

Heuristic Approach for the Design of a High Availability Structure

Rita Girão-Silva

*Dept. Electrical and Computer Engineering
University of Coimbra & INESC Coimbra
Coimbra, Portugal
rita@deec.uc.pt*

Lúcia Martins

*Dept. Electrical and Computer Engineering
University of Coimbra & INESC Coimbra
Coimbra, Portugal
lucia@deec.uc.pt*

Teresa Gomes

*Dept. Electrical and Computer Engineering
University of Coimbra & INESC Coimbra
Coimbra, Portugal
teresa@deec.uc.pt*

David Tipper

*Dept. Informatics and Networked Systems (DINS)
University of Pittsburgh
Pittsburgh, USA
dtipper@pitt.edu*

Abdulaziz Alashaikh

*DINS
University of Pittsburgh
Pittsburgh, USA
azizoozi@gmail.com*

Abstract—Communication networks are critical in today’s society and the availability of resources is a major concern for network managers. Adequate levels of availability for every demand must be provided in a network, to assure an adequate grade of service to the users. In this work, a set of edges is selected to provide a high availability path (the working path) and/or a high availability path pair (working path plus edge-disjoint backup path) for each demand. Exact formulations of the problems are presented and a heuristic solution approach considering an edge-betweenness centrality measure is put forward. Experimental results comparing this heuristic with other heuristics previously proposed, are presented.

Index Terms—availability, spine, resilience, heuristic, centrality.

I. INTRODUCTION

Services and applications carried over a communication network typically have different availability requirements. The network has to be capable of supporting these requirements for each demand. Essentially, this problem has been seen as a path selection problem where the designer tries to find a path with adequate availability for every demand [1]. A path can be accompanied by a supplementary protection scheme to achieve the required availability level. The choice of protection scheme ranges from shared protection with partially [2] or fully link disjoint [3], and dedicated protection [4]. While these schemes can create differentiated levels of availability, the resulting availability of each scheme depends essentially on the availability of the network parts and the number of hops along the path(s) of a connection [5], [6]. In addition, networks need to guarantee that all node pairs involved in critical service communications (e.g., financial transactions, emergency calls, smart grid communications, etc.) have a high

end-to-end availability. Such services require availability in the range of four to six 9’s, which might not be achieved using basic protection schemes e.g., 1+1 [7]. In this case, one can apply higher order configuration, i.e., 2+1 [8], which is known to be resource inefficient and constrained by the network density. Hence even though installing additional links [9] or improving the availability of the network parts would result in higher availability levels [10], [11], both options are typically not economically justifiable. Instead a network operator can work on improving the availability of a subset of nodes and links in the network. The authors of [12] seek to optimize the network availability by increasing the availability of a subset of physical links by shielding, but their approach does not necessarily support differentiation. Botton et al. [13] address a network design problem where a subset of edges, for a given cost, can be upgraded to be more reliable. It is shown that the overall resource efficiency may be improved by selecting a set of more reliable edges instead of edge-disjoint path pairs. In [14] we adapt an innovative technique of embedding a higher availability sub-structure into the network at the physical layer to improve the overall network availability without substantial modifications to the topology. We term the high availability sub-structure portion of the network the spine. The spine would connect those nodes with traffic needing a high level of availability and provide a basis for differentiated levels of availability. Some preliminary work regarding the use of cost functions associated with the spines was proposed in [15]. Some empirical formulas were devised and studied, but this work is still under progress as it is difficult to define appropriate cost functions that are close to real world costs.

A new approach designated FRAmework for DIaster Resilience (FRADIR) is presented in [16], combining network design, failure modeling and protection routing (GDP [17]). The aim is to improve disaster resilience of mission-critical applications. The regional failure model takes into account the availability of the edges of the physical network, where

The work of R. Girão-Silva, L. Martins and T. Gomes was partially supported by Fundação para a Ciência e a Tecnologia (FCT) under project grant UID/Multi/00308/2019 and was financially supported by FEDER Funds and National Funds through FCT under project CENTRO-01-0145-FEDER-029312.

a spine is assumed to be in place. The problem of selecting a set of edges to be upgraded at a minimum cost ensuring a required level of availability and geodiversity, as defined in [18], is tackled in [19]. The resulting high availability sub-graph can be considered as a spine, which is not required to be a spanning tree.

In this paper, we revise our previous work and reformulate the spine design problem with the objective of maximizing the average working (or active) path (WP) availability on the spine or alternatively maximizing the average connection availability of the path pair. For designing the spine, we consider a dedicated 1+1 link disjoint protection and we require that for each node pair a WP is routed over the spine and a backup path (BP) which can be routed freely on the network but shares no common link with the WP. We formulate this problem as a mixed integer linear programming problem and provide heuristics based on k -betweenness centrality measure used for spanning tree enumeration.

The work in [20] focuses on the optimization of the availability of the WP only (average and minimal). Different metrics for the enumeration of spanning trees are proposed and analyzed, with the availability of the edges determined by their length. In the present work, the tackled problems involve unweighted metrics as in [14], i.e. a fixed value a_{ON} is assigned to the availability of the edges on the spine, and a fixed value a_{OFF} is assigned to the availability of the remaining edges (not on the spine).

The study in [14] is the most closely related to the present work. However, [14] only focuses in the case of the BP having to avoid edges of the spine. In this work, we assume that the BP may use edges of the spine or not. If the BP is allowed to use edges of the spine, its availability (and consequently the availability of the path pair) should be higher. However, this requires a higher capacity of the links in the spine. Note that in this work we do not deal with the capacities of the links.

After this introductory section, we present the mathematical formulation of the optimization problems to be tackled in Section II. The heuristic solutions approaches are described in Section III. In Section IV, some experimental results are displayed and commented on. The paper ends with some conclusions and proposals for future work.

