A centrality-based heuristic for network design to support availability differentiation*

Rita Girão-Silva†‡ Teresa Gomes†‡ Lúcia Martins†‡
David Tipper§ Abdulaziz Alashaikh¶

{rita, teresa, lucia}@deec.uc.pt dtipper@pitt.edu asalashaikh@uj.edu.sa

January 2020

*This work was funded by ERDF Funds through the Centre's Regional Operational Program and by National Funds through the FCT – Fundação para a Ciência e a Tecnologia, I.P. under the project CENTRO-01-0145-FEDER-029312. This work was also partially supported by FCT under project UIDB/00308/2020.

†University of Coimbra, Department of Electrical and Computer Engineering, Coimbra, Portugal
‡Institute for Systems Engineering and Computers at Coimbra (INESC Coimbra), Coimbra, Portugal
§Department of Informatics and Networked Systems, University of Pittsburgh, Pittsburgh, USA
¶Department of Computer and Networks Engineering, University of Jeddah, Jeddah, Saudi Arabia

Abstract

In today's society, communication networks are of paramount importance, and providing adequate levels of availability of the resources in a cost effective way is crucial for network managers. We consider the design of a high availability structure (a spine) in the network, so that a desired availability for the network flows may be achieved. The tackled problem involves the selection of the edges forming the spine and the selection of the enhanced availability for each of those edges, aiming at fulfilling a prespecified availability value for each flow, at minimum cost. We solve the formulated Mixed Integer Linear Problem (MILP) for small networks, which allows us to identify some characteristics of the spanning tree formed by the set of links with upgraded availability. Afterwards, using that information, we propose a heuristic based on a centrality measure, which allows us to devise the appropriate set of links, and which may be used in larger networks. Experimental results show the effectiveness of the resolution approach in finding spines equal to the optimal ones or to the best known solutions.

Keywords: availability, resilience, heuristic, centrality, cost functions.

1 Introduction

In today's society, communication networks are critical as they provide numerous services that support day-to-day activities (e.g. financial transactions, smart grid communications), as well as services that are necessary in the event of failures or disasters (e.g. emergency calls). The provision of adequate levels of availability for every demand in a cost effective way is of paramount importance for network managers.

The concept of embedding a high availability substructure (termed the spine) in the network, as described in [16], is considered here. This substructure should be composed of a set of nodes and edges, and should be used by traffic needing a high level of availability. After tackling the problem of the identification of such a set of edges in [2, 3], the same authors proposed the use of cost functions associated with the spines [4], in order to assess the investment needed by network operators to achieve a certain level of required availability. Different cost functions were developed from the existing literature, since real cost data is difficult to obtain.

The spine concept can also be used as part of a strategy for disaster preparedness. The work in [8] focuses on the minimization of the cost of upgrading edges to ensure a certain level of availability, where geodiversity [7] was also taken into account, in order for the network to be more disaster resilient. In [13], the concept of the spine as a high availability structure is used in a framework that also encompasses an additional upgrade of the availability of the edges, so that even after the most probable disaster events the network remains connected, allowing for the survivable routing layer to protect the connections.

In this work, we consider designing a high availability substructure (spine) formed as a spanning tree, similar to [2,3]. For each demand, 1+1 protection is required, i.e. a working (or active) path (WP) is defined, along with an edge-disjoint backup path (BP). The WP is routed over the spine and the BP may use any edges of the network (except the edges of the WP). As the BP may use edges on the spine, it becomes easier to find feasible solutions for each demand, and the availability of the path pair for each demand tends to be higher. The downside is that the edges of the spine will require a higher capacity, but in this work we do not deal with capacity restrictions of the edges.

The work in [4] aims to find a high availability structure with minimal cost of availability upgrade, guaranteeing a minimum availability value for each WP. The proposed approach, however, does not scale for larger networks. Hence we present a scalable heuristic approach divided into two stages, as we first try to identify the edges (or some of them) that should form the spine (using a heuristic approach based on a centrality metric), and at a second stage calculate the remaining edges (if necessary) and the availability for all the edges in the spine (using the exact problem formulation). In effect the first stage reduces the search space over which the optimization model is solved, which greatly speeds up the computation.

After this introductory section, we present the optimization problem to be tackled in Section 2. This section is followed by details on the heuristic solution approach in Section 3. In Section 4, some experimental results are displayed and commented on. The paper ends with our conclusions and proposals for future work.

