FINAL REPORT

FAPESP - PROC. NO. 2012/25058-9

Adaptive mechanisms for fault tolerance management in complex network topologies

Researcher: Cinara Guellner Ghedini Hita

Supervisor: Carlos Henrique Costa Ribeiro

July, 2015

Adaptive mechanisms for fault tolerance management in complex network topologies

Researcher: Cinara Guellner Ghedini Hita

Supervisor: Carlos Henrique Costa Ribeiro

Final technical report proc. no. 2012/25058-9.

Date: 01/08/2013 to 30/07/2015.

São José dos Campos

July, 2015

Abstract

Applications based on self-organized agents are becoming pervasive all around, including in communication, monitoring, traffic and transportation systems. Their advantage is the possibility of providing services without the existence of a previous infrastructure. On the other hand, agents are prone to failure adding uncertainty and unpredictability in the environments in which they operate. So, a robust topology regarding failures is an imperative requirement. The robustness to failures means that, although some elements fail or become disconnected from the network, the dynamics and the main topological properties are maintained. In this way, this research main goal is to design mechanisms for both detecting and mitigating vulnerable topological configurations. This report summarizes the activities and achievements of a two-year project and presents the approaches that we had developed. The results demonstrate the feasibility of the proposed approaches regarding different context of applications.

Contents

1 Introduction

A complex network can be described as an abstract mathematical representation of a large number of elements that interact with each other and with the environment in order to organize themselves in an emerging structure [5]. Examples are commonplace in nature: ecological food webs [3], RNA structures in *Escherichia coli* bacteria [45] and neuronal topologies in *Caenorhabditis Elegans* worms [44]. They may also serve as models for both social [13,25] and technological networks, such as overlay, sensor and other communication networks [7,9,15,31].

Although complex network formations are targeted at achieving specific objectives and functionalities, they share organizational principles that result in similar topological properties, such as high global and local efficiencies and power-law degree distributions [1,33,41,44]. In spite of nuances deriving from topological characteristics of each network, these properties seem to be robust to random failures, in contrast to high sensitivity to failures of central elements (attacks). Results in the literature demonstrate that failures and (more significantly) attacks can degrade the evaluated properties [2, 11, 12, 28, 32].

An analysis considering an increasing fraction of removed nodes shows that even noticing that networks continue to exchange information in smaller networks, the overall efficiency is degraded. Therefore, the maintenance of nodes connected to the giant component will not only contribute to the network connectivity but also to maintain or enhance their efficiency. As the main goal for technological network applications is to ensure – - regardless of frequent changes in their topology — that their services will be available and efficiently delivered, it is necessary to provide such networks with mechanisms that maintain the network connectivity, as well as their main topological properties.

Despite the problem relevance, the detection and mitigation of vulnerable topological configuration are mostly disregarded. In the context of this work, a vulnerable state means that the network is potentially able to get disconnected if some node fail. Thus, the overall purpose of this research is to develop mechanisms that aim at improving the network robustness regarding its connectivity maintenance in case of failures.

We started with a previous approach based on the cluster coefficient property [20], that was

subject to further validation, including larger network models, additional real complex network datasets, and a performance confrontation with a random mechanism for vulnerability management. Section 4 details the validation model. The mechanisms proved to be valuable when confronted with random adjustments, but demonstrated sensitivity to the network topology. Despite some analysis, we could not find correlations between the topological properties of networks and the model performance.

For overcoming this feature, we came across with hierarchical structure that could be interesting to be evaluated. In this way, we developed a solution based on the concept of community structure, a well studied subject in the complex network area, but never applied to the failure tolerance problem. The model and results were presented in Section 5. This approach performance proved to be less sensitive to the network topology, and thus an interesting alternative to further investigation.

Meanwhile, we pursued a potential application where failure tolerance is an essential requirement. For that, we proposed in association with Unimore (University of Modena and Reggio Emilia), to address the robustness to failures in the context of multi-robot networks aiming at providing communication services for unstructured environments. More specifically, we address how to achieve a robust network concerning the connectivity maintenance. A robust network implies that, despite robot failures, most of its elements are still connected, being able to maintain a certain level of service. On the other hand, it is necessary to guarantee that the interconnection topology defines a connected graph, thus ensuring the possibility of exchanging data among all the nodes in the network.

