Exact calculation of network robustness

Oriol Lordan[1], Maria Albareda-Sambola[2]
1. Management Department. UPC, Terrassa, Spain. oriol.lordan@upc.edu
2. Statistics and Operations Research Department. UPC, Terrassa, Spain

Abstract

Finding the most critical nodes regarding network connectivity has attracted the attention of many researchers in infrastructure networks, power grids, transportation networks and physics in complex networks. Static robustness of networks under intentional attacks analyses the ability of a system to maintain its connectivity after the disconnection or deletion of a series of targeted nodes. In this context, connectivity is typically measured by the size of the remaining largest connected component. When targeting these nodes, previous literature has mostly used adaptive strategies that sequentially remove central nodes, or created heuristics in order to improve the results of the adaptive strategies. The proposed methodology based on mathematical programming allows to identify, for every fraction of disconnected or removed nodes, the set that minimizes the size of the largest connected component of a network, i.e. it allows to calculate the exact (most critical) robustness of a network.

Keywords: Robustness; Complex networks; Optimization; MILP

1 Introduction

Static robustness of networks under intentional attacks analyses the ability of a system to maintain its connectivity after the disconnection or deletion of a series of targeted nodes. In this context, the connectivity of the resulting network is typically measured by the size of the largest connected component (LCC). Finding the most critical nodes to disconnect a network has attracted the attention of many researchers that analyzed the vulnerability of the Internet (Cohen et al., 2001), power grids (Albert et al., 2004; Solé et al., 2008), infrastructure networks (Latora and Marchiori, 2005) and different transportation networks (Lordan et al., 2014; Feng and Wang, 2013).Indeed, maintenance and protection of real-world network-based systems can be performed more efficiently if the most critical nodes of the network are accurately identified.

The selection of the critical nodes is typically driven by some node properties that try to identify the most connected ones (Broder et al., 2000; Albert and Barabási, 2002; Lordan et al., 2014). Although other procedures and heuristics have been tested in order to find the most critical path (Jahanpour and Chen, 2013; Lordan et al., 2015; Pullan, 2015; Deng et al., 2016; Soria et al., 2017), the common procedure for disconnecting the network is to follow an adaptive (greedy) strategy: at each iteration, the most promising node according to its properties is selected and removed from the network. The properties of the remaining nodes are recomputed after every deletion and the process ends when the network is completely disconnected. The three node attributes that showed the best performance (Lordan et al., 2015; Petreska et al., 2010) so far are: *high damage*, *high betweenness* or *high degree*. Recall that, given a network, the damage of a node is the reduction of the LCC size when it is disconnected, and its betweenness is the sum, over all pairs of other nodes, of the fraction of shortest paths going through it.

None of the methods reported above is guaranteed to find the optimal set of nodes to disconnect a network up to any given percentage of its LCC. The aim of this paper is to define a methodology for calculating exactly the robustness of a network by finding, for each number of nodes to remove, the set that minimizes the size of the LCC or the remaining network. This can also be interpreted as finding the most critical or important nodes to keep the cohesion of a network.

Finding the optimal sets of nodes is computationally expensive. In this work, we will use two ideas in order to be able to deal with medium-sized and large networks. On the one side, we consider that network managers are most often not interested in completely disconnecting a network but disconnecting it only up to a given LCC size (for instance, 5% of the graph size). On the other hand, we will be able to reduce the graph whenever it has small communities of nodes that are connected to the rest of the network through one single node.

The rest of the paper is organized as follows. In Section 2 we give a complete description of the algorithm, and proof its validity. To illustrate the use of the algorithm, and the utility of exact algorithms to have precise information on the performance of the heuristics available in the literature, we have computed the robustness for several graphs. The obtained results are reported in Section 3. Finally, Section 4 exposes the conclusions drawn from our experiments.

2 Exact algorithm

We will consider two auxiliary subproblems related to the detection of critical nodes in a network:

Problem $\mathbf{P1(L)}$

Given a connected undirected graph $G = (N, E)$ and an integer L, identify $S_1(L) \subseteq N$ of minimum cardinality that, when removed, decomposes G into a number of connected components, each containing at most L nodes.