2 Problem Definition

The notation used in the problem formulation is displayed in Table 1. The parameters used in the heuristic are also given in the table.

A network is represented by an undirected graph $G = (V, E)$, where V is the set of nodes and E is the set of undirected edges representing bidirectional connections between the nodes. Each undirected edge may be represented as a pair of directed links in opposite directions pertaining to a set E_d. For each edge $e \in E$, we define an initial availability value $a_0(e) \in [0, 1]$.

The purpose of this work is to devise a high availability structure (i.e., a spine) with minimal cost of availability upgrade, so that a minimum value for the availability of each WP, $\widehat{a_{WP}}$, is achieved. The WP must include edges of the spine only. As for the BP for a particular demand, it must be edge-disjoint with the corresponding WP for that demand. The BPs may use edges on or off the spine. A spine for which an edge-disjoint pair of paths exists under these conditions is considered a feasible spine.

In order to achieve the desired availability values, some of the edges may have their availability upgraded while others may eventually have their availability downgraded, if it is deemed too high for the established availability goals. Therefore, it is possible to transfer some maintenance and repair capabilities between edges, which may be interesting for a company to explore [5].

As mentioned previously, $a_0(e)$ is the initial availability of edge $e \in E$; let $\breve{a}(e)$ be an upgraded or downgraded availability of the same edge. Following [4, Eqs.(3)-(6)], an upgrade cost (if positive) or a downgrade profit (if negative) may be calculated, considering different cost functions. We will focus on cost function f_{c3} of [4], i.e., the cost of upgrade (or downgrade) is given by

$$\mathcal{C}(e) = -\ln\left(\frac{1-\breve{a}(e)}{1-a_0(e)}\right)\ell(e)$$

with $\ell(e)$ [km] representing the length of edge e. In addition, we consider K different target availability values for each edge, i.e., we assume $\breve{a}(e)$ may take one of K possible values $a_k, k = 1, ..., K$ regardless of the initial availability value of each edge $a_0(e)$. Ultimately, the goal of this problem is to find the edges that should form the spine ($e \in \mathcal{S}$) and their final availability values, with minimal cost of upgrade, while satisfying the desired availability value $\widehat{a_{WP}}$.

We provide some information on the problem formulation so that the text is self-contained. For further details, see [4]. The problem is formulated in terms of directed links $(i, j) \in E_d$. The notation in [4] is used: the binary variables x_{ij} are 1 if link (i, j) is in the spine and 0 otherwise; r_{ij}^k are 1 if the final availability of link (i, j) is $a_0(i, j)$ ($k = 0$) or a_k, with $k = 1, ..., K$. The cost of upgrading (or downgrading) the availability of link (i, j) is redefined as $\mathcal{C}_{ij}^k = -\ln\left(\frac{1-a_k}{1-a_0(i,j)}\right)\ell(i, j)$, $k = 1, ..., K$. Obviously, $\mathcal{C}_{ij}^0 = 0$, $\forall(i, j)$.

The spine is obtained by solving a mixed integer linear optimization problem with objective function $\min \sum_{(i,j)\in E_d, i<j} \sum_{k=1}^{K} r_{ij}^k \mathcal{C}_{ij}^k$ (as in [4, Eq.(7)]), subject to constraints [4,

Table 1: List of parameters

Network topology	
$G = (V, E)$	undirected graph
V	set of nodes
$i, j, n, s, t \in V$	nodes
E	set of undirected edges
$e \in E$	undirected edge
E_d	set of directed links
$(i, j) \in E_d$	directed link from i to j
$\ell(e)$ or $\ell(i, j)$	length [km] of edge e or link (i, j)
$\mu(s, t)$	length [km] of shortest path between nodes s and t
Availability parameters	
$a_0(e)$ or $a_0(i.j)$	initial availability of edge e or link (i, j)
$\breve{a}(e)$ or $\breve{a}(i, j)$	upgraded or downgraded availability of edge e or link (i, j)
a_k	pre-defined k-th possible value for the availability of an edge $(k = 1, ..., K)$
$\widehat{a_{WP}}$	minimal availability of each WP
Cost parameters used in the MILP	
$\mathcal{C}(e)$	upgrade cost (if positive) or downgrade profit (if negative) of edge e
$\mathcal{C}_{ij}^k$	upgrade cost (or downgrade profit) of link (i, j) if the final availability of the link is a_k $(k = 1, ..., K)$; for $k = 0$, $\mathcal{C}_{ij}^0 = 0$
Binary decision variables used in the MILP	
x_{ij}	1 if link (i, j) is in the spine; 0 otherwise
r_{ij}^k	1 if the final availability of link (i, j) is $a_0(i, j)$ with $k = 0$ or a_k $(k = 1, ..., K)$; 0 otherwise
Cost and other parameters used in the heuristic	
$c_H(e)$	harmonic centrality measure of edge e
$\mathscr{C}(e)$	centrality cost of edge e
$\mathscr{C}_{\text{Prim}}(e)$	cost of edge e to be used in the least cost spanning tree calculation