Connectivity maintenance is a very well studied topic in the field of decentralized multi-robot systems, and several strategies can be found in the literature for solving this problem in different manners [35–38]. Despite that, the literature on connectivity maintenance, in general, does not consider robot failures and hence does not provide robust solutions in this respect. Robots are prone to failures due to hardware or communication issues. Based on the necessity of providing reliable solutions for mobile multi-robots systems and their intrinsical vulnerability to failures, we proposed to combine the mechanism to improve network robustness to failures [21] to the connectivity maintenance approach [35]. Section 6 details the application environment,

the model designed regarding the application features, the experimental setup and the results.

In addition, Section 2 presents the concepts that support the models evaluation such as complex network models, real network datasets and topological metrics. The problem relevance is emphasized in Section 3. Section 7 discusses our research main achievements and points out potential extensions and open issues. Finally, Section 8 discusses the goals set to this project and the activities carried out during the project.

Concerning the partial report submitted in 10/12/2014, we would like to thanks the reviewer for the comments and suggestions. This document were reviewed in order to address the issues surveyed, such as the number of iteration performed on the self-regenerating procedure, a detailed description of both the previous approach and the results of the comparative procedure.

The reviewer also stressed a considerable number of typos and grammar mistakes. We perfectly agree with him. Indeed, there are no excuses for such kind of occurrence. We would like to emphasize that it is not related to the fact of having paid little importance to the document, but with a combination of aspects, such as the arrangements to the BEPE Internships, a paper deadline, the necessity of documents translation from English to Portuguese, the latex library, etc. Of course, we could have asked for a deadline extension. Thus, we would like to sincerely apologize to the reviewer and to the FAPESP.

2 Preliminaries

A combination of models and metrics provides the benchmark for assessing the exposure of complex networks to failures and attacks, and for supporting a targeted analysis. This section outlines the concepts and models applied for carrying this analysis.

2.1 Complex network models

The interaction among agents in a complex system tends to create efficient networks at global and local levels, often under a scale-free degree distribution. Based on this, many researchers have proposed different models to create networks with particular topological properties as convenient simulations of real networks. For our study we consider two of the most widely used

models: the Watts and Strogatz (WS) model [43], the Barabasi and Albert's (BA) model [1] and the Klemm-Euguluz (KE) model [26], referred from this point on as WS networks, BA networks and KE networks, respectively.

Table 1 presents the models main topological properties values: the number of network nodes (n) and edges (|E|), the average degree ($\langle$k$\rangle$), and the global (E_{glob}) and local (E_{loc}) efficiencies - see Section 2.4 for technical details on how the efficiencies are computed.

Table 1: Topological properties of WS, BA and KE networks

Network	n	|E|	$\langle k \rangle$	E_{glob}	E_{loc}
BA	1000	5979	11.95	0.37	0.047
KE	1000	5973	11.94	0.30	0.60
WS	1000	6000	12	0.27	0.5

2.2 Real network datasets

Real datasets were considered for the experimental analysis. Some of them are classical benchmarks for studies in community-related approaches, the others are classical datasets in the complex networks literature, in general. The datasets and their main topological properties are shown in Table 2.

Table 2: Description and properties of real network datasets

Dataset	n	|E|	$\langle k \rangle$	E_{glob}	E_{loc}
The US heaviest traffic airports [40]	500	2980	11.92	0.37	0.62
The protein interaction of yeast [4]	417	511	2.45	0.19	0.05
American College football	115	613	10.6	0.45	0.40
Dolphins	62	159	5.13	0.37	0.26
Neural network of C. Elegans [43]	297	2345	15.79	0.40	0.30

2.3 Network modelling

A network is modeled as a graph $G = (N, E)$ defined by a set of nodes (or vertices) $N = 1, 2, ..., n$ and a set of links (or edges) $E \subseteq NXN$. A connection between vertices may be absent when there is no direct relationship or communication between them, or it may assume a value

in $[0, 1]$ representing the strength (weight) of the connection. Only undirected and unweighed networks are considered.

2.4 Evaluation Mechanisms

The assessment of the impact of failures and attacks on networks is supported by classical topological metrics related to the most important topological features found in real networks, as follows.

Global Efficiency Latora *et al.* [30] [29] introduced a measure of efficiency which computes how efficiently nodes exchange information either in a local or global scope, independently of whether the network is weighted or unweighted, connected or disconnected. For a given pair of nodes (i, j), its contribution to the global efficiency is inversely proportional to the shortest distance between them (d_{ij}), therefore $e_{ij} = \frac{1}{d_{ij}}$.