Problem $\mathbf{P2(K)}$

Given a connected undirected graph $G = (N, E)$ and an integer K, identify $S_2(K) \subseteq N$ with $|S_2(K)| \leqslant K$ that minimizes the size of the LCC (L^*_K) of the graph obtained after removing from G the nodes in $S_2(K)$.

These problems will be solved by exploiting the integer programming formulations proposed in Veremyev et al. (2014a) and the following extension of a result from the same work:

Proposition 1. *Given $G = (N, E)$ and $u \in N$ that disconnects G, let S_u be the set of nodes different from u not belonging to the LCC when u is disconnected (see Figure 1).*

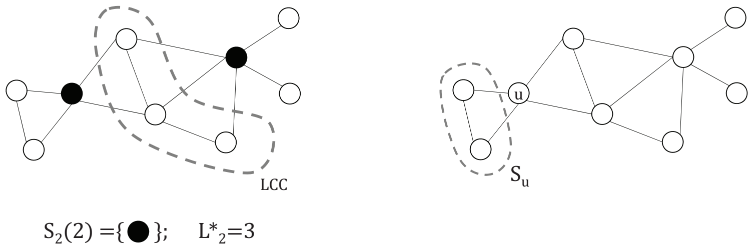

Figure 1: Notation: Example

Then, given a lower bound ℓ on L^*_K, if $|S_u| < \ell$, there exists an optimal solution of $P2(K)$ with $S_2(K) \cap S_u = \emptyset$.
The same result holds for $P1(L)$ and $\ell = L$.

Proof.
Suppose that $v \in S_2(K) \cap S_u$. Then, there are two possibilities:

- $\mathbf{u \notin S_2(K)}$: Consider the solution $S_2(K) \cup \{u\} \setminus \{v\}$.

  By including u in the set of interdicted nodes, S_u is disconnected from G, leading to one or more connected components of size smaller than $\ell \leqslant L^*_K$, which include node v. Also, the size of the original connected component containing u will be now reduced by at least 1. On the other hand, the sizes of the remaining connected components (those not intersecting $S_u \cup \{u\}$) remain unaltered.

  Therefore, the LCC defined by this new solution is not larger than L^*_K.

- $\mathbf{u} \in \mathbf{S_2}(\mathbf{K})$:

  In this case, by removing v from $S_2(K)$ some components might be merged. However, since $u \in S_2(K)$, S_u is disconnected from the rest of the graph and, therefore, none of the resulting components can have size larger than $|S_u|$, which is, in turn, smaller than L^*_K.

  Taking any other node $u' \in N \setminus (S_u \cup S_2(K))$ in the solution, the remaining connected components either keep the same size or become smaller.

  Therefore, again, we can build a solution without node v whose value is not larger than L^*_K.

If $|S_2(K) \cap S_u| > 1$, the previous reasoning can be repeated until all nodes from S_u have been removed from the solution.

Since the above operations do not modify the size of the solution set, the result also stands for problem $P1(L)$. □

The above result allows to reduce the size of the graphs where the auxiliary problems **P1** and **P2** will be solved, by removing nodes in S_u, for nodes u with maximal $|S_u| < \ell$. To this end, we will denote $w_u = |S_u|$, D the whole set of removed nodes according to this criterion, and $\bar{D} = N \setminus D$ and $E^{\bar{D}}$, respectively, the set of nodes and the set of edges of the reduced graph.

To formulate **P1** and **P2** we use binary variables:

- v_i for $i \in \bar{D}$. Takes value 1 if node i belongs to the solution, and zero otherwise.

- u_{ij} for $i, j \in \bar{D}$. Takes value 1 if nodes i and j belong to the same connected component of the graph obtained after removing the nodes in the solution, and 0 otherwise.