Eqs.(8)-(13),(15)-(17),(19),(22),(24)-(28)]. Constraint [4, Eq.(13)] maintains spine feasibility by imposing the disjointness constraint on the WP and BP of each demand. Constraint [4, Eq.(17)] is necessary to guarantee that the spine will be a spanning tree. The other constraints are used for path computation and availability calculations. In particular, constraint [4, Eq.(22)] is required for guaranteeing a minimal availability value $\widehat{a_{WP}}$ for the availability of each WP while minimizing the total upgrade cost. In our formulation, [4, Eq.(18)] is replaced with $\sum_{k=0}^{K} r_{ij}^{k} = x_{ij}$, $\forall (i,j) \in E_d, i < j$, which guarantees that only the edges of the spine may have their availability changed. The outputs of the optimization problem are the edges selected to form the spine and their modified (i.e., upgrade, downgrade, no change) availability value. Note that the optimization problem is NP-complete as discussed in [4], thus efficient heuristics are needed to provide scalable solutions.

3 Solution Approach

In our solution approach, the problem is divided into two stages: (i) first, we start by identifying a spanning tree (or at least some edges that should be part of the desired spanning tree); (ii) afterwards, given some or all of the edges of the spanning tree, we proceed to solving a problem of devising the remaining edges of the spanning tree (if necessary) and selecting the modified availability values for each edge in the spine.

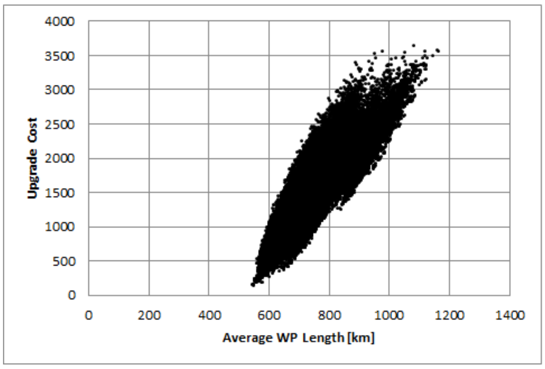
3.1 Identification of an Appropriate Spine

An appropriate spine is one that meets the target availability value, $\widehat{a_{WP}}$, for each demand with minimal edge availability upgrade cost. In order to identify such one, we started by performing a preliminary study inspecting the structural features of the spines found in a network. Table 2 shows the networks used in the study: polska network [12], spain network [11], and italia network [6]. For the first network, distances between nodes were calculated based on $x-y$ coordinates provided in the files associated with [12]; for the latter two networks, aerial distances between the cities in the nodes were considered.

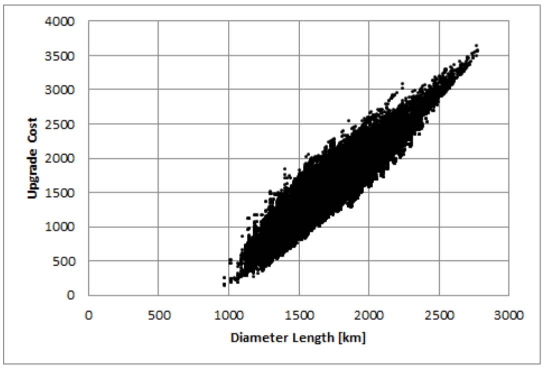
For each edge $e \in E$, we define an initial availability value such as in [4, Eqs.(1)-(2)]: $a_0(e) = 1 - \frac{MTTR}{MTBF(e)} \in [0,1]$. The mean time to repair a failure is $MTTR = 24$ h and the mean time between failures is $MTBF(e) = \frac{CC*365*24}{\ell(e)}$ [h]. The parameter CC represents the cable cut metric, which is set to 450 km here. In the networks studied, $K = 4$ and $a_1 = 0.995$, $a_2 = 0.999$, $a_3 = 0.9995$, $a_4 = 0.9999$.