The global efficiency $\mathcal{E}_{glob}(G)$ of a graph G can then be defined as:

$$\frac{\sum\limits_{i \neq j \in G} e_{ij}}{n(n-1)} = \frac{1}{n(n-1)} \sum_{i \neq j \in G} \frac{1}{d_{ij}}, \tag{1}$$

and therefore, $\mathcal{E}_{glob}(G) \geq 0$. From this point on, we normalize this measure, considering the ideal situation G_{ideal} where all the possible $n(n-1)/2$ edges are in the graph, this is the case when $\mathcal{E}_{glob}$ assumes its maximum value. Thus, the normalized efficiency is:

$$E_{glob}(G) = \frac{\mathcal{E}_{glob}(G)}{\mathcal{E}_{glob}(G_{ideal})}. \tag{2}$$

Local Efficiency The local efficiency is defined as the ratio between the number of edges that actually exist among i's neighborhood (not including i itself) and the total number of possible links. If the nearest neighborhood of i is part of a clique, there are $k_i(k_i - 1)/2$ edges among the corresponding nodes, where k_i is the degree (number of links) of node i. Formally,

$$E_{loc}(G) = \frac{1}{n} \sum_{i \in G} E_{loc}(G_i), \tag{3}$$

where

$$E_{loc}(G_i) = \frac{1}{k_i(k_i-1)} \sum_{l \neq m \in G_i} \frac{1}{d_{lm}}. \tag{4}$$

and G_i is the subgraph induced by the nodes directly connected to i.

Algebraic connectivity Let $\mathcal{L} \in \mathbb{R}^{N \times N}$ be the Laplacian matrix of graph $\mathcal{G}$. As is well known from algebraic graph theory, this matrix defines relevant properties of $\mathcal{G}$ [24]. The algebraic connectivity λ of a graph is defined as the second smallest eigenvalue of the Laplacian matrix $\mathcal{L}$ [16].

For a connected graph, the algebraic connectivity is greater than zero ($\lambda > 0$), and it defines a lower bound for node connectivity and link connectivity [6].

Giant component A *connected component* of a graph is a set of nodes such that a path exists between any pair of nodes in this set. In most real-world complex networks, it has been observed that there is a large connected component together with a number of small components containing no more than a few percent of the nodes [10]. For very large networks, this component is generally referred to as *giant component*. With a slight abuse of notation, hereafter it is used the following definition even for small and medium size networks: The Giant Component of a graph is defined as its largest connected component [33].

The connectivity of a network $\mathcal{G}$ can be estimated by the relative size $S(\mathcal{G})$ of the giant component, given by the fraction of nodes in the network taking part in the largest connected component [33]:

$$S(\mathcal{G}) = \frac{n_{Giant}}{N}, \tag{5}$$

where n_{Giant} is the number of nodes in the giant component and N is the number of nodes in the network. In fact, for a connected graph, all the nodes belong to the same component, implying $n_{Giant} = N$, and then $S(\mathcal{G}) = 1$.

Even though algebraic connectivity is most commonly used in the multi-robot systems literature for assessing the connectedness of a graph, using the giant component exhibits a few

remarkable advantages. In particular, as soon as the graph becomes disconnected, the algebraic connectivity goes to zero: this happens even if one single node loses connectivity with the rest of the nodes. On the other hand, the size of the giant component gives a better insight regarding network fragmentation as a process: if a small number of nodes lose connectivity with the other ones, then the size of the giant component decreases slightly. Conversely, a huge decrease in the size of the giant component is observed when a large fraction of the nodes loses connectivity with the remaining ones.

Centrality Generally speaking, different nodes have different roles in maintaining connectivity of the overall network. In particular, the concept of *centrality* is generally exploited for identifying the most important nodes within a graph [27]. Several indicators can be found in the literature for defining centrality. In particular, referring to connectivity maintenance, we will exploit the concept of *Betweenness Centrality* (BC) [42], which establishes higher scores for nodes that are contained in most of the shortest paths between every pair of nodes in the network. In fact, nodes with this feature are likely to be crucial for maintaining the network functionality.

For a given node i and a pair of nodes j, l, the importance of i as a mediator of the communication between j and l can be established as the ratio between the number of shortest paths linking nodes j, l which pass through node i ($g_{jl}(i)$), and the total number of shortest paths connecting nodes j and l (g_{jl}). Then, the BC of a node i is simply the sum of this value over all pairs of nodes, not including i:

$$BC(i) = \sum_{j<l} \frac{g_{jl}(i)}{g_{jl}}. \tag{6}$$

Once the BC has been computed for all the nodes, it is possible to order them from the *most central* (i.e. the node with highest value of BC) to the *less central* (i.e. the node with lowest value of BC). Hence, let $[v_1, \ldots, v_N]$ be the list of nodes ordered by descending value of BC.

