This is the accepted version of the following article: **Dorabella Santos, Lúcia Martins, Teresa Gomes, and Rita Girão-Silva. Near Optimal Network Design for Path Pair Availability Guarantees. *2022 12th International Workshop on Resilient Networks Design and Modeling (RNDM)*, Compiègne, France, Sep. 19-21, 2022, pp. 1-8. DOI: 10.1109/RNDM55901.2022.9927721**, which has been published in final form at `https://ieeexplore.ieee.org/document/9927721`.

Near Optimal Network Design for Path Pair Availability Guarantees*

Dorabella Santos† Lúcia Martins‡† Teresa Gomes‡† Rita Girão-Silva‡†

dorabella.santos@gmail.com {lucia, teresa, rita}@deec.uc.pt

May 23, 2022

Abstract

Guaranteeing high levels of availability in the network in a cost effective manner is of primary importance to network operators and managers. We address the network

*This work was partially funded by FCT – Fundação para a Ciência e a Tecnologia, I.P. under the project UIDB/00308/2020.

†Institute for Systems Engineering and Computers at Coimbra (INESC Coimbra), Coimbra, Portugal

‡University of Coimbra, Department of Electrical and Computer Engineering, Coimbra, Portugal

design problem for path pair availability guarantees, assuming links can be upgraded to have an increased availability. Since the path pair availability constraints are non-linear and not linearizable in an exact manner, this mathematical problem has been avoided by considering only the working path availability or availability guarantees for the working and backup paths separately in a disaggregated way. In this paper, we present an aggregated model, where only the path pair availabilities must be fulfilled. In this model, we consider a convex relaxation for an approximation of the path pair availability to obtain linear constraints, and describe an iterative approach to tighten the bounds of the solution space, in order to obtain near-optimal solutions. The results show that considering an aggregated model is more cost effective than considering a disaggregated model with explicit values for the availabilities of the working and the backup paths.

Keywords: resilient network design, availability guarantees, optimization, convex relaxation.

1 Introduction

Communication networks are extremely important in today's society, as they support many fundamental services, such as e-government services, e-banking, e-commerce, smart grid management, or remote working. Also in the event of failures or disasters, communication networks play a critical role in rescue and recovery efforts. Therefore, it is essential to guarantee the resilience of communication networks (expressed in terms of survivability and tolerance to faults, disruptions and traffic variations) and also a high dependability of the network resources (which includes reliability and availability) [8]. In particular, guaranteeing high levels of availability of the network resources, especially due to commitments in Service Level Agreements (SLAs), in a cost effective way is of primary importance for network operators and managers. Critical services may need availabilities of as much as "five-nines" (i.e., 0.99999 meaning that the service may be unavailable no more than 5 minutes and 15 seconds a year) or even of "six-nines" in some cases.

Protection strategies help increase network availability, but may not be sufficient to achieve the required levels of availability. There are different protection schemes, including shared and dedicated approaches, where the obtained availability depends on parameters such as the availability of the network elements, or the number of hops in a path. There are many protection schemes to guarantee a specified network availability, e.g. [5]. The concept of having a high availability subgraph in the network was explored in [1]. This subgraph is composed of a subset of nodes and edges of a network, and should be used to guarantee a high level of availability to the more critical services. It can also be devised in the context of disaster preparedness strategies. Note that different metrics may be considered in terms of disaster preparedness and to assess disaster recovery strategies [11, 12].

The increase of the availability of the edges entails a certain cost, and for any network operator or manager it is essential to consider how much to invest, in order to guarantee the contracted SLAs. It is difficult to assess the upgrade cost, but in [2] some cost functions are proposed to approximate actual real costs. Therefore, we assume the links of the subgraph can be upgraded at a given cost expressed by a cost function, in order to achieve the required end-to-end availability target. A cost-efficient deployment of service function chains (SFCs) with availability guarantees, is proposed in [4]. The Virtual Network Function (VNF) redundancy allocation is determined according to a metric that captures both the cost and the impact on the availability improvement of the selected VNF backup.

For each source-destination, a pair of paths is devised to guarantee $1 + 1$ protection. The working (or active or primary) path and the backup (or secondary) path must be link-disjoint, so that a failure in one of the links of the working path (WP) may be quickly overcome by using the backup path (BP). As the services should be provided with a high availability in regular conditions, it is considered that the paths used in regular conditions (i.e., the WPs) should use edges of the high availability subgraph only. However, we also want to guarantee an appropriate availability even when the BPs are used, so their availability also has to be taken into account. The availability of a path is the product of the availability of the elements that form it. Assuming that the nodes availability is much higher than the links availability, we will only consider the availability of the links in the calculation of the availability of a path. The availability of the path pair is calculated as the complementary value of the unavailability of the path pair, which is the product of the unavailabilities of the WP and the BP.

It is assumed that a BP may use edges on the high availability subgraph, if necessary, as long as it is link-disjoint to its WP. This should lead to higher values of availability of the BP, and consequently of the path pair. This will entail the need for a higher capacity of the edges in the high availability subgraph, but that aspect is outside the scope of this work. For a more efficient use of capacity, a shared backup path protection approach may be used, but the availability may not be maximized. In [17], a strategy to maximize the availability given a network capacity plan is proposed.