Table 2: Network characteristics ($|V|$, $|E|$, γ – average node degree, δ – diameter, $|\mathcal{T}|$ – total number of trees, $|\mathcal{T}_f|$ – total number of feasible trees)

Network	$\|V\|$	$\|E\|$	γ	δ	$\|\mathcal{T}\|$	$\|\mathcal{T}_f\|$
polska	12	18	3.00	4	5 161	1 862
spain	14	22	3.14	5	40 436	22 037
italia	14	29	4.14	3	1 194 812	n.a.



(a) Upgrade cost as a function of average WP length [km]

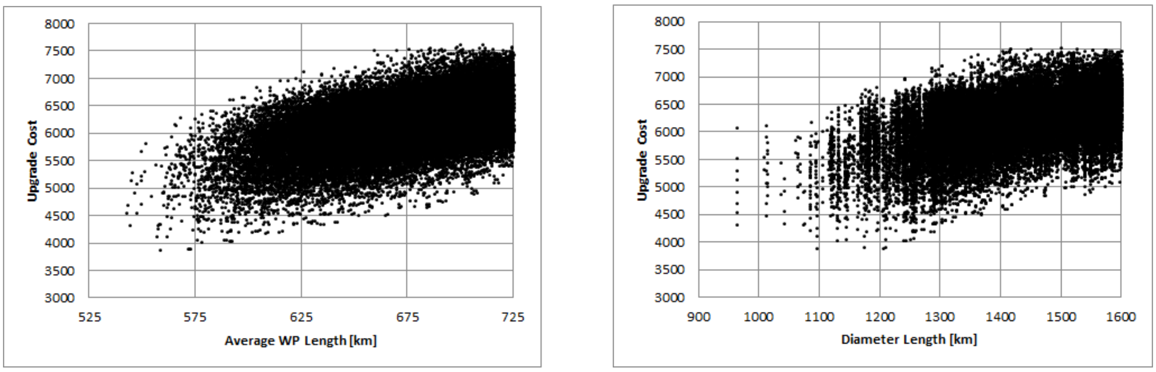


(b) Upgrade cost as a function of diameter length [km]

Figure 1: Upgrade cost for italia network, with target availability level $\widehat{a_{WP}} = 0.995$, considering the first 100 000 trees of minimal length

We focused our study on the italia network, which has a high number of feasible spines. The set of 100 000 spanning trees of minimal total length ($\mathcal{L}_\mathcal{S}$, given by the sum of lengths of all the edges in the tree) was obtained [10], of which 93.64% are feasible, i.e. it is possible to find an edge-disjoint path pair for every demand. Given the set of feasible spines, we calculated the total cost of upgrading the edges of each of those spines, so that the spine which allowed for a minimal cost could be identified. Note that in some cases it may not be possible to find a solution for the problem, i.e. given the possible upgraded availability values $a_k, k = 1, ..., K$, it might not be possible to find a solution with the target availability level $\widehat{a_{WP}}$ for a given feasible tree.

Fig. 1 shows the upgrade cost to achieve a target availability level $\widehat{a_{WP}} = 0.995$ in the italia network as a function of the average WPs length in each spine $\bar{\mathcal{L}}_{WP}$ (Fig. 1a) and as a function of the diameter length of each spine $di_\mathcal{S}$ (Fig. 1b), both in km. The figures clearly show that the trees for which the average length of the WPs and the diameter length are smaller lead to the least cost solutions when it is necessary to upgrade the links to achieve the desired availability level. The conclusion is similar to a target availability level



(a) Upgrade cost as a function of average WP length [km]

(b) Upgrade cost as a function of diameter length [km]

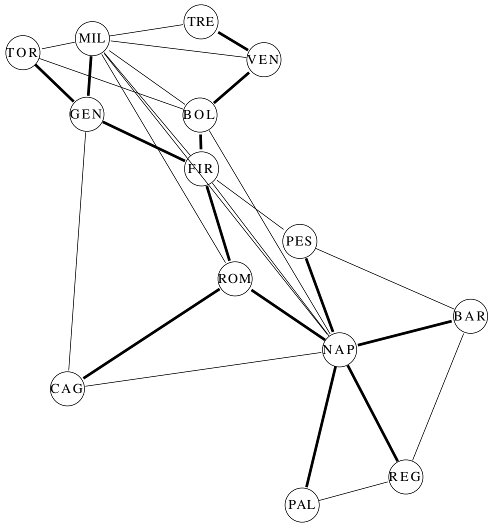
Figure 2: Upgrade cost for italia network, with target availability level $\widehat{a_{WP}} = 0.999$, considering the first 100 000 trees of minimal length (zoom focusing on the trees with least upgrade cost)