3 Problem statement

This section emphasizes the importance of mechanisms for detecting and mitigating harmful topological configurations. Consider as an example multi-robot system applications. For that, from a centralized point of view, a perturbation is introduced in randomly generated networks (parametrization details are described in Table 5 - Section 6.2.3). In particular, two perspectives are considered to produce such perturbations in the network:

1. Any robot can fail, and this can be simulated removing nodes from the network at random, with a uniform probability distribution.

2. The robot with the highest centrality fails.

The effect of removing the most central nodes can be quantitatively evaluated, for a given network, using the following procedure:

1. **network evaluation**: topological properties are computed: the algebraic connectivity (see Section 6.4) and the size of the giant component (eq. 5),

2. **nodes ranking**: the network nodes are ranked based on their Betweenness Centrality (BC) value (eq. 6),

3. **network perturbation**: the most central node is removed from the network based on the centrality ranking position.

As an example, consider a 20 robots network (Figure 1). Looking at the illustration on the left, is it possible to identify some key elements for maintaining the network connectivity. Focusing on the first two positions regarding BC ranking (red circles), nodes 7 and 19, respectively, if any of them fail the resulting network will be fragmented.

Similar characteristics can be found in larger networks as well. Consider the example depicted in Figure 2, which represents a 100 nodes network. Once again, this network exhibits some poor connectivity spots, as highlighted. Failures of nodes 41 and 11 or 74 and 11 disconnect the network, which prevents the multi-robot system from achieving cooperative objectives.

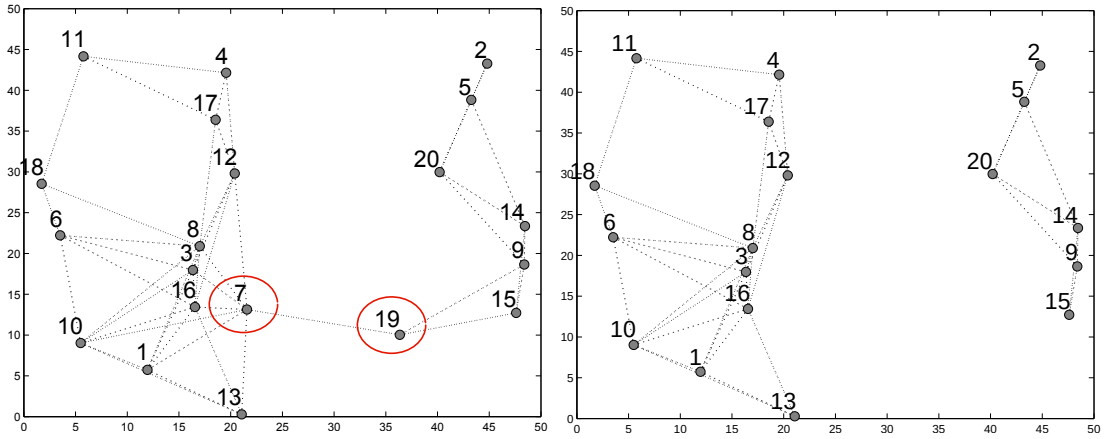


Figure 1: Initial network configuration (left) and network configuration after the two first highest BC nodes removal (right).

Even a single failure of any of them harshly affects the network connectivity. In this example, these nodes are also the first ones in the BC ranking, demonstrating the centrality measure performance for assessing the impact of robot failures. Notice that in both cases, the initial networks are connected but still in a vulnerable state.

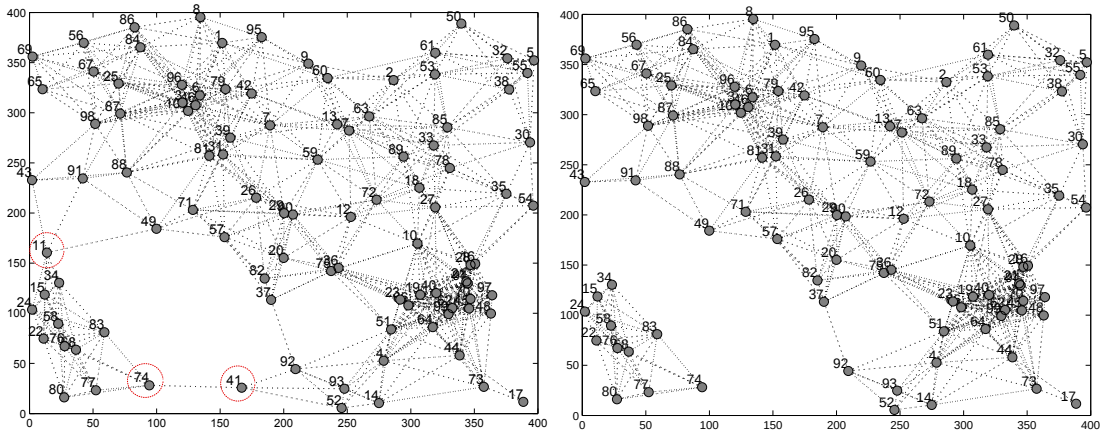


Figure 2: Initial network configuration (left) and network configuration after the three first highest BC nodes removal

