Heuristic Approach for the Design of a High Availability Structure

Rita Girão-Silva [a,b], Lúcia Martins [a,b], Teresa Gomes [a,b],
David Tipper [c], Abdulaziz Alashaikh [c]

a DEEC-FCTUC
b INESC-Coimbra
c Univ. Pittsburgh

DRCN 2019

15th International Conference on the
Design of Reliable Communication Networks (DRCN 2019)
Coimbra, Portugal March 19-21, 2019

Summary

1. Introduction and motivation
2. Problem formulation
   - Two different single objective routing models aimed at: (i) the maximization of the average availability for Working Paths (WPs); (ii) the maximization of the average availability for all the flows (with the Backup Path (BP) on the spine or off the spine)
3. Heuristic resolution approach
   - Cost metric used in the spanning tree generation
   - Other heuristics for comparison
4. Some experimental results
5. Conclusions and further work

1. Introduction and Motivation

- The availability of the resources in communication networks is critical.
- Providing adequate levels of availability for every demand in a network is of paramount importance.
- For each demand, a pair of edge-disjoint paths is selected: the WP is used in regular conditions; in the event of failures in any of the components of the WP, a BP is used.
- This path pair provides end-to-end protection for critical service demands in the network.

1. Introduction and Motivation

- Considering the topological structure of a network, we select a set of links that provide a high availability path to be used by the different end-to-end demands.
- This set of links constitutes a high availability structure (the *spine*) and is used as the WP for each demand. In our approach, the spine is a spanning tree.
- The BP may use any link in the network including links in the spine. However, the WP and the BP must be edge-disjoint.
- The availability of the edges depends on whether they belong to the high availability structure (a_{ON}) or not (a_{OFF}) (similarly to previous work [1]).

[1] A. Alashaikh, T. Gomes, D. Tipper.
The spine concept for improving network availability.
Computer Networks, 82:4-19, 2015.

1. Introduction and Motivation
Previous work

- In [1] the BP should avoid edges of the spine, which is not required here. Allowing the BP to use edges of the spine requires a higher capacity of the links in the spine.
- Previous work [2] focuses on the optimization of the availability of the WP only (average and minimal). Different metrics for the enumeration of the spanning trees are proposed. The availability of the edges depends on their length and is taken into account in the selection of a suitable spine.

[1] A. Alashaikh, T. Gomes, D. Tipper.
The spine concept for improving network availability.
Computer Networks, 82:4-19, 2015.
[2] R. Girão-Silva, L. Martins, T. Gomes, D. Tipper, A. Alashaikh.
Improving network availability – a design perspective.
ICICT2018, Advances in Intelligent Systems and Computing, vol. 797, pp. 799-815, Springer, 2019.

2. Problem Formulation

- We consider that the high availability structure (spine) should form a spanning tree. The total cost of the spanning tree is the sum of costs of the edges that form it.
- The availability of a path $p(s,t)$ between nodes s and t is the product of the availability of the edges that form it.
- Availability of a path pair:

$$A_{(s,t)} = 1 - \left(1 - A_{WP_{(s,t)}}\right)\left(1 - A_{BP_{(s,t)}}\right)$$

2. Problem Formulation
Problem 1

- Objective: maximization of the sum of the availabilities of the WPs of all the flows in the network (equivalent to the maximization of the average availability of the WPs of all the flows in the network).
- Constraints to be considered:
  - Flow conservation;
  - Loop free routing;
  - The WPs should form a spanning tree.
  - Edge-disjointness for the path pair;
- The function for calculating the availability of the WPs is linearized. An ILP model is formulated.

2. Problem Formulation
Problem 1

- As the availability of the edges in the spine is fixed (a_{ON}), finding the most available WP for each demand is equivalent to finding the WP with the smallest number of hops in the spine for each demand. ⇒ Problem of finding a spanning tree with the smallest number of hops for the paths in that spanning tree.
- Afterwards: application of Dijkstra algorithm to find an edge-disjoint BP for each demand, where the BP (i) may or (ii) may not use edges of the spine.

2. Problem Formulation
Problem 2

- Objective: maximization of the sum of the availabilities of all the flows in the network (equivalent to the maximization of the average availability of all the flows in the network).
- Similar constraints to the ones in Problem 1, plus constraints for calculation of the availability for each flow.
- Need for linearization of the expressions involving the product of variables.
- Two variations of the problem are formulated, regarding whether the BP (i) may or (ii) may not use edges of the spine.