$\widehat{a_{WP}} = 0.997$. However, for $\widehat{a_{WP}} = 0.999$, the observations are not identical – see Fig. 2, where a zoom for the trees with least upgrade cost is displayed. In this latter case, there are multiple solutions with smaller diameter length and smaller average WP length, for which the upgrade cost is higher than the minimal value. Therefore, it is difficult to pinpoint the features of the spanning tree associated to the least cost solution for availability upgrade.

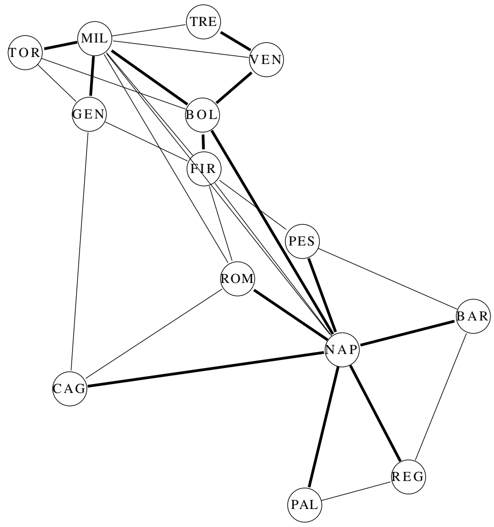
Nevertheless, finding spanning trees (or some edges of a spanning tree) with small average length of the WPs or small diameter length may be an initial step, followed by a selection of the availability upgrade for the edges of the spanning tree. In fact, as can be observed in Fig. 3b, a solution with higher average length of the WPs and higher diameter length may have an upgrade cost only slightly above the minimal cost.

The tree in Fig. 3a is one of those with smallest diameter (and it actually is the one with lowest upgrade cost when the purpose is to achieve $\widehat{a_{WP}} = 0.995$ or 0.997); it has a total length of 2 325 km, diameter of 962 km and average WP length of 544.18 km. The tree in Fig. 3b has an upgrade cost only 1.24% above the one in Fig. 3a, when the target availability is $\widehat{a_{WP}} = 0.997$. The tree in Fig. 3b differs in four edges from the other one and it has a total length of 2 602 km (11.91% higher), a higher diameter (1 144 km, i.e. 18.92% higher) and a higher average WP length (592.98 km, i.e. 8.97% higher). Still, in number of hops, it has a smaller diameter and a smaller number of average length for the WPs (because it relies on longer edges) than the spine in Fig. 3a. Both trees have a star-shaped configuration, as expected in this type of problem [4].

The main drawbacks of focusing in the trees featuring these characteristics are: (i)



(a) Spine with least upgrade cost for achieving $\widehat{a_{WP}} = 0.995$ (optimal solution) or 0.997 (best solution considering the first 100 000 trees of minimal length)



(b) Spine with upgrade cost 1.24% above the best upgrade cost for achieving $\widehat{a_{WP}} = 0.997$

Figure 3: Spines for italia network, for achieving a desired target availability level

many different solutions with close values of average length of the WPs or diameter length may lead to solutions with very different upgrade costs; (ii) solutions with similar edges (sometimes only with one different edge) may have different features (see Fig. 3), but still lead to solutions with close values of upgrade cost. To deal with this, rather than trying to devise the whole spine and then upgrading its edges, we will focus only on the most central part of a spine obtained with a heuristic (first step of resolution) by removing the edges with a leaf of the tree as a terminal node, and leave it for the exact resolution (second step of resolution) to find the remaining edges and the necessary availability upgrade to achieve the desired availability level.

3.2 Heuristic Resolution Approach

The heuristic proposed here is an improved version of the one presented in [9].

In this version, a certain degree of uncertainty is introduced, so that an edge should be avoided for a maximum number (rather than a fixed number) of iterations. Another difference with this version is that spines which are not feasible may also be accepted. Both these options should allow for a higher diversity of the possible spanning trees. Note that at the end of the heuristic only a set of the edges of the spine will actually be considered, as explained later, which allows for the use of infeasible spines. The feasibility of the final solution is always guaranteed by the exact problem resolution, through constraint [4, Eq.(13)].

The heuristic is as described next.