A formulation of P1 is then:

$$
\begin{aligned}
P1(L)\ \min & \sum_{i\in \bar{D}} v_i && && (1)\\
\text{s.t. } & u_{ij} + v_i \leqslant 1 && i,j \in \bar{D} && (2)\\
& u_{ij} + v_j \leqslant 1 && i,j \in \bar{D} && (3)\\
& u_{ij} + v_i + v_j \geqslant 1 && (i,j) \in E^{\bar{D}} && (4)\\
& u_{ij} - \sum_{(i,k)\in E^{\bar{D}}} u_{kj} \leqslant 0 && i,j \in \bar{D} && (5)\\
& u_{ij} + v_i - \frac{1}{|\delta(i)|} \sum_{(i,k)\in E^{\bar{D}}} u_{kj} \geqslant 0 && i,j \in \bar{D} && (6)\\
& \sum_{j\in \bar{D}} (w_j+1) u_{ij} \leqslant L && i \in \bar{D} && (7)\\
& u_{ij}, v_i \in \{0,1\} && i,j \in \bar{D} && (8)
\end{aligned}
$$

Here, constraints (2) and (3) ensure that the nodes that belong to a solution are not considered to be in any connected component. Constraints (4) ensure that, if none of the endpoints of an edge is part of the solution, then they must belong to the same connected component. Extending this idea, for each pair of nodes $i,j \in \bar{D}$, constraints (5) prevent i from belonging to the same connected component as j unless some node adjacent to i does. If this is the case, then i must either belong to the same component as j or belong to the solution. This is imposed by constraints (6). Here, $|\delta(i)|$ stands for the degree of node i in the reduced graph. Finally, constraints (7) limit the size of all connected components to L. Observe that, for any node that does not remain isolated in the solution, w_{ii} will equal $1 - v_i$ by constraints (2) and (6), so that the size of all connected components is well computed. Finally, note that if an upper bound U is available on the size of $S_1(L)$, the extra constraint

$$\sum_{i\in \bar{D}} v_i \leqslant U \qquad (9)$$

can be added to reduce the feasible region of $P1(L)$. In our computational experiments, we obtained U by applying the three greedy algorithms based on betweenness, damage and degree analyzed in Lordan et al. (2014), and taking the minimum of the obtained solution sizes for each value of L. Note that, by definition, *damage* gives the optimal result on the first disconnection.

Using the same variables as before, $P2$ can be formulated as:

$$
\begin{aligned}
P2(K)\ \ & \min\ z && && (10)\\
& \text{s.t. Constraints } (2),(3),(4),(5),(6),(8) && && \\
& \qquad \sum_{j\in\bar{D}} (w_j+1)u_{ij} \leqslant z && i \in \bar{D} && (11)\\
& \qquad \sum_{j\in\bar{D}} v_j \leqslant K && && (12)\\
& \qquad z \in \mathbb{Z}_+ && && (13)
\end{aligned}
$$

The left hand side of (11) computes the size of the connected component containing node i; therefore, together with the objective function, these constraints force z to account for the size of the LCC. Constraints in (12) limit the number of nodes in a solution.

Exact algorithms have been proposed in the literature for solving P1, P2 or similar problems (Di Summa et al., 2012; Veremyev et al., 2014b) but, to the best of our knowledge, they have never been extended to find the exact robustness of a network. Taking advantage of all the above definitions and results, the optimal robustness of a given network can be then obtained by solving a sequence of $P2(K)$ problems, with decreasing values of K.

As mentioned above, only those K values leading to LCCs of a minimum size $L_{\min}$ have practical interest. Consequently, the starting value of this sequence will be determined by solving first $P1(L_{\min})$. Note that, by using a decreasing sequence of K values, each solution provides a lower bound on the size of the LCC for the following ones, that allows to take advantage of Proposition 1. The procedure is depicted in Algorithm 1.