II. EXACT FORMULATION

In this section, we present the notation used and then proceed to the mathematical formulation of the problems to be tackled.

A. Notation

Sets

- \mathcal{N} is the set of physical nodes in the graph.
- \mathcal{L} is the set of physical undirected edges in the graph.
- \mathcal{D} is the set of directed links in the graph. An undirected edge with end nodes i and j may be represented as a pair of directed and opposite links $ij \in \mathcal{D}$ and $ji \in \mathcal{D}$.
- \mathcal{S} is the set of edges forming the spine.

- \mathcal{F} is the set of end-to-end demands or flows. A flow $f \equiv (s, t) \in \mathcal{F}$ is defined between nodes $s \in \mathcal{N}$ and $t \in \mathcal{N}$.

Availability

- $a(l)$ is the availability of edge $l \in \mathcal{L}$. The availability of a link $ij \in \mathcal{D}$ is represented by a_{ij} .
- $A_{WP(s,t)} = \prod_{l \in WP(s,t)} a(l)$ is the availability of the working path of flow (s, t) . If the path is defined in terms of directed links, then $A_{WP(s,t)} = \prod_{ij \in WP(s,t)} a_{ij}$. Similar expressions are used to calculate the availability of the backup path of flow (s, t) , $A_{BP(s,t)}$.
- $A_{(s,t)} = 1 - (1 - A_{WP(s,t)}) (1 - A_{BP(s,t)})$ is the availability of flow (or demand) (s, t) , assuming that $WP(s,t)$ and $BP(s,t)$ are edge-disjoint.
- We consider the case of a homogeneous spine with $a_{ij} = a_{ON}$ if $ij \in \mathcal{S}$ and $a_{ij} = a_{OFF}$ if $ij \notin \mathcal{S}$.

Performance measures related to the availability

- $A_a = \frac{1}{|\mathcal{F}|} \sum_{(s,t) \in \mathcal{F}} A_{(s,t)}$ is the average value of the availability $A_{(s,t)}$ for all the flows.
- $A_a^{WP} = \frac{1}{|\mathcal{F}|} \sum_{(s,t) \in \mathcal{F}} A_{WP(s,t)}$ is the average value of the availability for the WPs, $A_{WP(s,t)}$, of all the flows.

Other performance measures

- h_S is the average hop count of the WPs.
- di_S is the spine diameter, i.e. the hop count of the longest WP.

Variables to be used in the exact formulation of the problem

- z_{ij} is a binary variable, which is 1 if the link $ij \in \mathcal{D}$ is in the spine and 0 otherwise.
- $x_{ij}^{st} (y_{ij}^{st})$ is a binary variable, which is 1 if the link $ij \in \mathcal{D}$ is in the WP (BP) of the flow $(s, t) \in \mathcal{F}$ and 0 otherwise.

B. Problem 1: Maximization of the sum of availabilities of the WPs of all the flows

In problem 1, the objective function is the maximization of the sum of availabilities of the WPs of all the flows, which is equivalent to the maximization of the average value of the availability for the WPs of all the flows.

The problem is formulated as in [20], where more details may be found. For the sake of completeness, we include the formulation here:

$$\max \sum_{(s,t) \in \mathcal{F}} A_{WP(s,t)} \quad (1)$$

subject to

$$\sum_{hj \in \mathcal{D}} x_{hj}^{st} - \sum_{ih \in \mathcal{D}} x_{ih}^{st} = \begin{cases} 1 & \text{if } h = s \\ -1 & \text{if } h = t \\ 0 & \text{otherwise} \end{cases} \quad \forall h \in \mathcal{N}, (s, t) \in \mathcal{F} \quad (2)$$

$$\sum_{hj \in \mathcal{D}} y_{hj}^{st} - \sum_{ih \in \mathcal{D}} y_{ih}^{st} = \begin{cases} 1 & \text{if } h = s \\ -1 & \text{if } h = t \\ 0 & \text{otherwise} \end{cases} \quad \forall h \in \mathcal{N}, (s, t) \in \mathcal{F} \quad (3)$$

$$x_{ij}^{st} + x_{ji}^{st} \leq 1 \quad \forall ij \in \mathcal{D} \text{ with } i < j, (s, t) \in \mathcal{F} \quad (4)$$

$$y_{ij}^{st} + y_{ji}^{st} \leq 1 \quad \forall ij \in \mathcal{D} \text{ with } i < j, (s, t) \in \mathcal{F} \quad (5)$$

$$\sum_{hj \in \mathcal{D}} x_{hj}^{st} + \sum_{ih \in \mathcal{D}} x_{ih}^{st} \leq 2 \quad \forall h \in \mathcal{N}, (s, t) \in \mathcal{F} \quad (6)$$

$$\sum_{hj \in \mathcal{D}} y_{hj}^{st} + \sum_{ih \in \mathcal{D}} y_{ih}^{st} \leq 2 \quad \forall h \in \mathcal{N}, (s, t) \in \mathcal{F} \quad (7)$$

$$x_{ij}^{st} + y_{ij}^{st} \leq 1 \quad \forall ij \in \mathcal{D}, (s, t) \in \mathcal{F} \quad (8)$$

$$x_{ij}^{st} + y_{ji}^{st} \leq 1 \quad \forall ij \in \mathcal{D}, (s, t) \in \mathcal{F} \quad (9)$$

$$z_{ij} \geq x_{ij}^{st} \quad \forall ij \in \mathcal{D}, (s, t) \in \mathcal{F} \quad (10)$$

$$z_{ij} = z_{ji} \quad \forall ij \in \mathcal{D} \text{ with } i < j \quad (11)$$