Input: $G = (V, E)$, $maxIter$, $totalSeeds$, $maxEdges$.

Output: Set of edges from the spanning tree with minimal diameter (in number of hops) and among those, the one with minimal diameter length (in km).

for $seed \leftarrow 1$ **to** $totalSeeds$ **do**
  $currentSeed \leftarrow seed$.
  for $listReset \leftarrow 0$ **to** 1 **do**
    if $listReset = 1$ **then**
      Reset the current list of edges to be avoided.
    end if
    loop
      The first time this inner loop is run, no edges to be avoided are defined; the following $|E|$ times, one edge at a time should be avoided, with the edges being selected in decreasing order of centrality cost (i.e. the least central edges are avoided first).
      Select new edge to be avoided for a random number of iterations ($\leq maxIter$); the random value is calculated using $currentSeed$.
      Increase $currentSeed$.
      repeat
        For each edge, calculate centrality cost $\mathscr{C}(e)$ and $\mathscr{C}_{\text{Prim}}(e)$ to be used in the spanning tree calculation.
        Calculate spine that tries to avoid edges on the list and minimizes the total centrality cost (with a penalty), using Prim's algorithm [14].
        if spine is feasible **then**
          Calculate BPs for every demand.
        end if
        Calculate the diameter in hops (maximal number of hops for any path in this spine) and the diameter length (maximal length of any path in this spine).
        Keep information regarding this spine if it has minimal diameter in hops or minimal diameter length among the trees with minimal diameter in hops.

19: Update list of edges to be avoided by decreasing the number of iterations during which they should be avoided.
20: **until** a feasible spine is found
21: **end loop**
22: **end for**
23: **end for**
24: Identify edges terminating in a leaf of the tree. Among those edges, eliminate those with higher centrality cost, up to a maximum of $maxEdges$ edges.

The centrality cost mentioned in lines 8 and 12 is based on the harmonic centrality [15], which assigns each edge a value measuring how close the terminal nodes of an edge are to the other nodes. Let $\mu(s,t)$ represent the length (in km) of the shortest path between nodes $s, t \in V$. For edge $e \equiv (i,j) \in E$, we may define the harmonic centrality [15]

$$c_H(e) = c_H(i,j) = \sum_{n \in V \setminus \{i,j\}} \frac{1}{\min\left(\mu(i,n), \mu(j,n)\right)}$$

This centrality measure is applicable to unconnected graphs and that is the reason why it was selected to be used in this work. In fact, given that during the algorithm some edges of the network should be avoided, there may be situations in which the graph becomes unconnected and in that case, when calculating $c_H(i,j)$ there may be node pairs $i-n$ (and consequently $j-n$) for which no path may be established. In that case, $\min\left(\mu(i,n), \mu(j,n)\right)$ is infinite and the contribution of such term in the summation to calculate $c_H(i,j)$ is 0.

This centrality measure is transformed to a cost $\mathscr{C}(e) = -c_H(e) + \max_{\mathcal{E} \in E} c_H(\mathcal{E}) + 1$.

Later on the heuristic, the execution of Prim's algorithm [14] to find the least cost spanning tree (line 13) is performed considering a cost for each edge given by

$$\mathscr{C}_{\text{Prim}}(e) = (\mathscr{C}(e) + \ln(\#e + 1))\ell(e)$$

where $\#e$ is the number of times edge e has already appeared in the obtained spines. This is the penalty mentioned in line 13, as the edges appearing more often in the solutions will have a higher cost. This should allow for a greater diversification of the obtained spines and is an improvement over the previous version of this heuristic. Also notice that the length of the edge appears in this cost, much as in the expression for the cost of upgrading the edges, which allows to focus on shorter edges.

Some degree of uncertainty is introduced to diversify the obtained spanning trees. This is achieved by considering the number of times a certain edge should be avoided, as mentioned in line 8, to be a random value (from a uniform distribution), with maximum value $maxIter$. The number of different seeds considered in line 1 is $totalSeeds$.

Along the heuristic, spines which are not feasible will also be taken into account. Note that at the end of the algorithm (line 24), a pruning of the obtained tree is performed and the final set of edges may be part of a feasible solution. To perform this pruning, the edges of the resulting spine terminating in a leaf of the tree will be identified. Considering those edges, the ones with higher centrality cost, up to a maximum of $maxEdges$ edges will be removed. This way, a central substructure is kept. The edges in this substructure will be provided to the exact problem ($x_{ij} = 1$ following the notation in [4]) and a final solution will be obtained. This final solution will be feasible, as imposed by the constraints of the LP problem. In theory this set of edges could lead to an infeasible solution but this situation never occurred in our experiments.