The aim of this work is to select the edges that will form the high availability subgraph, and their enhanced availability, such that: (i) a link-disjoint path pair exists between any two nodes in the network; (ii) any WP uses exclusively links of the high availability subgraph; (iii) any path pair has an availability of at least Λ; (iv) the edge upgrade cost is minimized. We consider that the subgraph is a spanning tree, as in [2]. This problem is known to be NP-complete and the availability constraints comprising the path pairs are nonlinear. To circumvent the nonlinearity of these constraints, it has been common in previous works, e.g. [2], to disregard the BP availability in the model. In [13] we did consider the path pair, but with disaggregated availability constraints where given minimum values for the working and backup path availabilities were considered separately, which may lead to excessively high costs. These strategies aim to obtain mixed integer linear programming (MILP) models which can then be solved with optimization solvers.

To deal with the nonlinearity of the path pair availability constraints, we consider the convex relaxation of these constraints [3]. This is a technique commonly used when solving bilinear problems [6, 7, 10]. The convex envelop is defined by estimated bounds, and the solution is closer to the optimal one (true solution) the tighter the estimated bounds are [9]. To the best of our knowledge, this technique has not been used to deal with the nonlinearity of path pair availability constraints in a network design problem. In [14] we have used this approach in the context of a Software Defined Networking (SDN) controller placement problem, to deal with intercontroller availability guarantees involving a very small set of nodes, and therefore no tree subgraph was considered.

The main contributions of this paper are:

- Comparison of the spanning tree upgrade cost for different pairs of availability target values for the WP and BP, in the disaggregated model. The values were determined to achieve the path pair target availability Λ, and they illustrate the dependency of the disaggregated model solutions on the required WP and BP availability values.

- Adaptation of an aggregated model with path pair availability guarantees between all pairs of nodes. To linearize the model, we consider the convex relaxation of these constraints, to achieve the path pair target availability without the need of suboptimal disaggregated constraints.

- Use of an algorithm to solve the aggregated model by iteratively tightening the convex envelope to obtain near-optimal solutions.

- Comparison of the solutions in terms of the spanning tree upgrade cost between the aggregated and the disaggregated models.

The paper is organized as follows: in the next section, the link availability upgrade model is presented, followed by the problem formulation in section 3. In sections 4 and 5, two different approaches for dealing with the availability constraints are explored. After presenting the resolution strategy for the aggregated model (section 6), the computational results are displayed and analysed in section 7. The paper ends with the conclusions.

2 Link Availability Upgrade Model

Consider that the network can be represented as a graph $G = (N, E)$, where N is the set of nodes in the network and E is the set of bidirectional edges or links connecting the nodes. Each link is represented by its end nodes $\{i, j\}$ and has an associated length ℓ_{ij}.

Each link has an associated availability which is dependent on its length. We consider that the default link availability is given by [16]:

$$a^0_{ij} = 1 - \frac{MTTR}{MTBF_{ij}} \tag{1}$$

where $MTTR$ denotes the mean time to repair and is assumed to be 24 h, and $MTBF_{ij}$ denotes the mean time between failures for link $\{i, j\}$, which is given by $CC \cdot 365 \cdot 24/\ell_{ij}$, where CC is the cable cut metric and is assumed to be 450 km.

We assume that each link (in the spanning tree) can be upgraded to have increased availability. We consider the incremental availability link upgrade used in [2]. Consider K levels of link upgrade, where in each level the link unavailability is decreased by a factor $0 < \varepsilon < 1$. Let a^k_{ij} denote the link availability in level $k = 1, \dots, K$ which can be expressed as:

$$a^k_{ij} = a^{k-1}_{ij} + \varepsilon(1 - a^{k-1}_{ij}) \tag{2}$$

The cost of upgrading link $\{i, j\}$ to level $k = 1, \dots, K$ is given by [2]:

$$c^k_{ij} = -\ell_{ij} \cdot \ln\left(\frac{1 - a^k_{ij}}{1 - a^0_{ij}}\right) \tag{3}$$

3 Problem Formulation

The problem we address is to devise a spanning tree in the network whose links can be upgraded, to achieve the required target availability Λ between all pairs of nodes in the network. To increase the availability, we require path protection between each node pair via a pair of link-disjoint working and backup paths. The WPs are routed over the spanning tree, and each pair of working and backup paths must guarantee an end-to-end availability of at least Λ. The objective of the optimization problem is to minimize the upgrade cost of the spanning tree.