3. Heuristic Resolution Approach

- As we are dealing with an unweighted graph, we consider the length of each edge as 1.
- Let $\sigma_k(s,t) = |P_k(s,t)|$, where $P_k(s,t)$ is the set of paths between s and t, whose length is not higher than the shortest path length L_0 plus k (non-negative integer).
- Let $\sigma_k(s,t|l)$ be the number of paths in $P_k(s,t)$ which include edge l.
- Topological structural measure: k-betweenness centrality for edge l [3]:

$$B_k(l) = \sum_{\substack{\forall\, s,t: \\ s \neq t}} \frac{\sigma_k(s,t|l)}{\sigma_k(s,t)}$$

[3] K. Jiang, D. Ediger, D. A. Bader. Generalized k-betweenness centrality using short paths and a parallel multithread implementation. 2009 International Conference on Parallel Processing (ICPP'09), pp. 542-549, Sep. 2009.

3. Heuristic Resolution Approach

- Centrality cost of each edge:
$$c_k(l) = -B_k(l) + \max_{\forall \ell} B_k(\ell) + 1$$
- The cost of an edge is higher if it is less central (i.e. it appears in a smaller number of short paths).
- Heuristic based on the calculation of spanning trees (using Prim's algorithm) minimizing the total centrality cost and avoiding certain edges (to diversify the obtained trees).
- For a (s, t) demand for which no edge-disjoint BP exists, consider s or t (the one with higher average centrality) and consider avoiding one of the edges in the spine passing in this node, for a certain number of iterations.
- Average centrality cost for a node: average value of $c_k(l)$ for all edges l leaving or entering the node.

3. Heuristic Resolution Approach

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**

  loop

    if $listReset == 1$ **then**

      Reset the current list of edges to be avoided.

    end if

    The first time this inner loop is run, no new edges to be avoided are defined; the following $|L|$ 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).

3. Heuristic Resolution Approach

repeat

Calculate spine (with Prim's algorithm) that tries to avoid edges on the list and minimizes the total centrality cost, with centrality cost of each edge $c_k(l)$.

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

3. Other Heuristic Resolution Approaches for Comparison

- In [1], structural properties of a network topology are used to select a suitable spine. The edge cost is a weighted sum of the edge degree and the edge betweenness.
- A minimum cost spanning tree is calculated.
- If an edge disjoint BP cannot be obtained for a node pair, the set of conflicting edges (those shared by a WP and the maximally link disjoint corresponding BP) are identified.
- A new spanning tree is obtained avoiding each of those links (and all possible combination of them).
- Among the admissible spanning trees the one with best performance according to the relevant metric is saved.

[1] A. Alashaikh, T. Gomes, D. Tipper.
The spine concept for improving network availability.
Computer Networks, 82:4-19, 2015.

3. Other Heuristic Resolution Approaches for Comparison

- In [2], a heuristic based on the enumeration of candidate spines (spanning trees) by non-decreasing order of the total cost is proposed.
- The cost metric for each edge is $c_k(l)$.
- A total of $|N| \cdot |L|$ trees is obtained, of which some may be infeasible due to not having edge-disjoint path pairs for all the demands.

[2] R. Girão-Silva, L. Martins, T. Gomes, D. Tipper, A. Alashaikh.
Improving network availability – a design perspective.
ICICT2018, Advances in Intelligent Systems and Computing, vol. 797, pp. 799-815, Springer, 2019.

4. Experimental Results

- Results for two networks

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

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

- Exact results were obtained for problem 1 (polska and italia) and for problem 2 (polska) where the BP (i) may or (ii) may not use edges of the spine.
- No distinction is made between alternative optimal solutions.

4. Experimental Results

- Experiments with the heuristic using the cost metric $c_k(l)$ 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 [2], 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.

[2] R. Girão-Silva, L. Martins, T. Gomes, D. Tipper, A. Alashaikh.
Improving network availability – a design perspective.
ICICT2018, Advances in Intelligent Systems and Computing, vol. 797, pp. 799-815, Springer, 2019.