Obviously, the larger the number of edges provided to the LP problem, the faster the solution is found. This is especially important for larger networks, for which the exact problem may take up to days to execute. In that situation, it is beneficial to provide the LP problem with a large number of edges (i.e. not many edges should be pruned and a value for $maxEdges$ should be tuned) and the ones to be pruned should be carefully selected.

4 Results

The conditions of the experimentation were already described in Section 3.1. Table 3 shows the results of the algorithmic approach where the heuristic was used to get the set of edges that should be in the spine and the exact problem was solved to obtain the remaining edges of the spine and the final availability values for all the edges of the spine. The set of edges is an input to the exact problem, that will find the remaining edges of the spine and the final availability values, so as to reach the specified target availability level $\widehat{a_{WP}}$. No results for polska with the target $\widehat{a_{WP}} = 0.995$ are presented, as a minimal availability with this value may be achieved for all WPs with certain spines, without the upgrade of edges incurring a positive upgrade cost.

The results in the table are the ones obtained with the heuristic for $totalSeeds = 10$, and the value of $maxIter$ varied between 1 and 20. As the networks are not very large, the value of $maxEdges$ was set to $|V| - 1$, which means that the value of $maxEdges$ actually

Table 3: Results when the heuristic was used to obtain a set of edges that should belong to the spine, followed by the exact resolution approach to calculate the remaining edges and the availability upgrade cost to achieve the desired target availability level

Target availability level $\widehat{a_{WP}}$	Network	$maxIter$	Upgrade cost	$A_{\min}^{WP}$	$di_{\mathcal{S}}$ [km]
0.995					
0.997	polska	1-2;10-11;16	597.53(*)	0.997003	866
0.999		3-5;12-15;17-20	2 894.94(*)	0.999000	990
0.995		all but 2;4;12	712.88(*)	0.995202	1 215
0.997	spain	1;3;7-8;10-11;14-17;20	2 170.96(*)	0.997003	1 291
0.999		4	6 348.50(*)	0.999000	1 401
0.995		2;4	159.22(*)	0.995141	962
0.997	italia	2;4	1 323.53	0.997003	962
0.999		2;4	3 888.64	0.999000	1 095

has no influence in the pruning procedure (line 24 of the algorithm).

The indicated number of maximal iterations during which an edge should be avoided are the ones for the best results. All the results are equal to the optimal one, marked with (*), or the best one considering the first 100 000 trees of minimal length (when the optimal result could not be obtained after a 48h run for the italia network).

For the italia network, the spines obtained when $\widehat{a_{WP}} = 0.995$ and 0.997 are the same (see Fig. 3a), obviously with different upgraded availability of the edges. Another interesting result is that the number of maximal iterations in the heuristic is low, which means that the edges should not be avoided for long.

A closer look at some results provided by our algorithmic approach allows to find some interesting solutions to the problem (not presented here due to lack of space). Although not optimal, these sub-optimal solutions may have some advantages over the best solution found, in terms of diameter and average WP length (in km and in hops). This observation has already been alluded to in Section 3.1, regarding the identification of an appropriate spine, in which the focus was on the italia network. For the spain network, with $\widehat{a_{WP}} = 0.999$, our algorithmic approach (with $maxIter$ different from the one in Table 3) allowed us to find a sub-optimal solution differing from the optimal one in one edge only and with an upgrade cost only 1.60% higher and total length of the tree 0.99% higher. Both have the same diameter in hops and in km; the sub-optimal solution has better values in terms of average WP length in km (1.76% inferior) and in hops (3.33% inferior), when compared to the optimal solution.

All running times were about a few minutes (except the experiences for the italia network, in particular for $\widehat{a_{WP}} = 0.997; 0.999$, which took about 1 hour) in a Dell Precision 7500, Intel(R) Xeon(R) CPU X5660 (Six Core, 2.80GHz, 6.4GT/s, 12MB), with 48GB of RAM, and using CPLEX 12.5 [1] for the resolution of the exact problem.

The large running times in some situations are due to the fact that only a small subset of the edges of the spine is provided to the exact problem. If a larger number of edges or even the complete set of edges is provided, then the running times are only of the order of seconds, even for large networks. Therefore, two possibilities for further exploration are: (i) impose a maximum number of edges to be pruned from the spanning tree given by the heuristic; (ii) identify some interesting spines to be provided to the MILP program.