Consider A as the set of arcs, i.e., directed links. Each arc is denoted by its end nodes (i, j) directed from i to j. Denote $V(i)$ as the set of nodes adjacent to i. Consider the following decision variables:

- z_{ij}^k binary variable that is 1 if link $\{i, j\}$ is in level k where $k = 0, 1, ..., K$, and 0 otherwise
- w_{ij} binary variable that is 1 if link $\{i, j\}$ belongs to the spanning tree, and 0 otherwise
- x_{ij}^{sd} binary variable that is 1 if arc (i, j) belongs to the WP from nodes s to d, and 0 otherwise
- y_{ij}^{sd} binary variable that is 1 if arc (i, j) belongs to the BP from nodes s to d, and 0 otherwise

The problem can be formulated as (adapted from [2]):

$$\min \sum_{k=1}^{K} c_{ij}^k z_{ij}^k \tag{4}$$

s.t.

$$\sum_{j \in V(i)} \left(x_{ij}^{sd} - x_{ji}^{sd}\right) = \begin{cases} 1 & i = s \\ -1 & i = d \\ 0 & \text{otherwise} \end{cases} \qquad s \in N, d \in N, i \in N \tag{5}$$

$$\sum_{j \in V(i)} \left(y_{ij}^{sd} - y_{ji}^{sd}\right) = \begin{cases} 1 & i = s \\ -1 & i = d \\ 0 & \text{otherwise} \end{cases} \qquad s \in N, d \in N, i \in N \tag{6}$$

$$x_{ij}^{sd} + x_{ji}^{sd} + y_{ij}^{sd} + y_{ji}^{sd} \leq 1 \qquad s \in N, d \in N, \{i, j\} \in E \tag{7}$$

$$x_{ij}^{sd} + x_{ji}^{sd} \leq w_{ij} \qquad s \in N, d \in N, \{i, j\} \in E \tag{8}$$

$$\sum_{\{i,j\} \in E} w_{ij} \leq |N| - 1 \tag{9}$$

$$\sum_{k=0}^{K} z_{ij}^k = 1 \qquad \{i,j\} \in E \tag{10}$$

$$\sum_{k=1}^{K} z_{ij}^k \leq w_{ij} \qquad \{i,j\} \in E \tag{11}$$

$$\textit{availability constraints} \tag{12}$$

$$w_{ij} \in \{0,1\} \qquad \{i,j\} \in E \tag{13}$$

$$z_{ij}^k \in \{0,1\} \qquad \{i,j\} \in E, k = 0, \ldots, K \tag{14}$$

$$x_{ij}^{sd}, y_{ij}^{sd} \in \{0,1\} \qquad s \in N, d \in N, (i,j) \in A \tag{15}$$

The objective function (4) minimizes the cost of upgrading the spanning tree links.

Constraints (5) and (6) are the flow conservation constraints for the WPs and BPs between nodes s and d, respectively. Constraints (7) guarantee that each pair of working and backup paths are link-disjoint.

Constraints (8)-(9) guarantee that the WPs are routed over a spanning tree.

Constraints (10) guarantee that each link $\{i,j\}$ is either not upgraded, i.e., $z_{ij}^0 = 1$, or is upgraded to one and only one level $k = 1, \ldots, K$. Constraints (11) guarantee that only the links of the spanning tree can be upgraded.

Constraints (12) are nonlinear in nature and will be expressed in two ways: in a disaggregated version (developed in section 4), where availability target values are considered separately for the working and backup paths respectively, leading to more costly solutions; and in an aggregated version (developed in section 5), where the path pair availability is considered via the convex relaxation to obtain linearized constraints.

Finally, constraints (13)-(15) are the variable domain constraints.

4 Disaggregated Model

We consider the disaggregated model presented in [2]. In this model, target availability values are considered separately for working and backup paths: a WP should guarantee an availability of at least Λ_{WP}, while a BP should guarantee an availability of at least Λ_{BP}. Consider the additional decision variables:

p_{ij}^{sd} real variable accounting for the unavailability of link $\{i,j\}$ if it belongs to the WP between nodes s and d

q_{ij}^{sd} real variable accounting for the unavailability of link $\{i, j\}$ if it belongs to the BP between nodes s and d

Then constraints (12) are given as in [2]:

$$p_{ij}^{sd} = \left(x_{ij}^{sd} + x_{ji}^{sd}\right) \sum_{k=0}^{K} z_{ij}^{k}(1 - a_{ij}^{k}) \qquad s \in N, d \in N, \{i, j\} \in E \tag{16}$$

$$q_{ij}^{sd} = \left(y_{ij}^{sd} + y_{ji}^{sd}\right) \sum_{k=0}^{K} z_{ij}^{k}(1 - a_{ij}^{k}) \qquad s \in N, d \in N, \{i, j\} \in E \tag{17}$$

$$1 - \sum_{\{i,j\} \in E} p_{ij}^{sd} \geq \Lambda_{WP} \qquad s \in N, d \in N \tag{18}$$

$$1 - \sum_{\{i,j\} \in E} q_{ij}^{sd} \geq \Lambda_{BP} \qquad s \in N, d \in N \tag{19}$$

Constraints (16) and (17) are the definition of variables p_{ij}^{sd} and q_{ij}^{sd}, respectively. Constraints (18) and (19) guarantee the target availability for WPs and BPs, respectively. The known result stating that the unavailability of a series of components may be approximated by the sum of the unavailabilities of the components was used [15].