4. Experimental Results

TABLE II: Results for the polska network

(a) BP may use edges of the spine

Method	k	max $Iter$	A_a	A_a^{WP}	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^{WP}	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

[14]↔[1]
[20]↔[2]

4. Experimental Results

TABLE III: Results for the italia network

(a) BP may use edges of the spine

Method	k	max $Iter$	A_{a}	A_{a}^{WP}	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}^{WP}	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

[14]↔[1]
[20]↔[2]

4. Experimental Results

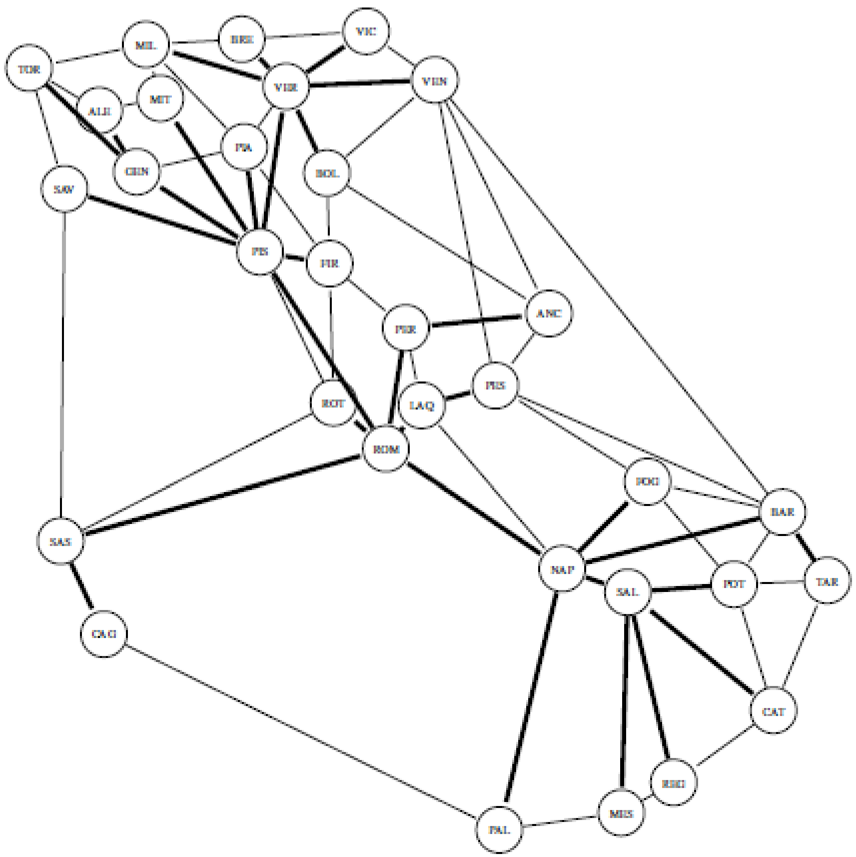
- Information on the tables: availability values A_a and A_a^{WP} obtained in the experiments considering the exact formulations and the heuristics; h_S, average number of hops in the spine; di_S, spine diameter.
- No distinction is made between alternative optimal solutions. Example: different spines with the same value of A_a^{WP} may lead to different solutions in terms of BPs and A_a.
- Execution times:
  - Exact resolution – polska: a few seconds for problem 1 and a few hours for problem 2; italia: a few days for problem 1.
  - Heuristics – a few seconds for any network, except heuristic in [2], which takes about 10 minutes for the italia network.

[2] R. Girão-Silva, L. Martins, T. Gomes, D. Tipper, A. Alashaikh. Improving network availability – a design perspective. ICICT2018, Advances in Intelligent Systems and Computing, vol. 797, pp. 799-815, Springer, 2019.