Algorithm 1 Computation of the optimal robustness

```
 1: Input $G = (N, E), L_{\min}$
 2: Initialize $L := L_{\min}, \ell := 1$
 3: Reduce $G$ using Proposition 1 $\longrightarrow (\bar{D}, E^{\bar{D}}), w = \{w_i\}_{i\in\bar{D}}$
 4: Solve $P1(L) \longrightarrow S_1(L)$
 5: Set $K := |S_1(L)|$, $K_{\max} := K$
 6: while $K > 0$ do
 7:     Solve $P2(K) \longrightarrow S_2(K), L^*_K$
 8:     Set $\ell := L^*_K$ and update $\bar{D}, E^{\bar{D}}, w$ accordingly
 9:     K:=K-1
10: Return $\{L^*_1, \dots, L^*_{K_{\max}}\}$
```

The input of the algorithm (line 1) is a graph $G = (N, E)$ and the smallest LCC size to be attained, $L_{\min}$; and the output is the sequence of sizes of the smallest LCCs attainable by removing $1, 2, \dots, K_{\max}$ nodes, where $K_{\max}$ is the minimum number of nodes that need to be removed to get a LCC of size at most $L_{\min}$.

The first problem solved is $P1(L_{\min})$. Since no lower bound on $|S_1(L_{\min})|$ is available, we start taking the trivial value $\ell = 1$. Before solving this first problem, the graph reduction provided by Proposition 1 is applied for this case (line 3). This yields the largest K value that needs to be evaluated. Starting from this value, the algorithm proceeds backwards, solving $P2(K)$ for a smaller K value at each iteration.

Note that L^*_K can be a very tight lower bound on $|S_2(K-1)|$, therefore, we can further reduce the graph at each iteration, taking advantage of Proposition 1 and this new bound (line 8).

3 Obtained results

We next analyze and compare the results of the previous methods among them and with the exact robustness. To this end, we show the analyses on four real networks and on a generated network that shows greater differences between exact and adaptive methods. The four real networks have different sizes in order to observe to what extent the observed results scale. The chosen networks were:

1. The Dolphins network presented in Lusseau et al. (2003), with 62 nodes and 159 edges,

2. the network *Les Miserables* as described in Knuth (1993) with 77 nodes and 254 edges,

3. the Southwest Airlines airport network (WN) Lordan et al. (2016) with 91 nodes and 580 edges, and

4. the co-authorships network built in Newman (2006), with 379 nodes and 914 edges.

In the cases where the original graph was not connected, the robustness of their LCC has been studied. For those graphs, the reported numbers of nodes and edges already correspond to this LCC. The corresponding results can be seen in Figures 2, 3, 4 and 5, respectively.

Last, the simulated network (with 90 nodes and 136 edges) has been generated adding randomly 1 and 2 edges in each time step according to the Barabasi-Albert model (Barabási, 1999) with $\gamma = 0.2$. Figure 6 shows the robustness results for the modeled (BA) network.

The results obtained on the *Dolphins* network show that, although the greedy algorithm based on the betweenness of the nodes is much more accurate than those based on the other criteria, there is still room for improvement, especially for central K values, where the smallest LCC sizes are still

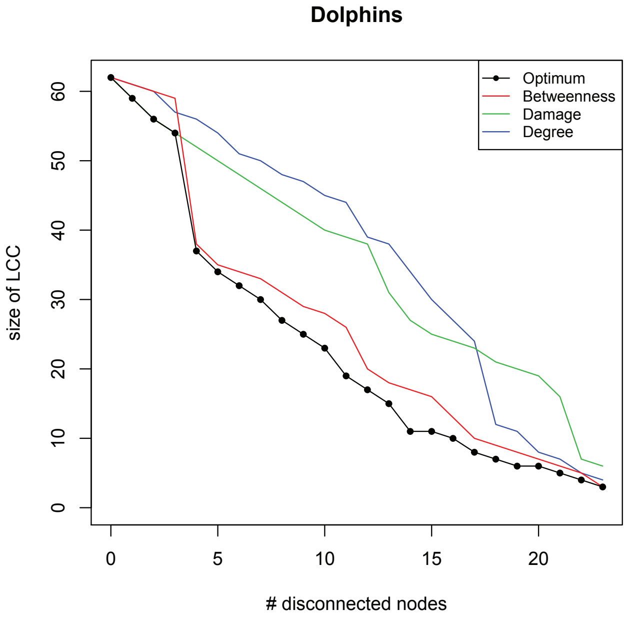

Figure 2: Exact and estimated robustness for the Dolphins network.

overestimated. These differences are quite relevant, given the small size of the graph.