Note that constraints (16) and (17) are nonlinear. However, since variables z_{ij}^{k}, x_{ij}^{sd} and y_{ij}^{sd} are binary, they can be exactly linearized using McCormick envelopes. Therefore, constraints (16) are linearized in the following way:

$$p_{ij}^{sd} \leq x_{ij}^{sd} + x_{ji}^{sd} \qquad s \in N, d \in N, \{i, j\} \in E \tag{20}$$

$$p_{ij}^{sd} \leq \sum_{k=0}^{K} z_{ij}^{k}(1 - a_{ij}^{k}) \qquad s \in N, d \in N, \{i, j\} \in E \tag{21}$$

$$p_{ij}^{sd} \geq x_{ij}^{sd} + x_{ji}^{sd} + \sum_{k=0}^{K} z_{ij}^{k}(1 - a_{ij}^{k}) - 1 \qquad s \in N, d \in N, \{i, j\} \in E \tag{22}$$

Constraints (17) are linearized in the same way. Therefore, the disaggregated model is formulated as a MILP model.

5 Aggregated Model

The aggregated model considers the path pair availability, and so constraints (18) and (19) are generalized. Consider the additional decision variables:

λ_{WP}^{sd} real variable accounting for the availability of the WP from s to d

λ_{BP}^{sd} real variable accounting for the availability of the BP from s to d

Then constraints (18) and (19) are replaced by the generalized constraints given by:

$$1 - \sum_{\{i,j\}\in E} p_{ij}^{sd} \geq \lambda_{WP}^{sd} \qquad s \in N, d \in N \tag{23}$$

$$1 - \sum_{\{i,j\}\in E} q_{ij}^{sd} \geq \lambda_{BP}^{sd} \qquad s \in N, d \in N \tag{24}$$

$$1 - (1 - \lambda_{WP}^{sd})(1 - \lambda_{BP}^{sd}) \geq \Lambda \qquad s \in N, d \in N \tag{25}$$

Constraints (25) can be rewritten as:

$$\lambda_{WP}^{sd} + \lambda_{BP}^{sd} - \lambda_{WP}^{sd} \cdot \lambda_{BP}^{sd} \geq \Lambda \qquad s \in N, d \in N \tag{26}$$

Note that these constraints are nonlinear. Consider the additional decision variables given by $\phi^{sd} = \lambda_{WP}^{sd} \cdot \lambda_{BP}^{sd}$. Since λ_{WP}^{sd} and λ_{BP}^{sd} are real variables and not binary, the McCormick envelopes can only provide a convex relaxation. Therefore, we get $\phi^{sd} \leq \lambda_{WP}^{sd} \cdot \lambda_{BP}^{sd}$ instead of guaranteeing the equality [3].

Constraints (26) are then linearized using McCormick envelopes to obtain the convex relaxation in the following way:

$$\lambda_{WP}^{sd} + \lambda_{BP}^{sd} - \phi^{sd} \geq \Lambda \qquad s \in N, d \in N \tag{27}$$

$$\phi^{sd} \geq \chi^L \lambda_{BP}^{sd} + \nu^L \lambda_{WP}^{sd} - \chi^L \nu^L \qquad s \in N, d \in N \tag{28}$$

$$\phi^{sd} \geq \chi^U \lambda_{BP}^{sd} + \nu^U \lambda_{WP}^{sd} - \chi^U \nu^U \qquad s \in N, d \in N \tag{29}$$

$$\phi^{sd} \leq \chi^U \lambda_{BP}^{sd} + \nu^L \lambda_{WP}^{sd} - \chi^U \nu^L \qquad s \in N, d \in N \tag{30}$$

$$\phi^{sd} \leq \chi^L \lambda_{BP}^{sd} + \nu^U \lambda_{WP}^{sd} - \chi^L \nu^U \qquad s \in N, d \in N \tag{31}$$

$$\chi^L \leq \lambda_{WP}^{sd} \leq \chi^U,\ \nu^L \leq \lambda_{BP}^{sd} \leq \nu^U \qquad s \in N, d \in N \tag{32}$$

where χ^L and χ^U are the lower and upper bounds for λ_{WP}^{sd}, and ν^L and ν^U are the lower and upper bounds for λ_{BP}^{sd} (constraints (32)) – these bounds need to be estimated and given as parameters. Constraints (28) and (29) are the underestimates for ϕ^{sd}, while constraints (30) and (31) are the overestimates.

In this way, the aggregated model can be approximated by a MILP model.

6 Solving the Aggregated Model

Note that the convex relaxation guarantees $\phi^{sd} \leq \lambda^{sd}_{WP} \cdot \lambda^{sd}_{BP}$, instead of the equality. This means that when $\phi^{sd} < \lambda^{sd}_{WP} \cdot \lambda^{sd}_{BP}$, we have that (27) is satisfied, but not necessarily (26). Consider the slack variable $\delta^{sd} \geq 0$:

$$\phi^{sd} + \delta^{sd} = \lambda^{sd}_{WP} \cdot \lambda^{sd}_{BP} \tag{33}$$

Then, constraints (26) become:

$$\lambda^{sd}_{WP} + \lambda^{sd}_{BP} - \left(\phi^{sd} + \delta^{sd}\right) \geq \Lambda \tag{34}$$

Consider Δ as a lower bound of $\delta^{sd}, s \in N, d \in N$. Then, (34) can be rewritten as:

$$\lambda^{sd}_{WP} + \lambda^{sd}_{BP} - \phi^{sd} \geq \Lambda + \Delta \tag{35}$$

So instead of considering that the target availability is Λ, we consider Λ' where $\Lambda' = \Lambda + \Delta$.