Figure 3 illustrates two networks with different values for the algebraic connectivity: $\lambda = 0.83$ on the left side and $\lambda = 1$ on the right side. It is clear that the network with the larger algebraic connectivity will be more affected if the central node fails. Thus, mechanisms for algebraic connectivity maintenance or improvement do not guarantee a robust topology.

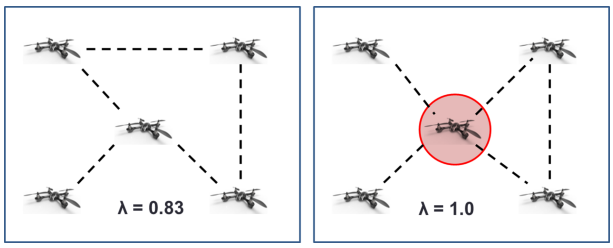


Figure 3: Algebraic connectivity

Figure 4 shows the algebraic connectivity (on the left) and the giant component (on the right) evolution as a result of the network perturbations (node failures). The horizontal axis represents the fraction of nodes removed from networks, and the vertical axis the property value. Green lines represent the $N = 20$ case while blue lines represent the $N = 100$ case. The same experiments were repeated removing the same number of nodes, but randomly chosen (dotted lines). The results demonstrate that the connectivity of both networks was considerably affected by successive losses of the highest BC nodes.

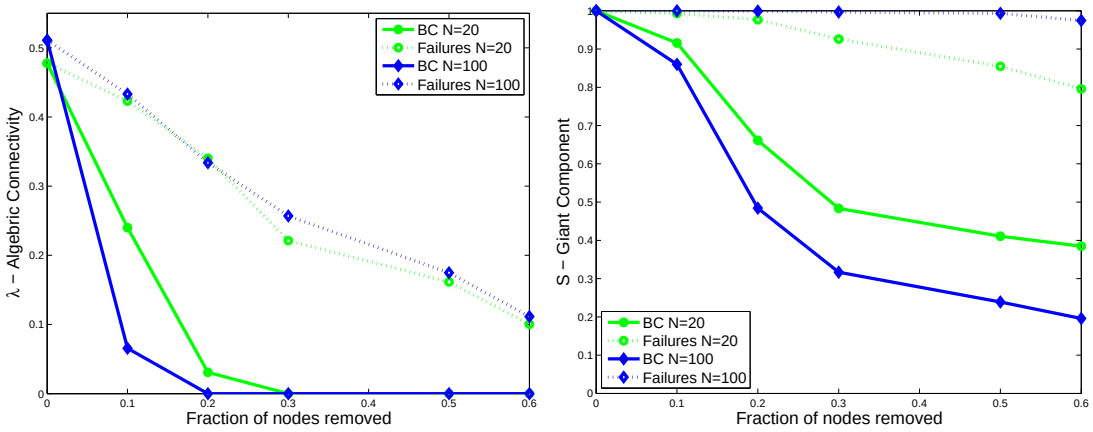


Figure 4: The algebraic connectivity (on the left) and the size of the giant component (on the right).

To summarize, connectivity is a crucial requirement in decentralized multi-robot systems: to achieve a common objective, robots may need to exchange information. On the other hand, robots are likely to fail. Besides, for any decentralized communication network the robustness to failures and attacks is a real concern.

4 Cluster Coefficient-based model for vulnerability management

Complex networks are supposed to be self-organized and self-configurable, therefore, mechanisms based on centralized control are unfeasible. In this sense, nodes themselves are the most appropriate components for controlling the necessary information and procedures to manage vulnerability. Defining nodes as the main agents in this process means that the concern is to realize which information is meaningful to locally support the mechanisms responsible for evaluating and promoting changes in the network topology. For this, we assume that specific nodes are most likely to affect the network connectivity, and thereby, a node can be defined as harmful according to some requirements to achieve such likelihood status.