Les Miserables network and the WN network –which was already found to be really robust in Lordan et al. (2016)– yield close results for different measures. The same thing seems to happen with the optimal results that are not far from the adaptive ones (see 3 and 4). However, if we look at the BA network (see 6), we can see that the results given by the adaptive strategies differ a lot from the optimal result, giving the wrong impression that the network is more robust than actually is. These large differences illustrate the need for an exact procedure for finding the optimal solutions.

We can observe now that in the case of these three networks, the greedy solution based on the node betweenness was quite accurate for much values of K, although in all cases there are K values where even this method (which, in general, outperforms the other two) overestimates the network robustness. Note that the use of the valid inequality from (9) was quite helpful in these cases thanks to the quality of the upper bound provided by the greedy methods.

Note also that we were able to compute the exact robustness of the co-authorships network, which is much larger than the other ones (379 nodes). This was attained, to a large extent, thanks to its particular structure, that allows to take advantage of Proposition 1.

Lastly, we want to analyze the accuracy of the greedy methodologies; *i.e.*

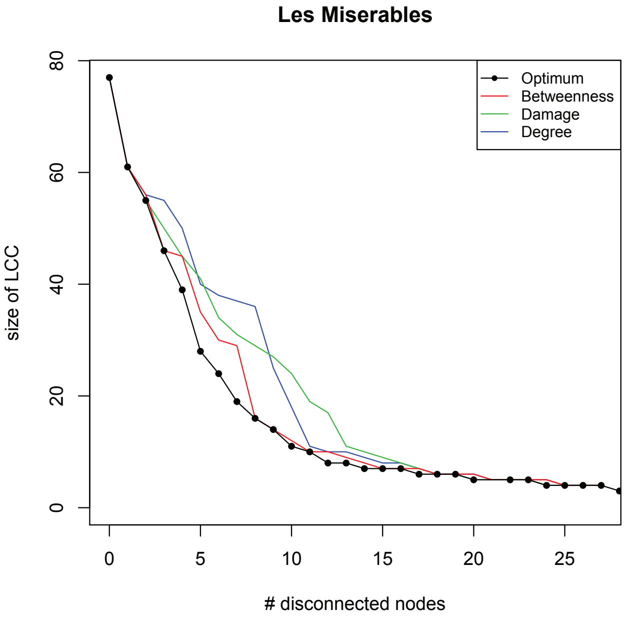

Figure 3: Exact and estimated robustness for the network *Les Miserables*.

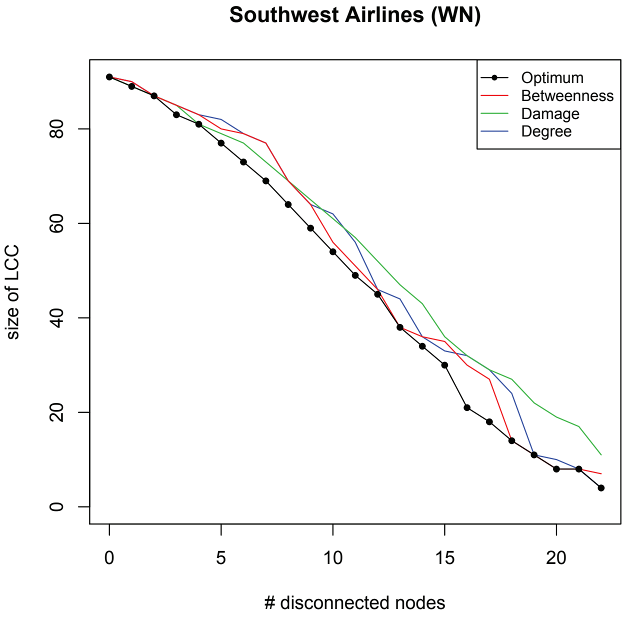

Figure 4: Exact and estimated robustness for the SW network.