5 Conclusions and Further Work

In this work, we tackle the problem of devising a high availability structure in a network (a spine), to support an availability differentiation to services in communication networks. The aim is to select the edges of that structure and modify their availability if necessary, to achieve a certain availability target at minimum cost. A heuristic based on a centrality measure is used to select a set of edges that should be part of that structure, followed by the resolution of an exact problem to devise the remaining edges and the availability of each edge in the spine. Our approach was able to obtain the best spine and also enabled us to explore other sub-optimal spines, which showed the interest in including other criteria/metrics into the selection processes. These observations suggest as future work the study of the problem as a multi-criteria problem. Future work also includes the calculation of a set of different spines (rather than only a set of edges that should belong to the spine), so as to decrease the total execution time of the algorithmic approach, in particular for larger networks.

Acknowledgment

This article is based on work from COST Action CA15127 – RECODIS. ("Resilient communication services protecting end-user applications from disaster-based failures"), supported by COST (European Cooperation in Science and Technology).

References

[1] *IBM ILOG CPLEX Optimization Studio V12.5*. IBM, 2012.

[2] A. Alashaikh, T. Gomes, and D. Tipper. The spine concept for improving network availability. *Computer Networks*, 82:4–19, May 2015.

[3] Abdulaziz Alashaikh, David Tipper, and Teresa Gomes. Exploring the logical layer to support differentiated resilience classes in multilayer networks. *Annals of Telecommunications*, 73(1):63–79, Feb. 2018.

[4] Abdulaziz Alashaikh, David Tipper, and Teresa Gomes. Embedded network design to support availability differentiation. *Annals of Telecommunications*, 74(9–10):605–623, Oct. 2019.

[5] Hung-Yi Chang and Pi-Chung Wang. Upgrading service availability of optical networks: A labor force perspective. *International Journal of Communication Systems*, 31(9):e3553, 2018.

[6] D. Colle, S. De Maesschalck, C. Develder, P. Van Heuven, A. Groebbens, J. Cheyns, I. Lievens, M. Pickavet, P. Lagasse, and P. Demeester. Data-centric optical networks and their survivability. *IEEE Journal on Selected Areas in Communications*, 20(1):6–20, Jan. 2002.

[7] A. de Sousa, D. Santos, and 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, Germany, March 8-10 2017.

[8] Amaro de Sousa, Teresa Gomes, Rita Girão-Silva, and Lúcia Martins. Minimization of the network availability upgrade cost with geodiverse routing for disaster resilience. *Optical Switching and Networking*, 31:127–143, 2019.

[9] Rita Girão-Silva, Lúcia Martins, Teresa Gomes, David Tipper, and Abdulaziz Alashaikh. Heuristic approach for the design of a high availability structure. In *15th International Conference on the Design of Reliable Communication Networks (DRCN 2019)*, pages 29–36, Coimbra, Portugal, March 2019.

[10] N. Katoh, T. Ibaraki, and H. Mine. An algorithm for finding k minimum spanning trees. *SIAM Journal on Computing*, 10:247–255, 1981.

[11] R. Martínez, R. Casellas, R. Vilalta, and R. Muñoz. GMPLS/PCE-controlled multi-flow optical transponders in elastic optical networks [Invited]. *IEEE/OSA Journal of Optical Communications and Networking*, 7(11):B71–B80, Nov. 2015.

[12] S. Orlowski, R. Wessäly, M. Pióro, and A. Tomaszewski. SNDlib 1.0–Survivable Network Design library. *Networks*, 55(3):276–286, 2010. http://sndlib.zib.de.

[13] Alija Pašić, Rita Girão-Silva, Balázs Vass, Teresa Gomes, Ferenc Mogyorósi, Péter Babarczi, and János Tapolcai. FRADIR-II: An improved framework for disaster resilience. In *11th International Workshop on Resilient Networks Design and Modeling (RNDM 2019)*, Nicosia, Cyprus, Oct. 14-16 2019.

[14] R. C. Prim. Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389–1401, Nov. 1957.

[15] Yannick Rochat. Closeness centrality extended to unconnected graphs: the harmonic centrality index. In *6th Applications of Social Network Analysis Conference (ASNA 2009)*, Zürich, Switzerland, Aug. 26-28 2009.

[16] David Tipper. Resilient network design: challenges and future directions. *Telecommunication Systems*, 56(1):5–16, 2014.