$$\sum_{ij \in \mathcal{D}, i < j} z_{ij} \leq |\mathcal{N}| - 1 \quad (12)$$

$$A_{WP(s,t)} = \prod_{ij \in WP(s,t)} a_{ij} \quad \forall (s, t) \in \mathcal{F} \quad (13)$$

$$x_{ij}^{st}, y_{ij}^{st}, z_{ij} \text{ binary} \quad (14)$$

Constraint (13) is used to calculate the availability of the WP of each flow. This constraint has to be linearized, which is accomplished by applying logarithms to both sides of the equality. Considering the parameter $LA_{WP(s,t)} = -\log(A_{WP(s,t)})$, the problem may now be formulated as:

$$\min \sum_{(s,t) \in \mathcal{F}} LA_{WP(s,t)} \quad (15)$$

subject to

$$LA_{WP(s,t)} + \sum_{ij \in \mathcal{D}} x_{ij}^{st} \log(a_{ij}) = 0 \quad \forall (s, t) \in \mathcal{F} \quad (16)$$

and constraints (2)-(12), (14).

Note that since the availability of the edges on the spine are fixed (a_{ON}), constraint (16) becomes $LA_{WP(s,t)} + \log(a_{ON}) \sum_{ij \in \mathcal{D}} x_{ij}^{st} = 0, \forall (s, t) \in \mathcal{F}$.

For this same reason, it is also evident that finding the most available WP for each demand is equivalent to finding the WP with the smallest number of hops for each demand. Therefore, we are indeed tackling a problem of finding a spanning tree with the least number of hops for the paths in that spanning tree:

$$\min \sum_{(s,t) \in \mathcal{F}} \sum_{ij \in \mathcal{D}} x_{ij}^{st} \quad (17)$$

subject to constraints such that the paths in the spine are the WP for the demands and that an edge-disjoint BP for each demand must be found in the network.

Once this problem is solved, we have a feasible spine, i.e. a spanning tree containing all the WPs for all the demands and for which an edge-disjoint BP can be found for each demand. Next, we apply the Dijkstra algorithm to find the most available BP for each demand. The BP must be edge-disjoint of the corresponding WP and when it is being found, one of two possibilities may occur: i) the BP may use edges on the spine; ii) the BP should avoid edges on the spine.

C. Problem 2: Maximization of the sum of availabilities of all the flows

In problem 2, the objective function is the maximization of the sum of availabilities of all the flows, which is equivalent to the maximization of the average value of the availability for

all the flows. In this case, we have to calculate the availability for each flow, which is

$$A_{(s,t)} = 1 - (1 - A_{WP(s,t)}) (1 - A_{BP(s,t)}) \quad \forall (s, t) \in \mathcal{F} \quad (18)$$

as the WP and the BP are edge-disjoint.

1) *BP on the spine*: Assuming that the BP for the different flows may use edges of the spine (obviously as long as those edges are not in the corresponding WP), the problem is formulated as:

$$\max \sum_{(s,t) \in \mathcal{F}} A_{(s,t)} \quad (19)$$

subject to

$$A_{BP(s,t)} = \prod_{ij \in BP(s,t)} a_{ij} \quad \forall (s, t) \in \mathcal{F} \quad (20)$$

and constraints (2)-(14), (18).

Expressions (13), (20) and (18) present non-linearities that must be dealt with. The constraints (13) and (20) calculate the availability of the active path or the backup path (respectively) of each flow. These constraints have to be linearized, which is accomplished by a common approximation¹:

$$\begin{aligned} A_{WP(s,t)} &= \prod_{ij \in \mathcal{D}} (1 - x_{ij}^{st} (1 - a_{ij})) \approx 1 - \sum_{ij \in \mathcal{D}} x_{ij}^{st} (1 - a_{ij}) = \\ &= 1 - \sum_{ij \in \mathcal{D}} x_{ij}^{st} (1 - a_{ON}) \quad \forall (s, t) \in \mathcal{F} \quad (21) \end{aligned}$$

$$\begin{aligned} A_{BP(s,t)} &= \prod_{ij \in \mathcal{D}} (1 - y_{ij}^{st} (1 - a_{ij})) \approx 1 - \sum_{ij \in \mathcal{D}} y_{ij}^{st} (1 - a_{ij}) = \\ &= 1 - \sum_{ij \in \mathcal{D}} y_{ij}^{st} (1 - (a_{ON} z_{ij} + a_{OFF} (1 - z_{ij}))) \quad \forall (s, t) \in \mathcal{F} \quad (22) \end{aligned}$$

Note that the availability of each edge is not intrinsic to the edge itself, but rather dependent on whether the edge belongs to the spine or not. In expression (21), the value a_{ij} is simply replaced by a_{ON} , as all the edges in the WP belong to the spine. In expression (22), the value of a_{ij} is calculated as $a_{ON} z_{ij} + a_{OFF} (1 - z_{ij})$, which introduces the product of two binary variables, y_{ij}^{st} and z_{ij} .

Let the product of the two binary variables be represented by another binary variable: $w_{ij}^{st} = y_{ij}^{st} z_{ij}$. The usual linearization yields

$$w_{ij}^{st} \leq y_{ij}^{st} \quad (23)$$

$$w_{ij}^{st} \leq z_{ij} \quad (24)$$

$$w_{ij}^{st} \geq y_{ij}^{st} + z_{ij} - 1 \quad (25)$$

$$w_{ij}^{st} \text{ binary} \quad (26)$$

Note that equations (23)-(26) do not constitute an approximation. These are an exact approach to the linearization of the product of two binary variables.

¹This approximation is achieved by applying logarithms to both sides of the equality and then approximating the logarithmic function by the first terms of an expansion in Taylor series. It is a known result, that states that the unavailability of a series of components may be approximated by the sum of the unavailabilities of the components [21].