A function which represents the level of harmfulness of a node i with respect to the network property focus of analysis on a specific time t can be defined by:

$$Ph_{i,t} = f(i_t, N_{i,t}), \tag{7}$$

where i_t and $N_{i,t}$ represents respectively the information about node i and i neighborhood on a specific time t. To properly determine if a node should be classified as harmful, a parameter γ was introduced. If $Ph_{i,t} \geq \gamma$, node i is classified as harmful ($h_{i,t} = 1$), otherwise it is not considered to be harmful ($h_{i,t} = 0$):

$$h_{i,t} = \begin{cases} 1 & \text{if } Ph_{i,t} \geq \gamma \\ 0 & \text{if } Ph_{i,t} < \gamma \end{cases} \tag{8}$$

If a node is aware of its own and of its neighborhood harmfulness status, it can estimate its state of vulnerability, in other words, if it is in a vulnerable state [18]. We then define that a node can be considered to be in a vulnerable state, if itself or its neighbors are harmful to the network connectivity. Therefore two states are defined: vulnerable ($V_{i,t} = 1$) and not vulnerable ($V_{i,t} = 0$), according to:

$$V_{i,t} = \begin{cases} 1 & \text{if } h_{i,t} = 1 \text{ or } Q_{i,t} \geq \beta \text{ or } k_i \leq 1 \\ 0 & \text{if } h_{i,t} = 0 \text{ and } Q_{i,t} < \beta \end{cases}$$

where

$$Q_{i,t} = \frac{\sum_{j \in N_i}(h_{i,t})}{k_i}. \tag{9}$$

$Q_{i,t}$ represents the fraction of harmful nodes in the i neighborhood at a specific time t. The threshold to set a node as vulnerable is given by the parameter β. The vulnerability condition can be used to trigger procedures to try to reverse or minimize adverse settings.

4.1 Adaptive mechanism

The adaptive mechanisms are responsible for promoting changes when they are relevant to improve the network configuration concerning network connectivity, and, in consequence, global efficiency. The first assumption adopted to define such mechanisms is that each node must be able to *locally* manage and keep the required information to evaluate its neighbors harmfulness, parameter values, and policies from communication protocols.

Regarding the evaluation of node harmfulness, analytical measures must be based on the information of subgraphs around the nodes. In such a case, the first natural option is the local efficiency, under the premise that nodes with poor local efficiency can be harmful to the overall network connectivity if they fail. Thus, the higher the local efficiency of a node i, the less likely is its potential danger to the network connectivity, in case of failure. As $E_{loc}(G_i)$ ranges in $[0, 1]$, the degree of harmfulness of a node i is given by:

$$Ph_{i,t} = 1 - E_{loc}(i, t), \tag{10}$$

where $E_{loc}(i, t)$ represents the local efficiency of node i ($eq.$ 4) at time t. As $E_{loc}(i, t)$ captures the configuration in a node neighborhood, it can determine if a node is in a vulnerable state without any reference to the proportion of harmful nodes in the neighborhood ($Q_{i,t}$). Accordingly, we set $V_{i,t} = Ph_{i,t}$.

For the adaptation process two scenarios were explored: a) maintenance of the number of

links in the network (for each new link created, another link must be removed), and b) creation of new links (for each new link created, no other link is removed). For both cases, the vulnerability assessment and the search for a new connection are performed. In compliance with the attacks and failures protocol, after each node removal, every node i assesses if it is in a vulnerability state. If applicable (*i.e*, when $Ph_{i,t} \geq \gamma$), node i triggers the process for searching another non-vulnerable node j ($Ph_{j,t} < \gamma$) at a distance greater or equal to d, to request a connection. If such node j is found, a new link between nodes i and j can be established.

With regard to the first scenario, node j must choose one of its neighbors r to disconnect. Several strategies can be considered for choosing a node for disconnection, but considering that nodes with high degree tend to be less affected by the loss of a connection, they are set as more likely to lose it. The strategy used to explore the network to find candidates for new connections is based on a depth-first search (DFS) [23]. In this technique, a search starts from a initial node i and explores the network by deepening the search branch until it finds another node j with the desired features (candidate node), the search space is completely explored, or a stop condition is reached.