By tightening the envelop bounds $\chi^U, \chi^L, \nu^U, \nu^L$, the solution for the convex relaxation becomes closer to the optimal one. We propose an algorithm which is an iterative approach that takes advantage of some characteristics specific to the link availability upgrade problem, to tighten the bounds of the convex envelope in order to obtain near-optimal solutions. Since we are minimizing the upgrade cost, the lower bounds χ^L, ν^L have the most impact on the problem. The upper bounds χ^U, ν^U only need to be large enough to contain all the desired solutions. Therefore, the upper bounds can be set to $\chi^U = \nu^U = \Lambda$.

It has been shown in [2] that improving the WP availability has a lower impact on the cost function, than improving the BP availability. Given this, the lower bound ν^L is set to a sufficiently small value so as to not remove any desired solutions, but large enough so as to not provide too many undesired solutions (solutions that satisfy (27) but not (26)). Then, χ^L is the bound that will be adjusted in the algorithm by tightening it iteratively within a range $[\chi^L_{\min}, \chi^L_{\max}]$. The parameter $\chi^L_{\min}$ is set to be the availability of the longest shortest path between any two nodes in the network, and $\chi^L_{\max}$ is set to a sufficiently large number so as to not remove any desired solutions in the network. The granularity step by which χ^L is incremented is denoted by σ.

With all these considerations, the iterative approach is given as follows:

1. Set $\chi^L = \chi^L_{\min}$ and $\Lambda' = \Lambda + \Delta$

2. Solve the convex relaxation with χ^L and Λ'

3. If the solution obtained does not satisfy (26), then increment χ^L by σ, i.e., $\chi^L + \sigma$

4. Repeat steps (2)-(3) until a solution that satisfies (26) is obtained or until $\chi^L_{\max}$ is reached (note that this is the lowest cost solution; i.e., if we proceed with the iterative approach the following solutions will have a higher cost, in general)

7 Computational Results

We used the three networks in [2]: polska, spain and italia14. The characteristics of these networks are in Table 1 showing the number of nodes $|N|$, the number of links $|E|$, the average node degree and the network diameter which is given by the longest shortest path between any two nodes in the network.

Table 1: Characteristics of the networks

Network	$\lvert N\rvert$	$\lvert E\rvert$	Avg node deg	Diameter (km)
polska	12	18	3.00	811
spain	14	22	3.14	1034
italia14	14	29	4.14	950

For the link availability parameters, we have considered $\varepsilon = 0.5$ and $K = 5$ levels of upgrade. The target path pair availability is "five-nines", i.e., $\Lambda = 0.99999$.

For the disaggregated model (section 4), we have considered three pairs of values for the working and backup path availability targets. For the working path, we have considered $\Lambda_{WP} = \{0.997, 0.998, 0.999\}$, while the backup path target was determined to achieve Λ with the given Λ_{WP}:

$$\Lambda_{BP} = \frac{\Lambda - \Lambda_{WP}}{1 - \Lambda_{WP}} \tag{36}$$

which gives $\Lambda_{BP} = \{0.99(6), 0.995, 0.99\}$, respectively.

The aggregated model (section 5) was solved using the iterative algorithm described in section 6. The upper bounds χ^U, ν^U were set to Λ. The lower bound ν^L was set to 0.99 and $\chi^L_{\max}$ was set to 0.999 by computational tests. Parameter $\chi^L_{\min}$ was set to the availability of the longest shortest path in the network. The granularity step σ was chosen so that the range $[\chi^L_{\min}, \chi^L_{\max}]$ for incrementing χ^L was divided into 50 slots. This choice was motivated by the trade-off between solution quality and computational effort. Finally, Δ was set to

0.00001 by computational tests to account for the gap to achieve Λ. The parameters and their values are summarized in Table 2.

Table 2: Parameter values used in the iterative approach

Parameter	Value
Δ	0.00001
ν^U, χ^U	Λ
ν^L	0.99
χ^L	$[\chi^L_{\min}, \chi^L_{\max}]$
$\chi^L_{\min}$	availability of the longest shortest path
χ^L_{max}	0.999
σ	$\left(\chi^L_{\max} - \chi^L_{\min}\right)/50$

The disaggregated model and iterative approach were implemented in C/C++ and CPLEX 12.9 Callable libraries were used for solving the MILP models. All computational tests were run on a 8-core Intel Core i7 PC with 64 GB RAM @ 3.6 GHz.

We compare the solutions of the aggregated model solved by the iterative approach with those of the disaggregated model for the different values of Λ_{WP} (and respective Λ_{BP} values). The results are summarized in Table 3. The first column shows the network. Column 'Λ_{WP}' indicates if the aggregated model was used ('agg') or in case of the disaggregated model shows the value of Λ_{WP} considered. Column 'Cost' shows the spanning tree upgrade cost to achieve the desired Λ, and columns 'k' indicate how many links were upgraded to level $k = 1, 2, 3, 4, 5$. Finally, column 'Time (s)' shows the computational time for solving each instance in seconds.