Therefore, the approximation in equation (22) can be rewritten as

$$A_{BP(s,t)} \approx 1 - \sum_{ij \in \mathcal{D}} (y_{ij}^{st} (1 - a_{\text{OFF}}) + w_{ij}^{st} (a_{\text{OFF}} - a_{\text{ON}})) \quad \forall (s,t) \in \mathcal{F} \quad (27)$$

along with equations (23)-(26).

The constraint (18) may also be approximated, given (21) and (27):

$$A_{(s,t)} \approx 1 - (1 - a_{\text{ON}}) (1 - a_{\text{OFF}}) \sum_{ij \in \mathcal{D}} \sum_{\substack{kl \in \mathcal{D} \\ kl \neq ij \\ kl \neq ji}} r_{ijkl}^{st} - (1 - a_{\text{ON}}) (a_{\text{OFF}} - a_{\text{ON}}) \sum_{ij \in \mathcal{D}} \sum_{\substack{kl \in \mathcal{D} \\ kl \neq ij \\ kl \neq ji}} v_{ijkl}^{st} \quad \forall (s,t) \in \mathcal{F} \quad (28)$$

where $r_{ijkl}^{st} = x_{ij}^{st} y_{kl}^{st}$ and $v_{ijkl}^{st} = x_{ij}^{st} w_{kl}^{st}$ and satisfying the following constraints $\forall (s,t) \in \mathcal{F}, ij \in \mathcal{D}, kl \in \mathcal{D}, kl \neq ij, kl \neq ji$:

$$r_{ijkl}^{st} \leq x_{ij}^{st} \quad (29)$$

$$r_{ijkl}^{st} \leq y_{kl}^{st} \quad (30)$$

$$r_{ijkl}^{st} \geq x_{ij}^{st} + y_{kl}^{st} - 1 \quad (31)$$

$$v_{ijkl}^{st} \leq x_{ij}^{st} \quad (32)$$

$$v_{ijkl}^{st} \leq w_{kl}^{st} \quad (33)$$

$$v_{ijkl}^{st} \geq x_{ij}^{st} + w_{kl}^{st} - 1 \quad (34)$$

$$r_{ijkl}^{st}, v_{ijkl}^{st} \text{ binary} \quad (35)$$

As previously noted, equations (29)-(35) do not constitute an approximation. Thus, the problem is formulated as:

$$\max \sum_{(s,t) \in \mathcal{F}} A_{(s,t)} \quad (36)$$

subject to constraints (2)-(12), (14), (23)-(26) and (28)-(35).

2) *BP off the spine*: We now formulate the corresponding problem assuming that the BP for the different flows should avoid edges of the spine (obviously the edges in the corresponding WP remain forbidden). The objective function is

$$\begin{aligned} \max \sum_{(s,t) \in \mathcal{F}} A_{(s,t)} - M \sum_{ij \in \mathcal{D}} \sum_{(s,t) \in \mathcal{F}} y_{ij}^{st} z_{ij} &\equiv \\ \equiv \max \sum_{(s,t) \in \mathcal{F}} \left(A_{(s,t)} - M \sum_{ij \in \mathcal{D}} y_{ij}^{st} z_{ij} \right) &\quad (37) \end{aligned}$$

with a high value for M . As the function is to be maximized, then the term $M \sum_{ij \in \mathcal{D}} \sum_{(s,t) \in \mathcal{F}} y_{ij}^{st} z_{ij}$ should be as small as possible, which is accomplished if y_{ij}^{st} and z_{ij} are not simultaneously 1, i.e. we should avoid having the link $ij \in \mathcal{D}$ in the spine and in a backup path simultaneously.

The problem is formulated as:

$$\max \sum_{(s,t) \in \mathcal{F}} \left(A_{(s,t)} - M \sum_{ij \in \mathcal{D}} w_{ij}^{st} \right) \quad (38)$$

subject to constraints (2)-(12), (14), (23)-(26) and (28)-(35).

III. HEURISTIC FORMULATION

In an unweighted graph, the distance between the edge's end nodes is disregarded in all the calculations. In this situation, the length of each edge is simply one and the length of a path is simply the number of hops of that path. This is used in the enumeration of shortest paths for each node pair (necessary for the calculation of the metric described in this section). In this work, we reserve the term 'edge cost' to the value assigned to each edge during the calculation of spanning trees, and the term 'edge length' to the value assigned to each edge during the calculation of shortest paths.

The WP for each demand (source-destination pair) is given by edges in the spanning tree. Only the spanning trees for which a BP (which is edge-disjoint of the corresponding WP) may be found for every node pair are admissible. We discard the spines for which there is at least one demand without an edge-disjoint BP.

Considering the information on the spine and the path pair (WP and BP) for each node pair, then the performance measures (maximal availability, for instance) may be calculated. Notice that in an unweighted graph the edges availability are assumed to be known in advance. In these experiments, the assigned value for the availability of each edge is a_{OFF} for edges off the spine and a_{ON} for edges on the spine.

A. Cost metric used for the spanning tree calculation

The considered cost metric is equivalent to metric $\{C\}$ defined in [20]. However, that metric was defined in the context of a weighted graph.

Let $\mathcal{P}_k(s,t)$ be the set of paths between nodes s and t , whose length is not higher than the length of the shortest path (L_0) plus k . As the length of a path in an unweighted graph is always an integer number (it is the number of hops of the path), then the parameter k used here is integer-valued.