The following pseudo-code (Figure 5) presents the procedure described above. The adaptation mechanism evaluates if each node of network G is in a vulnerable state. For each one in such condition, a search is started (find_connection(G, i, i, d)). If a node is found, the node for disconnection is defined (find_disconnection(G, j)). Then, a link between j and r is removed and a new link between j and i is created.

```
adaptation(G)
for all i ∈ G
  if V_{i,t} % node i is vulnerable at time t
    [list, j] ← find_connection(G, i, i, d)
    if empty(list)
      r ← find_disconnection(G, j)
      if r ≠ −1
        add_link(G, i, j)
        remove_link(G, j, r)
      else
        print 'Adaptation failed'
      end
    else
      print 'Fail to find a new connection'
    end
  end
end
```

```
find_disconnection(G, j)
sumDegree ← 0
for all v ∈ N_j do
  sumDegree ← sumDegree + k_v
  node ← −1
  while
    for each v ∈ N_j
      if k_v/sumDegree ≥ rand()
        node = v
        return node
      end
    end
  end
end
```

```
find_connection(G, i, list, d)
if ∼ empty(list)
  reference ← first(list)
  if ∼ V_{reference,t} and dist((g, i, reference) >= d)
    list = []
    return list, reference
  else
    candidate ← find_candidate(G, reference)
    if ∼ empty(candidate)
      list = [candidate, list]
      [list, candidate]=find_connection(G, i, candidate, d)
    end
  end
end
```

```
find_candidate(G, i)
candidate=[]
for all v ∈ N_i do
  if v is not visited
    candidate ← [candidate v]
  end
end
```

Figure 5: Pseudo-code of the adaptation process.

4.2 Results

The results illustrate the initial scores for the properties evaluated (y axis) and their variations when exposed to continuous failures/attacks (fraction of nodes removed – x axis). The solid line represents the behavior of the original networks (G) without any adaptation process, $Hnew$ illustrates the effect of the adaptation process in the case of creating new connections, and H the

performance from keeping the number of links in the network. The parameters are $\gamma = 0.5$ and $d = 2$.

Figures from 6 to 11 show the results focusing on those cases in which the impact of attacks and failures was more significant or relevant to demonstrate the network sensitivity and the adaptation mechanism performance.

BA networks exhibit a very poor local efficiency ($E_{loc} = 0.047$). So, the majority of the nodes nodes are in a vulnerable state, in contrast of a few non-vulnerable nodes to accomplish new connections, resulting in a poor performance for both adaptation strategies at the beginning of the attacks process. On the other hand, as the fraction of removed nodes increases ($f \geq$0.25), a performance improvement for the $Hnew$ strategy is achieved for global and local efficiencies and the size of the giant component. This demonstrates that, over time, few adjustments may result in some performance improvement. Notice that at this point ($f \approx 0.25$) is when the BA networks begin to become partitioned.

Although the γ parameter could be set to values below 0.5, the fraction of nodes with $E_{loc} > 0$ is only about 0.06 in the initial network configuration. As previously mentioned, despite the degree distribution feature and a better performance for local efficiency than the classical random model of Erdös-Rényi [14], the BA model cannot produce the high local efficiency observed in many real networks.

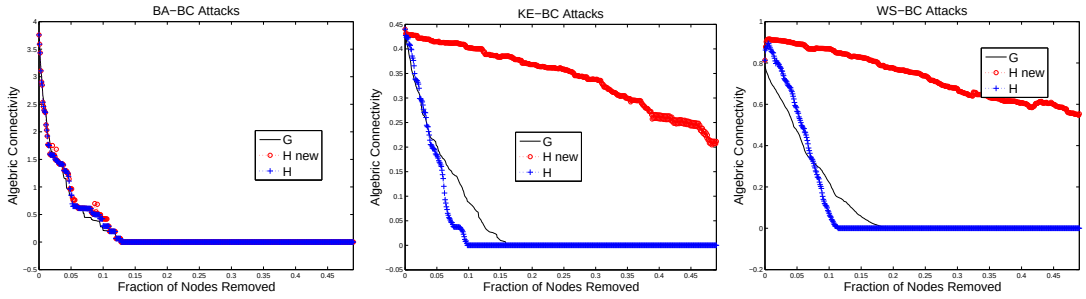


Figure 6: Performance of the adaptation process regarding algebraic connectivity (λ) — BC attacks in BA, KE and WS networks ($\gamma = 0.5$ and $d = 2$).