Note that for all networks, the aggregated model provided the best solution, i.e., the solution with lowest cost. It is also the model that demanded higher computational effort due to the iterative algorithm (as seen by the computational time). In the disaggregated model, the Λ_{WP} value that gave the best solution was 0.998, however still with a cost much higher than the cost of the solution of the aggregated model. Also in the disaggregated model, the Λ_{WP} value that gave the highest cost solution was 0.999 (with $\Lambda_{BP} = 0.99$) for all networks, since in these networks the availability of the BPs are typically much higher than 0.99 and so the target availability of 0.999 for the WPs is unnecessarily high. Interestingly, $\Lambda_{WP} = 0.997$ provides a worse solution than $\Lambda_{WP} = 0.998$, since it requires an availability for the BPs of $\Lambda_{BP} = 0.99(6)$ instead of 0.995, and it has been shown [2] that improving the availability of the BPs has a higher impact on the upgrade cost.

For illustrative purposes, we show the spanning trees for each solution of polska in

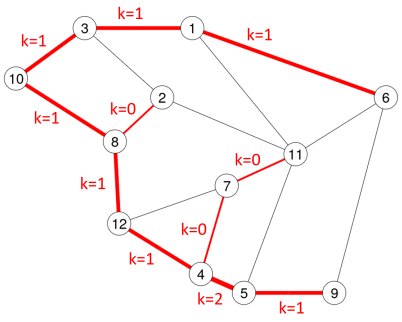
Table 3: Comparison between the aggregated and disaggregated models

Network	Λ_{WP}	Cost	k 1	2	3	4	5	Time (s)
polska	agg	988.4	7	1	0	0	0	1887.2
	0.999	2837.7	3	5	2	1	0	90.8
	0.998	1795.3	3	7	0	0	0	53.2
	0.997	1882.6	6	4	0	0	0	168.1
spain	agg	3293.1	5	6	0	0	0	136811.6
	0.999	5393.4	6	5	2	0	0	1417.0
	0.998	4065.3	4	7	2	0	0	1890.6
	0.997	4825.7	2	4	6	0	0	3225.8
italia14	agg	2376.8	9	2	2	0	0	64376.7
	0.999	4399.4	6	5	2	0	0	7201.2
	0.998	2783.0	6	5	1	1	0	7520.8
	0.997	4127.0	2	10	1	0	0	8200.1

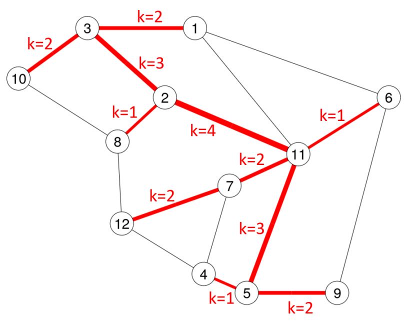
Fig. 1. The spanning tree links are shown in red and their thickness is proportional to their upgrade level k. Note that in the aggregated model which gave the lowest cost, 7 links are upgraded to level $k = 1$ and link $\{4, 5\}$ is upgraded to level $k = 2$, while 3 links in the spanning tree do not need to be upgraded (links with $k = 0$). On the other hand, in the disaggregated model for $\Lambda_{WP} = 0.999$ which gave the solution with highest cost, all the spanning tree links are upgraded. This is also true for the spain and italia14 network, where the total number of links upgraded for $\Lambda_{WP} = 0.999$ is $|N| - 1$ (see Table 3). For the other two cases (disaggregated model with $\Lambda_{WP} = 0.998$ and 0.997), we can see that only one link of the spanning tree is not upgraded in Fig. 1.

In Fig. 2, we show the minimum, maximum and average availabilities for the WPs (dotted columns), BPs (striped columns) and path pairs (solid color columns) in each solution (aggregated model, and disaggregated model with different values of Λ_{WP}).

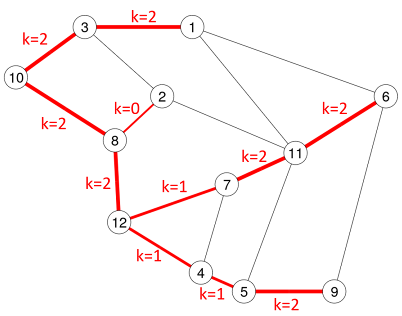
We can see that the BP availabilities for the disaggregated model with $\Lambda_{WP} = 0.999$ are the smallest on average among all cases, and the WP availabilities are the largest on average, which is expected since the target WP availability is quite high in that case. On the other hand, the BP availabilities for the disaggregated model with $\Lambda_{WP} = 0.997$ are the largest on average among all cases, since the required BP availability target is the highest. The aggregated model which provides the lowest cost solutions, manages the availabilities of the WPs and BPs in a much more balanced manner: note that the average