The difference in length between successive paths found in the path enumeration algorithm is always a non-negative integer. By including in set $\mathcal{P}_k(s,t)$ all the paths whose length is L such that $L_0 \leq L \leq L_0 + k$, we are in fact considering the possibility of having paths with a maximum of $k+1$ different length values. However, note that we cannot guarantee having an exact total of $k+1$ different path length values in $\mathcal{P}_k(s,t)$. As an example, assume that we find a set of paths with length L_0 , followed by a set of paths with length $L_1 = L_0 + k$. In this case we only include in $\mathcal{P}_k(s,t)$ paths with 2 different length values L_0, L_1 .

Given the set $\mathcal{P}_k(s,t)$, we may define $\sigma_k(s,t) = |\mathcal{P}_k(s,t)|$ and $\sigma_k(s,t|l)$, which is the number of paths in the set $\mathcal{P}_k(s,t)$ that include edge l .

A topological structural measure defined in [22] is the k -betweenness centrality for edge l , given by

$$\mathcal{B}_k(l) = \sum_{s,t \in \mathcal{N}, s \neq t} \frac{\sigma_k(s,t|l)}{\sigma_k(s,t)} \quad (39)$$

Considering the k -betweenness centrality for each edge, the cost of edge l is defined as $c_k(l) = -\mathcal{B}_k(l) + \max_{\ell \in \mathcal{L}} \mathcal{B}_k(\ell) + 1$.

B. Proposal of heuristic considering edges to be avoided

The heuristic is based on considering some edges that should be avoided, in order to identify different possible spines.

Input: $k, maxIter$.

Output: Feasible spines from which the one with best A_a^{WP} and the one with best A_a may be identified.

for $listReset = 0$ **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 $|\mathcal{L}|$ 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 added first).

repeat

Calculate spine that tries to avoid edges on the list and minimizes the total centrality cost (using Prim's algorithm [23]).

if spine is feasible **then**

Calculate BPs for every demand.

Calculate performance measures.

else

Identify a demand $s - t$ for which no disjoint BP exists.

Considering s and t , select the node with higher average centrality.

The edge (or one of the edges) in the spine passing in this node should be avoided for $maxIter$ iterations.

Update list of edges to be avoided: decrease the number of iterations during which they should be avoided.

end if

until a feasible spine is found.

end loop

end for

The parameter $maxIter$ indicates the maximum number of iterations during which an edge should be avoided. The inner loop of the heuristic is run a total of $|\mathcal{L}| + 1$ times, as it is run first without any prior edges to be avoided and then with each one of the edges in the network to be avoided. The edges are selected in decreasing order of centrality cost.

During each run, spines are calculated using $c_k(l)$ as the cost of each edge l . Once a feasible spine is found, each run halts. After the $|\mathcal{L}| + 1$ runs, the solution leading to the best value for one of the performance measures is identified.

In this procedure, a value of average centrality cost for a node is considered. For a node $n \in \mathcal{N}$, the average centrality cost is simply the average value of $c_k(l)$, considering all the edges l leaving or entering n . A higher average centrality cost corresponds to a lower centrality. This calculation is related to the node betweenness centrality.

As for the BP calculation for each node pair, the Dijkstra's algorithm is used. For each node pair, a modified network is devised, where the edges forming the WP are removed from the original network. In this modified network, the most available path is obtained. As mentioned previously, we assume that the BP may use edges on the spine or not.

A final note on the number of times the inner loop is run: the first time, it is run without any prior edges to be avoided; the following times, it is run with each of the $|\mathcal{L}|$ edges as an edge to be avoided. Even if some of the edges do not actually need to be avoided (because of their high centrality cost, they will most likely never be part of the spine), this option was taken to try to diversify the spines to be obtained.

C. Other heuristics used for comparison

Two other heuristics were used for comparison with the results of the proposed heuristic.

1) *Heuristic proposed in [14]*: Structural properties of a network topology to select a suitable spine are explored in these heuristics. The edge cost is a weighted sum of the edge degree and the edge betweenness, where the weights, summing one, are considered in 0.1 steps. A minimum cost spanning tree is calculated which is only admissible if an edge disjoint BP can be obtained for every node pair. When that is not verified, the set of conflicting edges (those shared by a WP and the maximally link disjoint corresponding BP) are identified, and a new spanning tree is obtained avoiding each of those links (and all possible combination of them). Finally among the admissible spanning trees the one with best performance according to the relevant metric is saved. Recall that results published in [14] focus only on the situation in which the BP should avoid edges of the spine, whereas we will present new results obtained with this heuristic for the case of the BP being allowed to use edges of the spine.

2) *Heuristic proposed in [20]*: This heuristic is based on the enumeration of candidate spines (spanning trees) by increasing order of a cost metric, which in this case is $c_k(l)$ described in III-A. The spanning tree enumeration is performed by the algorithm in [24] and only a maximum number of $|\mathcal{N}| \cdot |\mathcal{L}|$ trees was calculated. The WP for each demand (source-destination pair) is given by edges in the spanning tree. The BP is selected as the most available among the paths which are edge-disjoint of the corresponding WP. It is also possible to indicate whether the BP may use edges on the spine or if edges on the spine should be avoided (by assigning them a length much higher than one).

Note that the results published in [20] focus only on performance measures related to the availability of the WP, whereas we will present results of this heuristic considering performance measures related to the availability of the path pairs and taking into account whether the BP is allowed to use edges of the spine or not.

IV. EXPERIMENTAL RESULTS

Experiments were conducted with the reference networks of polska [25] and italia (obtained from ITNet in [26, Fig.5])

after removing the edge GEN \leftrightarrow SAV). The topology features of these networks are described in Table I.

Exact results were obtained by enumerating all the possible trees (which is possible for the smaller networks) or by solving the formulated problems with CPLEX 12.5 [27]. In this case, the spines for which the objective functions are optimized (subject to the indicated constraints) are found. Note that no distinction is made between alternative optimal solutions. We simply consider the solution provided by the exact solver or by the heuristics, even if there may be alternative optimal solutions.