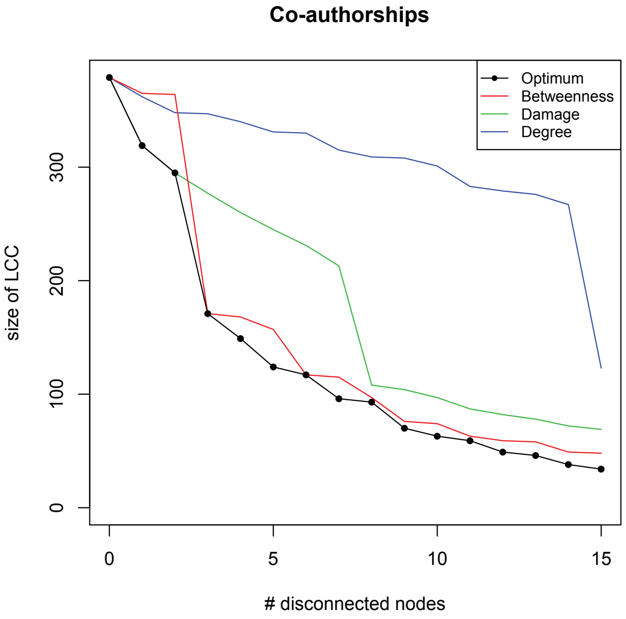

Figure 5: Exact and estimated robustness for the co-authorships network.

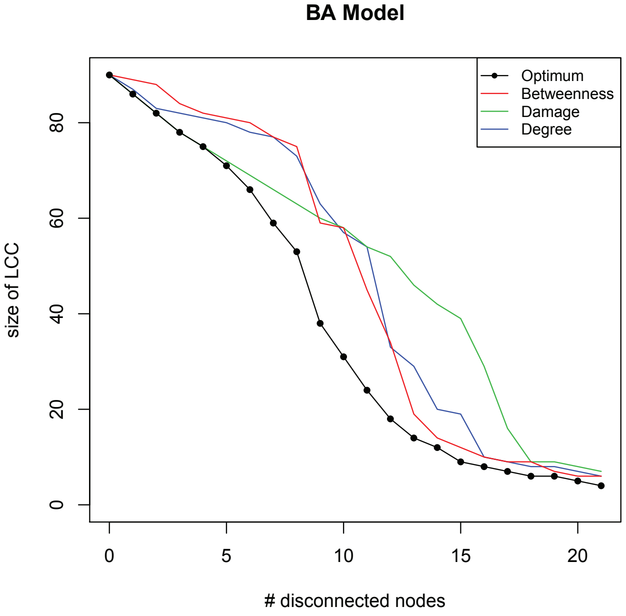

Figure 6: Exact and estimated robustness for the BA network.

we want to have a global comparison of the solutions they provide with the optimal solution. To this end, we will adapt the unique robustness measure defined in (Schneider et al., 2011). This measure requires to disconnect all nodes of the network. Since we want to analyze the robustness only to a given LCC size $L_{\min}$, we redefine the unique robustness measure as:

$$R = \frac{1}{|N|} \sum_{k=1}^{|S_1(L_{\min})|} \frac{L_k^*}{|N|} \tag{14}$$

where $|S_1(L_{\min})|$ is the minimum number of nodes that need to be interdicted to obtain a LCC of size $L_{\min}$, and $L_k^*/|N|$ represents the fraction of nodes in the LCC after removing k nodes. Using the normalization factor $1/|N|$ ensures that we can compare the robustness of different sized networks. As we want to compare the accuracy of different attacks with the optimal one, we define the accuracy of a method as:

