117 – 133, Special Issue on RNDM 2013. doi:10.1016/j.osn.2014.05.004.

[11] D. S. Yadav, A. Chakraborty, B. S. Manoj, A multi-backup path protection scheme for survivability in elastic optical networks, Optical Fiber Technology 30 (2016) 167–175.

[12] V. Miletic, D. Maniadakis, B. Mikac, D. Varoutas, On the influence of the underlying network topology on optical telecommunication network availability under shared risk link group failures, in: Proceedings of the 10th International Conference on the Design of Reliable Communication Networks (DRCN), 2014.

[13] M. Aibin, K. Walkowiak, A. Sen, Software-defined adaptive survivability for elastic optical networks, Optical Switching and Networking 23 (2) (2017) 85–96.

[14] J. Rak, D. Hutchison, E. Calle, T. Gomes, M. Gunkel, P. Smith, J. Tapolcai, S. Verbrugge, L. Wosinska, Recodis: Resilient communication services protecting end-user applications from disaster-based failures, in: 18th International Conference on Transparent Optical Networks (ICTON), 2016, invited paper.

[15] D. Tipper, Resilient network design: challenges and future directions, Telecommunication Systems 56 (1) (2014) 5–16. doi:10.1007/s11235-013-9815-x.

[16] A. Alashaikh, T. Gomes, D. Tipper, The spine concept for improving network availability, Computer Networks 82 (0) (2015) 4–19. doi:10.1016/j.comnet.2015.02.020.

[17] A. Alashaikh, D. Tipper, T. Gomes, Supporting differentiated resilience classes in multilayer networks, in: 12th International Conference on the Design of Reliable Communication Networks (DRCN), 2016, pp. 31–38. doi:10.1109/DRCN.2016.7470832.

[18] T. Gomes, J. Tapolcai, C. Esposito, D. Hutchison, F. Kuipers, J. Rak, A. de Sousa, A. Iossifides, R. Travanca, J. André, L. Jorge, L. Martins, P. O. Ugalde, A. Pašić, D. Pezaros, S. Jouet, S. Secci, M. Tornatore, A survey of strategies for communication networks to protect against large-scale natural disasters, in: 8th International Workshop on Resilient Networks Design and Modeling (RNDM), 2016, pp. 11–22. doi:10.1109/RNDM.2016.7608263.

[19] J. Rohrer, A. Jabbar, J. P. G. Sterbenz, Path diversification: a multipath resiliency mechanism, in: 7th International Workshop on the Design of Reliable Communication Networks – DRCN 2009, Washington, DC, USA, 2009, pp. 343–351.

[20] Y. Cheng, M. T. Gardner, J. Li, R. May, D. Medhi, J. P. Sterbenz, Analysing geopath diversity and improving routing performance in optical networks, Computer Networks 82 (2015) 50–67. doi:https://doi.org/10.1016/j.comnet.2015.02.021.

[21] Y. Cheng, J. P. G. Sterbenz, Critical region identification and geodiverse routing protocol under massive challenges, in: 2015 7th International Workshop on Reliable Networks Design and Modeling (RNDM), 2015, pp. 14–20. doi:10.1109/RNDM.2015.7324303.

[22] T. Gomes, J. Craveirinha, Efficient calculation of the most reliable pair of link disjoint paths in telecommunication networks, European Journal of Operational Research 182 (3) (2007) 1055–1064.

[23] J. Y. Yen, Finding the k shortest loopless paths in a network, Management Science 17 (11) (1971) 712–716.

[24] E. Martins, M. Pascoal, J. Santos, An algorithm for ranking loopless paths, Tech. Rep. 99/007, CISUC (1999).

[25] E. Martins, M. Pascoal, J. Santos, Deviation algorithms for ranking shortest paths, International Journal of Foundations of Computer Science 10 (3) (1999) 247–263.

[26] D. A. Schupke, F. Rambach, A link-flow model for dedicated path protection with approximative availability constraints, IEEE Communications Letters 10 (9) (2006) 679–681.

[27] IBM ILOG CPLEX Optimization Studio V12.6.1, IBM, 2014.

[28] J.-P. Vasseur, M. Pickavet, P. Demeester, Network Recovery – Protection and Restoration of Optical, SONET-SDH, IP, and MPLS, Elsevier, 2004.