Experiments with the cost metric $c_k(l)$ described in III-A were performed, for $k = 0, 1, 2$. An increase in k means that a larger number of shortest paths is taken into consideration in the calculation of the cost metric. According to the conclusions in [20], the value of k does not have to be very high, and it is enough to find the shortest paths and the paths with length close to these ones.

In tables II-III, information on the values of the availability measures obtained in the experiments considering the exact formulation and the heuristics is provided. Other parameters presented in the tables include the average number of hops in the spine and the spine diameter. For each method, the first row refers to the spine maximizing $A_a^{W_P}$ (problem 1) and the second row refers to the spine maximizing A_a (problem 2). In the availability values, we present at least 3 significant digits after the nines.

The optimal value is displayed in bold for the corresponding problem. A lexicographic approach was not considered, in the sense that given different trees with the same value of an availability measure, no further selection was performed among them. For instance, we may have a spine leading to a certain value of $A_a^{W_P}$ and different possible solutions for the BPs which will lead to different solutions for the value of A_a .

In terms of execution times of the exact algorithmic approach, the resolution of problem 1 takes a few seconds for polska, but the resolution of problem 2 takes over a day for the same network. As for the italia network, even though it is a medium size network, the exact approach takes a few days for problem 1 and for problem 2, no results were available even after a few days of execution. This is the greatest shortcoming of the exact resolution approach.

Another shortcoming is the fact that we are using an approximation for $A_{(s,t)}$ in problem 2. Some details on the associated approximation error are provided in [21]. This approximation error is of the same order of magnitude of the default tolerance values used in CPLEX. Therefore, even if we establish more stringent tolerance values to try to obtain more

TABLE I: Network characteristics ($|\mathcal{N}|$, $|\mathcal{L}|$, μ – average node degree, δ – diameter)

Network	$ \mathcal{N} $	$ \mathcal{L} $	μ	$ \mathcal{N} \cdot \mathcal{L} $	δ
polska	12	18	3.00	216	4
italia	32	69	4.31	2208	6

TABLE II: Results for the polska network

(a) BP may use edges of the spine

Method	k	max $Iter$	A_a	$A_a^{W_P}$	h_S	di_S
Exact			0.9999487	0.99734	2.6667	5
			0.9999566	0.99660	3.4091	8
Heuristic in [14]			0.9999347	0.99734	2.6667	5
			0.9999560	0.99658	3.4242	8
Heuristic in [20]	2		0.9999488	0.99734	2.6667	5
	1,2		0.9999534	0.99720	2.8030	6
Proposed heuristic	1,2	≥ 3	0.9999488	0.99734	2.6767	5
	0		0.9999531	0.99728	2.7272	5

(b) BP should avoid edges of the spine

Method	k	max $Iter$	A_a	$A_a^{W_P}$	h_S	di_S
Exact			0.9999289	0.99734	2.6667	5
			0.9999480	0.99660	3.4091	8
Heuristic in [14]			0.9999294	0.99734	2.6667	5
			0.9999480	0.99660	3.4091	8
Heuristic in [20]	2		0.9999322	0.99734	2.6667	5
	0		0.9999480	0.99660	3.4091	8
Proposed heuristic	2	2	0.9999322	0.99734	2.6667	5
	0		0.9999480	0.99660	3.4091	8

TABLE III: Results for the italia network

(a) BP may use edges of the spine

Method	k	max $Iter$	A_a	$A_a^{W_P}$	h_S	di_S
Exact			0.9999137	0.99643	3.5726	6
			Results not available for problem 2			
Heuristic in [14]			0.9998982	0.99616	3.8468	8
			0.9999003	0.99545	4.5605	10
Heuristic in [20]	1		0.9999164	0.99620	3.8085	8
	0		0.9999180	0.99598	4.0262	9
Proposed heuristic	1	5	0.9999139	0.99619	3.8185	7
	0		0.9999194	0.99600	4.0040	9

(b) BP should avoid edges of the spine

Method	k	max $Iter$	A_a	$A_a^{W_P}$	h_S	di_S
Exact			0.9998297	0.99643	3.5726	6
			Results not available for problem 2			
Heuristic in [14]			0.9998015	0.99623	3.7742	8
			0.9998115	0.99348	6.5423	15
Heuristic in [20]	1		0.9997968	0.99620	3.8085	8
	2		0.9998394	0.99600	4.0121	9
Proposed heuristic	1	17,18	0.9998146	0.99619	3.8185	7
			0.9998392	0.99567	4.3427	10

accurate results, the approximation error of $A_{(s,t)}$ persists. Due to this situation, the exact results for problem 2 displayed in Table II are the ones obtained from the enumeration of all the possible trees (only for smaller networks).

The heuristic in [14] requires only a few seconds (less than 3) for the polska network and around 20 seconds for the italia network. Still in terms of execution times, the other heuristics take a short time (under 1 sec) for the smaller networks. However, for a medium size network such as the italia network, the heuristic in [20] takes over 10 min, while the heuristic proposed here takes about 1 sec.

Considering the results, the spines leading to the best solutions in terms of the A_a measure have a higher diameter (and the average number of hops of the WPs are obviously higher). This means that the spine does not include mainly the central edges, as observed when the best solution in terms of the A_a^{WP} measure is found. We realize that the maximization of the WP availability is not closely related to the maximization of the path pair availability, as is shown in [14]. In fact, if the average number of hops of the WPs is higher, then the average availability of the WPs is lower. This observation justifies the need to study the problems regarding both parameters and not just A_a^{WP} .