$$acc_{meth} = 1 - \frac{R_{meth} - R_{opt}}{R_{max} - R_{opt}} \tag{15}$$

where R_{meth} and R_{opt} are the R estimate given by the analyzed method and the optimal R value, respectively. R_{max} is the upper bound on R associated with having a complete graph (for every disconnected node the size of the LCC decreases by exactly one). 1 shows the accuracy of the different methods on the five analyzed networks. Here we can see that, as already suggested by the above figures, among the tested greedy methods, the one based on the betweenness is the most effective one. Greedy removals based on damage and degree show different performances depending on the network, although damage gives us valuable information (usually giving the optimal result further from the first node disconnection) in the removal of small numbers of nodes, as shown in Figures 2-6. Note also that, despite the quality of the betweenness-greedy approach, it still provides rather poor accuracy values for some of the networks.

	BA	SW	Dolphins	Miser.	Co-auth.
Betw.	0.773	0.906	0.897	0.967	0.891
Damage	0.693	0.765	0.538	0.906	0.669
Degree	0.751	0.870	0.515	0.907	0.084

Table 1: Accuracy of the greedy methods on the five studied networks

No *simple* network characteristic such as density, average degree, average betweenness, etc. yielded a significant correlation with the accuracy of any of the greedy methods. To further illustrate this fact, we generated four more networks according to the Barabasi-Albert model with the same parameter setting as for the BA network used above (BA2,...,BA5). The number of edges of the obtained networks ranges between 121 and 139. As before, we applied the exact and the sequential methods to these four new networks. The sequential method accuracies for these networks are given in Table 2. This table shows that the accuracy of the sequential methods has a high variability even if the considered graphs have been generated with very similar structures. Therefore, we believe that there is no natural graph measure that can be used to predict the accuracy of greedy methods and decide when it is convenient to use the exact method instead.

	BA	BA2	BA3	BA4	BA5
Betw.	0.773	0.826	0.894	0.946	0.968
Damage	0.693	0.732	0.688	0.827	0.806
Degree	0.751	0.680	0.686	0.841	0.816

Table 2: Accuracy of the greedy methods on five BA networks

A factor that is clearly related with this accuracy is the relationship among the sets of critical nodes of different sizes. By construction, each of the sets obtained with a greedy method contains all the sets of smaller size, which does not necessarily happen with the optimal solutions. Indeed, if this was the case, the sequence of sets provided by the damage-based method would be optimal. Although being relevant, these differences between successive optimal sets of critical nodes could neither explain completely the different accuracies obtained for the different graphs. In any case, these differences cannot be used as a predictive tool to avoid applying the exact method when the sequential ones are accurate enough.

4 Conclusions

In this work we have presented a general methodology that allows computing the exact robustness of a graph and tested it on five graphs of different dimensions and with different structures. In the computational experiments, we also compared the exact robustness with the estimates obtained by some commonly used greedy adaptive methods based on alternative node characteristics (betweenness, damage, and degree). The results obtained on these graphs allow to draw the following conclusions.

On the one hand, despite some previous works exist concerning the optimal detection of critical nodes of a graph, this is the first time that the optimal robustness of networks is computed in the literature. This has allowed us to evaluate to what extent the adaptive methods used so far can overestimate the robustness of networks, and provides researchers with a tool to evaluate the effectiveness of new heuristics for this problem.

On the other hand, it is straightforward that the damage criterion provides the optimal solution when one single node is to be interdicted. However, our results show that, in practice, it also provides the optimal solutions to problems $P2(K)$ for small values of K other than one. Since for larger K values the greedy solutions obtained with the betweenness criterion are the best among the greedy ones, in situations where the exact robustness computation is not affordable, it becomes advisable to use methods that combine both, damage and betweenness criteria.

Acknowledgements

Authors acknowledge the support of the *Spanish Ministry of economy and competitiveness*, through Project MTM2015-63779-R

References

Albert, R., Albert, I., and Nakarado, G. (2004). Structural vulnerability of the North American power grid. *Physical Review E*, 69(2):1–4.

Albert, R. and Barabási, A.-L. (2002). Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74(1):47–97.