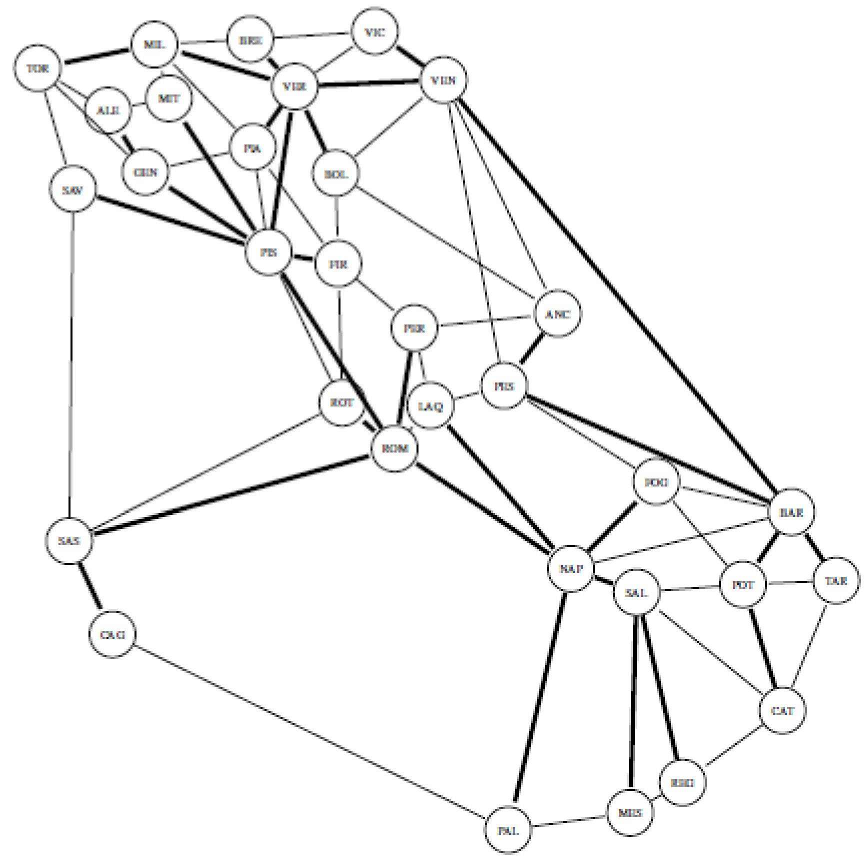
4. Experimental Results

- The spines leading to the best solution in terms of A_a present a higher diameter and a higher average number of hops of the WPs. The spine does not include mainly the central edges.
- The spines leading to the best solution in terms of A_a^{WP} tend to include mainly the central edges. They present a lower diameter and a lower average number of hops of the WPs.
- The maximization of the WP availability is not closely related to the maximization of the path pair availability. In fact, for the best solution in terms of A_a, the average number of hops of the WPs is higher and consequently 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}.

4. Experimental Results



(a) Exact solution for problem 1



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

4. Experimental Results

- The exact solution that minimizes A_a^{WP} (see Fig. (a)) 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 solution obtained considering the edge cost $c_k(l)$ with $k = 2$ (see Fig. (b)) presents some very long and apparently not central edges. 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 $c_k(l)$.

4. Experimental Results

- 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 higher k, i.e. longer paths in the calculation of the metric $B_k(l)$ (used in the spanning tree calculation), the measure of centrality of the edges becomes more accurate. By identifying the more central edges, the obtained spines tend to focus on those edges, originating trees with smaller diameter.

4. Experimental Results

- Comparing the results with different heuristics:
  - polska, allowing the BPs to use edges of the spine: the exact solution and the solution with heuristic in [1] present a better value for A_a, at the expense of longer WPs (trade-off).
  - italia, allowing the BPs to use edges of the spine: the proposed heuristic finds the best solution for A_a, without incurring in longer WPs.
  - italia: no heuristic found a solution with the optimal exact value of A_a^{WP} (see Fig. (a)).

[1] A. Alashaikh, T. Gomes, D. Tipper.
The spine concept for improving network availability.
Computer Networks, 82:4-19, 2015.

5. Conclusions

- The proposed heuristic (based on spanning tree calculation with some edges being avoided) leads to solutions with high availability.
- The considered centrality measure used for calculation of the costs $c_k(l)$ proved to be appropriate, leading to solutions with high availability A_a.
- Future work:
  - other performance measures related to the minimal availability of the WPs or the path pairs;
  - turning the proposed heuristic into a tabu search procedure, with some degree of randomness in the selection of edges to avoid;
  - considering information on traffic demands and edge capacities;
  - other spine configurations (not necessarily a spanning tree).

Acknowledgements and Funding

- This article is based upon work from COST Action CA15127 “Resilient communication services protecting end-user applications from disaster-based failures” (RECODIS).

EUROPEAN COOPERATION IN SCIENCE AND TECHNOLOGY

RECODIS
Resilient communication services protecting end-user applications from disaster-based failures

- 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.

Cofunded by: CENTRO 2020 PORTUGAL 2020 UNIÃO EUROPEIA Fundo Europeu de Desenvolvimento Regional FCT Fundação para a Ciência e a Tecnologia