In Figure 1, examples of spines are displayed for the italia network. In Figure 1a, the exact solution that minimizes A_a^{WP} is displayed. Note that the solution mainly includes more central edges, as expected, as this type of solution tends to lead to a set of shorter (in hop count) and more available WPs in the spine. The minimal cost admissible solutions obtained considering the edge cost $c_k(l)$ with $k = 0, 2$ are displayed in Figures 1b-1c. These solutions present some differences to the solution in Figure 1a, mainly the fact that some very long and apparently not central edges are used. Although in topological terms, these edges do not seem to be very central, they are used in many min-hop paths, and so they will have a low centrality cost, given the definition of edge cost $c_k(l)$.

In the extensive research we performed with the heuristics proposed here, we noticed that the best values of A_a^{WP} were usually obtained with $k = 2$, whereas the best values of A_a were usually obtained with $k = 0$. This has to do with the nature of the centrality measure considered in this study. By considering longer paths in the calculation of $B_k(l)$ the measure of centrality of the edges becomes more accurate, as can be seen in Figures 1b-1c. By identifying the more central edges, the obtained spines tend to focus on those edges, originating trees with smaller diameter.

A comparison of the results of the heuristics for the polska network does not show remarkable differences. When the BP is expected to avoid edges of the spine, all the heuristics find the optimal solutions. When the BP is allowed to use edges of the spine, the heuristic in [14] finds a better solution for A_a , but at the expense of longer WPs. Therefore, a trade-off exists and it would be necessary to establish whether longer WPs or a higher average availability for the path pair was more desirable.

For the italia network, the heuristic in [14] is not so effective in finding the best solutions, although it manages to find the best solution for A_a^{WP} with the BP avoiding edges of the spine. In this situation, the spanning tree enumeration heuristic [20] finds better results than the heuristic proposed here, but with a much longer execution time (as mentioned earlier). When the BP is allowed to use edges of the spine, the heuristic proposed here finds the best solution in terms of A_a , without incurring in longer WPs.

A final note on the results for the italia network is that none of the heuristics managed to find a solution with the same value of A_a^{WP} that the exact resolution approach provided.

However, it should be remarked that the italia network has a total number of spanning trees of the order of 10^{14} and it is very difficult for any of the heuristics to explore a total of solutions close to that number. For instance, the heuristic [20] only enumerated $|\mathcal{N}| \cdot |\mathcal{L}| = 2208$ trees (see Table I).

V. CONCLUSIONS

This work focuses on the design of a high availability structure (a spanning tree termed the spine) according to two different criteria, the maximization of the average availability of the WPs and the maximization of the average availability of the path pairs. The mathematical formulation of these problems is presented.

As the exact resolution of the problems is not always possible (due to excessive running time and/or due to the necessary approximations that must be used to linearize the availability calculation), heuristics were considered for the resolution of the problem. Two heuristics already proposed were used and a third novel heuristic was put forward. This heuristic relies on the spanning tree calculation when some of the edges are to be avoided.

A k -betweenness centrality measure for the edges was used to define the cost metric to be employed in two of the heuristics, the novel one and the one based on tree enumeration. The results show the adequacy of the heuristics, in particular the ones based on the defined centrality measure, as they lead to shorter WPs in the network.

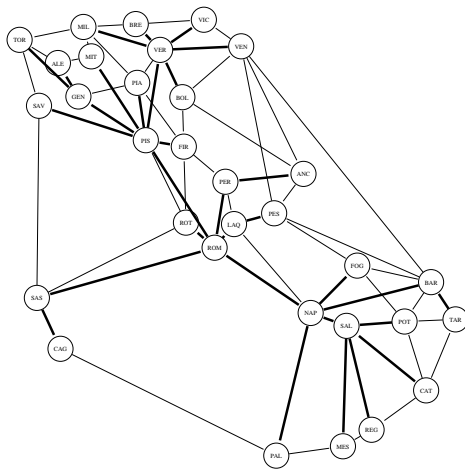
Regarding further work, we feel that other performance measures related to the minimal value of availability for the WPs or for the path pairs should be explored. We also plan on turning the proposed heuristic into a tabu search procedure, where some degree of randomness is introduced, to further diversify the obtained results. Finally, some information on traffic demands and capacities for the edges should be considered, in an effort to make this problem closer to reality.

ACKNOWLEDGMENT

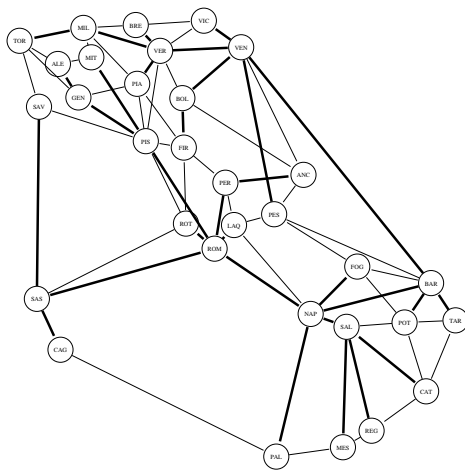
This article is based upon work from COST Action CA15127 – RECODIS.