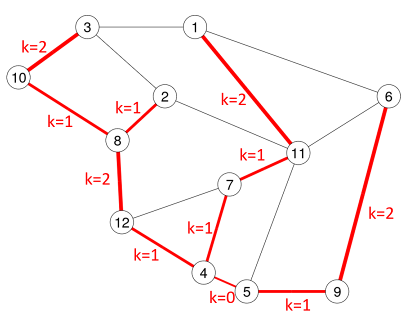
(a) Aggregated model



(b) Disaggregated model with $\Lambda_{WP} = 0.999$



(c) Disaggregated model with $\Lambda_{WP} = 0.998$



(d) Disaggregated model with $\Lambda_{WP} = 0.997$

Figure 1: Spanning trees with upgraded links for polska

WP availability for the aggregated model tends to be smaller than for the other models, since the aggregated model takes advantage of the BP availability which is considered concurrently with the WP availability for each path pair. In the disaggregated model, the case with $\Lambda_{WP} = 0.998$ has a more balanced behavior, although still with higher average WP availabilities than the aggregated model. In this situation, the availability of the path pair is achieved mainly with the increase of the WP availability, and with only a small increase of the BP availability.

Finally, in Table 4 we show the values of χ^L for which the solutions of the aggregated model were found. Note that the minimum availability of the WPs is close to χ^L for polska and spain, but is much higher than χ^L for italia14, since χ^L is a minimum bound for the convex relaxation (constraints (27)-(32)) and not necessarily the minimum WP availability (determined by λ_{WP}^{sd} – constraints (23)).

Table 4: Values of χ^L for which the solution of the aggregated model was found

Network	χ^L
polska	0.995349163
spain	0.994560000
italia14	0.996034003

8 Conclusions

In this paper, we addressed the problem of path pair availability guarantees with working and backup paths between all pairs of nodes in the network. We assume that the working paths are routed over a spanning tree whose links can be upgraded to have increased availability at a given cost. We formulate the problem as an optimization problem minimizing the upgrade cost, while guaranteeing a pair of working and backup paths with availability of at least Λ. The path pair availability constraints are nonlinear. To solve the problem we adapted a previously proposed disaggregated model [2] where the working and backup paths are considered separately to obtain linear constraints. To approximate the solutions to the optimal ones, we also propose an aggregated model that considers a convex relaxation model with linear constraints and present an iterative approach to solve it.

The results show that by considering the path pair availability in an integrated way (rather than in a separate way as in the disaggregated model), a lower availability upgrade cost is achieved. Moreover, we have seen in the disaggregated model, that for different pairs of working and backup target availability values, the solutions may be quite different, and that when the WP target availability is reasonable (neither too high nor too low) then the solutions are better, although still worse than with the aggregated model.

As future work, we envision to solve this problem in a more efficient way in terms of computational time and solution quality. The possibility of considering heuristics or meta-heuristics for solving larger instances of problems (much larger networks) will be considered.

References

[1] Abdulaziz Alashaikh, Teresa Gomes, and David Tipper. The spine concept for improving network availability. *Computer Networks*, 82:4–19, 2015.

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

[3] Pedro M. Castro. Tightening piecewise McCormick relaxations for bilinear problems. *Computers & Chemical Engineering*, 72:300–311, 2015.

[4] NGOC-Thanh Dinh and Younghan Kim. An efficient reliability guaranteed deployment scheme for service function chains. *IEEE Access*, 7:46491–46505, 2019.

[5] Haijun Geng, Han Zhang, Xingang Shi, Zhiliang Wang, Xia Yin, Ju Zhang, Zhiguo Hu, and Yong Wu. A hybrid link protection scheme for ensuring network service availability in link-state routing networks. *Journal of Communications and Networks*, 22(1):46–60, Feb. 2020.

[6] Mahdi Ghamkhari, Ashkan Sadeghi-Mobarakeh, and Hamed Mohsenian-Rad. Strategic bidding for producers in nodal electricity markets: A convex relaxation approach. *IEEE Transactions on Power Systems*, 32(3):2324–2336, 2017.

[7] Akshay Gupte, Shabbir Ahmed, Myun Seok Cheon, and Santanu Dey. Solving mixed integer bilinear problems using MILP formulations. *SIAM Journal on Optimization*, 23(2):721–744, 2013.

[8] David Hutchison and James P.G. Sterbenz. Architecture and design for resilient networked systems. *Computer Communications*, 131:13–21, Oct. 2018.

[9] Mowen Lu, Harsha Nagarajan, Russell Bent, Sandra D. Eksioglu, and Scott J. Mason. Tight piecewise convex relaxations for global optimization of optimal power flow. In *2018 Power Systems Computation Conference (PSCC)*, pages 1–7, 2018.

[10] Saeed D. Manshadi and Mohammad E. Khodayar. A tight convex relaxation for the natural gas operation problem. *IEEE Transactions on Smart Grid*, 9(5):5467–5469, 2018.

[11] Júlio Mendonça, Ricardo Lima, Ermeson Andrade, Julian Araujo, and Dong Seong Kim. Multiple-criteria evaluation of disaster recovery strategies based on stochastic models. In *2020 16th International Conference on the Design of Reliable Communication Networks DRCN 2020*, pages 1–7, Milan, Italy, 25-27 Mar. 2020.