Barabási, A.-L. (1999). Emergence of Scaling in Random Networks. *Science*, 286(5439):509–512.

Broder, A., Kumar, R., Maghoul, F., Raghavan, P., Rajagopalan, S., Stata, R., Tomkins, A., and Wiener, J. (2000). Graph structure in the web. *Computer networks*, 33(1):309–320.

Cohen, R., Erez, K., Ben-Avraham, D., and Havlin, S. (2001). Breakdown of the Internet under Intentional Attack. *Physical Review Letters*, 86(16):3682–3685.

Deng, Y., Wu, J., and jin Tan, Y. (2016). Optimal attack strategy of complex networks based on tabu search. *Physica A: Statistical Mechanics and its Applications*, 442:74 – 81.

Di Summa, M., Grosso, A., and Locatelli, M. (2012). Branch and cut algorithms for detecting critical nodes in undirected graphs. *Computational Optimization and Applications*, 53(3):649–680.

Feng, F. and Wang, L. (2013). Robustness Measure of China's Railway Network Topology Using Relative Entropy. *Discrete Dynamics in Nature and Society*, 2013:1–8.

Jahanpour, E. and Chen, X. (2013). Analysis of complex network performance and heuristic node removal strategies. *Communications in Nonlinear Science and Numerical Simulation*, 18(12):3458–3468.

Knuth, D. E. (1993). *The Stanford GraphBase: a platform for combinatorial computing*, volume 37. Addison-Wesley Reading.

Latora, V. and Marchiori, M. (2005). Vulnerability and protection of infrastructure networks. *Physical Review E*, 71(1):015103.

Lordan, O., Sallan, J. M., Escorihuela, N., and Gonzalez-Prieto, D. (2016). Robustness of airline route networks. *Physica A: Statistical Mechanics and its Applications*, 445(1-3):18–26.

Lordan, O., Sallan, J. M., Simo, P., and Gonzalez-Prieto, D. (2014). Robustness of the air transport network. *Transportation Research Part E: Logistics and Transportation Review*, 68:155–163.

Lordan, O., Sallan, J. M., Simo, P., and Gonzalez-Prieto, D. (2015). Robustness of airline alliance route networks. *Communications in Nonlinear Science and Numerical Simulation*, 22(1-3):587–595.

Lusseau, D., Schneider, K., Boisseau, O. J., Haase, P., Slooten, E., and Dawson, S. M. (2003). The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405.

Newman, M. E. (2006). Finding community structure in networks using the eigenvectors of matrices. *Physical review E*, 74(3):036104.

Petreska, I., Tomovski, I., Gutierrez, E., Kocarev, L., Bono, F., and Poljansek, K. (2010). Application of modal analysis in assessing attack vulnerability of complex networks. *Communications in Nonlinear Science and Numerical Simulation*, 15(4):1008–1018.

Pullan, W. (2015). Heuristic identification of critical nodes in sparse real-world graphs. *Journal of Heuristics*, 21:577–598.

Schneider, C. M., Moreira, A. A., Andrade, J. S., Havlin, S., and Herrmann, H. J. (2011). Mitigation of malicious attacks on networks. *Proceedings of the National Academy of Sciences of the United States of America*, 108(10):3838–41.

Solé, R. V., Rosas-Casals, M., Corominas-Murtra, B., and Valverde, S. (2008). Robustness of the European power grids under intentional attack. *Physical Review E*, 77(2):1–7.

Soria, M., Lordan, O., and Sallan, J. M. (2017). Heuristics of node selection criteria to assess robustness of world airport network. *Chinese Journal of Aeronautics*, 30(4):1473–1480.

Veremyev, A., Boginski, V., and Pasiliao, E. (2014a). Exact identification of critical nodes in sparse networks via new compact formulations. *Optimization Letters*, 8:1245–1259.

Veremyev, A., Prokopyev, O. A., and Pasiliao, E. L. (2014b). An integer programming framework for critical elements detection in graphs. *Journal of Combinatorial Optimization*, 28(1):233–273.