REFERENCES

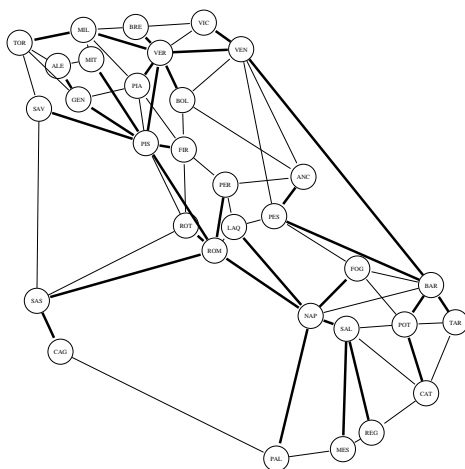
- [1] D. Schupke, "Guaranteeing service availability in optical network design," *Proc. SPIE 6022, Network Architectures, Management, and Applications*, vol. 6022, 2005.
- [2] S. Yang, S. Trajanovski, and F. Kuipers, "Availability-Based Path Selection," in *6th International Workshop on Reliable Networks Design and Modeling (RNDM2014)*, Barcelona, Spain, 2014, pp. 17–19.
- [3] L. Zhou, M. Held, A. Member, and U. Sennhauser, "Connection Availability Analysis of Shared Backup Path-Protected Mesh Networks," *Journal of Lightwave Technology*, vol. 25, no. 5, pp. 1111–1119, 2007.
- [4] M. Tornatore, D. Lucerna, B. Mukherjee, and A. Pattavina, "Multilayer Protection with Availability Guarantees in Optical WDM Networks," *Journal of Network and Systems Management*, vol. 20, no. 1, pp. 34–55, Sep. 2012.
- [5] D. Mello, D. Schupke, M. Scheffel, and H. Waldman, "Availability maps for connections in WDM optical networks," in *DRCN 2005. 5th International Workshop on Design of Reliable Communication Networks - "Reliable Networks for Reliable Services"*. IEEE, 2005, pp. 77–84.
- [6] J. Segovia, E. Calle, P. Vila, J. Marzo, and J. Tapolcai, "Topology-focused availability analysis of basic protection schemes in optical transport networks," *Journal of Optical Networking*, vol. 7, no. 4, pp. 351–364, 2008.



(a) Exact solution for problem 1



(b) First admissible spine for $k = 0$



(c) Solution that minimizes the total centrality cost, calculated for $k = 2$ (which is also the first admissible spine)

Fig. 1: Examples of spines for the italia network

[7] D. A. Garbin and J. E. Knepley, "Design and analysis of high availability networks," in *2009 IEEE Conference on Technologies for Homeland Security*, May 2009, pp. 1–6.

[8] J. Ahmed and C. Cavdar, "Hybrid survivability schemes achieving high connection availability with a reduced amount of backup resources," *Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. 152–161, 2013.

[9] J. M. Gutierrez, K. Katrinis, K. Georgakilas, A. Tzanakaki, and O. B. Madsen, "Increasing the cost-constrained availability of WDM networks with degree-3 structured topologies," in *2010 12th International Conference on Transparent Optical Networks*, June 2010, pp. 1–4.

[10] B. Basch, M. Mezhoudi, C. K. Chu, and R. Goudreault, "Maintenance activities, operations and environment conditions impact on the availability of highly reliable optical transport networks," in *Networks 2006. 12th International Telecommunications Network Strategy and Planning Symposium*, Nov 2006, pp. 1–7.

[11] I. Rados, T. Sunaric, and P. Turalija, "Suggestions for Availability Improvement of Optical Cables," in *Circuits and Systems for Communications, 2002. Proceedings. ICCSC '02. 1st IEEE International Conference on*. IEEE, 2002, pp. 234–239.

[12] J. Zhang, E. Modiano, and D. Hay, "Enhancing network robustness via shielding," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2209–2222, Aug 2017.

[13] Q. Botton, B. Fortz, and L. Gouveia, "On the hop-constrained survivable network design problem with reliable edges," *Computers & Operations Research*, vol. 64, pp. 159–167, 2015.

[14] A. Alashaikh, T. Gomes, and D. Tipper, "The spine concept for improving network availability," *Computer Networks*, vol. 82, pp. 4–19, 2015.

[15] A. Alashaikh, D. Tipper, and T. Gomes, "Designing a high availability subnetwork to support availability differentiation," in *14th International Conference on the Design of Reliable Communication Networks (DRCN)*, Paris, France, Feb. 19 2018.

[16] A. Pašić, R. Girão-Silva, B. Vass, T. Gomes, and P. Babarçzi, "FRADIR: A novel framework for disaster resilience," in *10th International Workshop on Resilient Networks Design and Modeling (RNDM2018)*, Longyearbyen, Norway, Aug. 27-29 2018.

[17] P. Babarçzi, A. Pašić, J. Tapolcai, F. Németh, and B. Ladóczki, "Instantaneous recovery of unicast connections in transport networks: Routing versus coding," *Computer Networks*, vol. 82, pp. 68–80, 2015.

[18] 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, March 8-10 2017.

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

[20] R. Girão-Silva, L. Martins, T. Gomes, A. Alashaikh, and D. Tipper, "Improving network availability - a design perspective," in *Third International Congress on Information and Communication Technology (ICICT 2018)*, ser. Advances in Intelligent Systems and Computing, X.-S. Yang, S. Sherratt, N. Dey, and A. Joshi, Eds., vol. 797. Singapore: Springer, 2019, pp. 799–815.

[21] D. A. Schupke and F. Rambach, "A link-flow model for dedicated path protection with approximative availability constraints," *IEEE Communications Letters*, vol. 10, no. 9, pp. 679–681, 2006.

[22] K. Jiang, D. Ediger, and D. A. Bader, "Generalizing k -betweenness centrality using short paths and a parallel multithreaded implementation," in *2009 International Conference on Parallel Processing (ICPP'09)*, Sep. 2009, pp. 542–549.

[23] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technical Journal*, vol. 36, no. 6, pp. 1389–1401, Nov. 1957.

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

[25] S. Orlowski, R. Wessály, M. Pióro, and A. Tomaszewski, "SNDlib 1.0–Survivable Network Design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010, <http://sndlib.zib.de>.

[26] M. Tornatore, G. Maier, and A. Pattavina, "Availability design of optical transport networks," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 8, pp. 1520–1532, Aug 2005.

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