[12] Jacek Rak, David Hutchison, Janos Tapolcai, Rasa Bruzgiene, Massimo Tornatore, Carmen Mas-Machuca, Marija Furdek, and Paul Smith. Fundamentals of communication networks resilience to disasters and massive disruptions. In Jacek Rak and David Hutchison, editors, *Guide to Disaster-Resilient Communication Networks*, pages 1–43. Springer, Cham, 2020.

[13] Dorabella Santos, Teresa Gomes, and David Tipper. SDN controller placement with availability upgrade under delay and geodiversity constraints. *IEEE Transactions on Network and Service Management*, 18(1):301–314, 2021.

[14] Dorabella Santos, Lúcia Martins, and Teresa Gomes. A biobjective availability optimization problem with non-linear constraints. *Submitted for publication*, 2022.

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

[16] Jean-Philippe Vasseur, Mario Pickavet, and Piet Demeester. *Network Recovery – Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Elsevier, 2004.

[17] Wenjing Wang and John Doucette. Availability optimization in shared-backup path protected networks. *Journal of Optical Communications and Networking*, 10(5):451–460, May 2018.

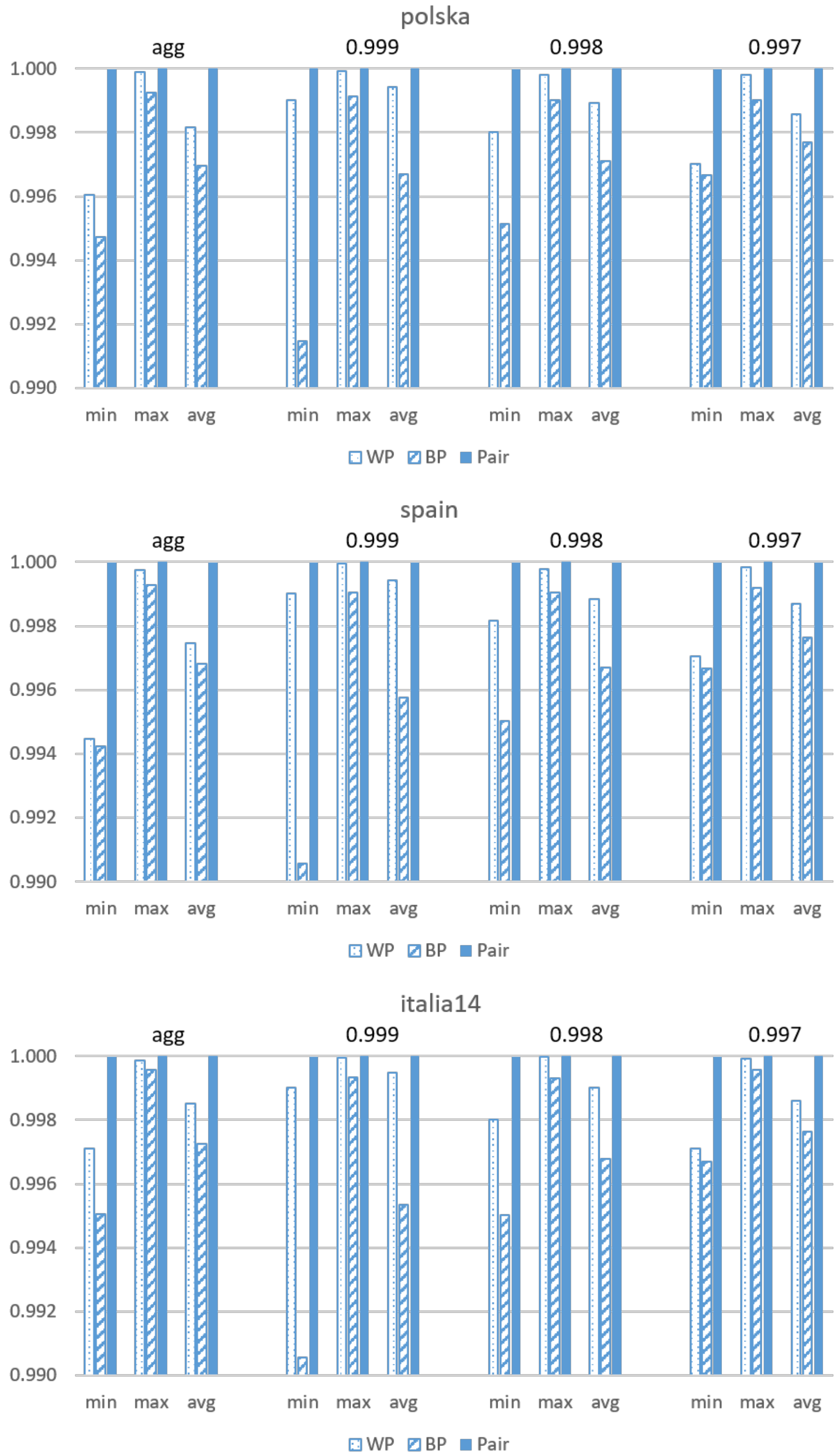


Figure 2: Minimum, maximum and average availabilities for WPs, BPs and path pairs in the solutions