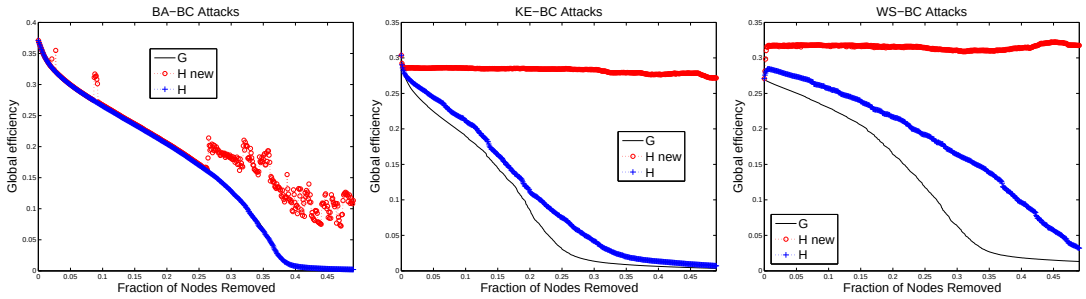


Figure 7: Performance of the adaptation process regarding global efficiency (E_{glob}) — BC attacks in BA,KE and WS networks ($\gamma = 0.5$ and $d = 2$).

For BC attacks in KE and WS networks, in cases where the number of connections was maintained, the values of global efficiency had some improvement, in contrast, the local efficiency was penalized. This happens because, in most cases, connections are rewired to another neighborhood, lowering the local efficiency. This is tantamount to creating shortcuts in the network by linking nodes in different neighborhoods: it rewards global properties, but penalizes local ones. On the other hand, the giant component endured the effects of attacks after $f > 0.2$, reinforcing the importance of continued adjustments in the topology of systems facing continuous attacks and failures.

On the other hand, creating new connections does not penalize the values for local efficiency because connections are wired between a vulnerable and a non-vulnerable neighborhood, *i.e.*, nodes with a poor local connectivity can create a new link with a node with a high local connectivity. Thus, when it is possible to create connections, the values of the properties evaluated are remarkably enhanced considering the potential harmfulness of targeted attacks and, in most cases they remain almost the same as at the beginning of the perturbation process (*e.g.* global and local efficiencies and the size of the giant component for both KE and WS networks).

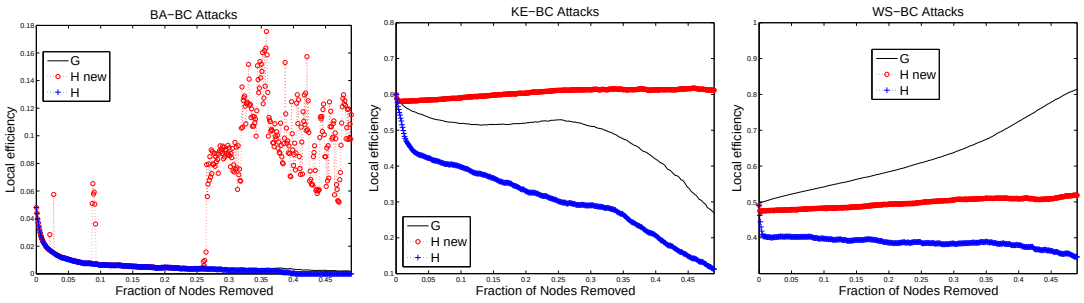


Figure 8: Performance of the adaptation process regarding local efficiency (E_{loc}) — BC attacks in BA,KE and WS networks ($\gamma = 0.5$ and $d = 2$).

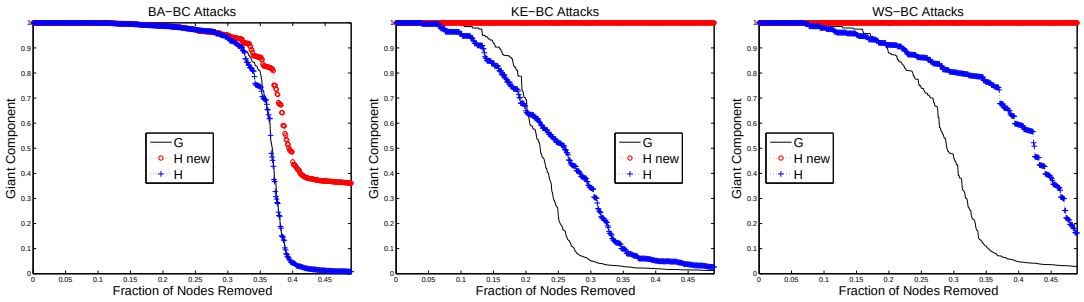


Figure 9: Performance of the adaptation process regarding giant component (S) — BC attacks in — BC attacks in BA,KE and WS networks ($\gamma = 0.5$ and $d = 2$).

It is important to highlight the results for failures. Although they are not as harmful to network operation as attacks, successive failures may also produce a considerable impact. For KE (Figure 10) and WS (not shown) networks, the property values were maintained or smoothly increased. In the case of BA networks, the values of all properties assessed were significantly improved, mainly for local efficiency and algebraic connectivity (Figure 11), which may also be useful for enhancing network robustness.

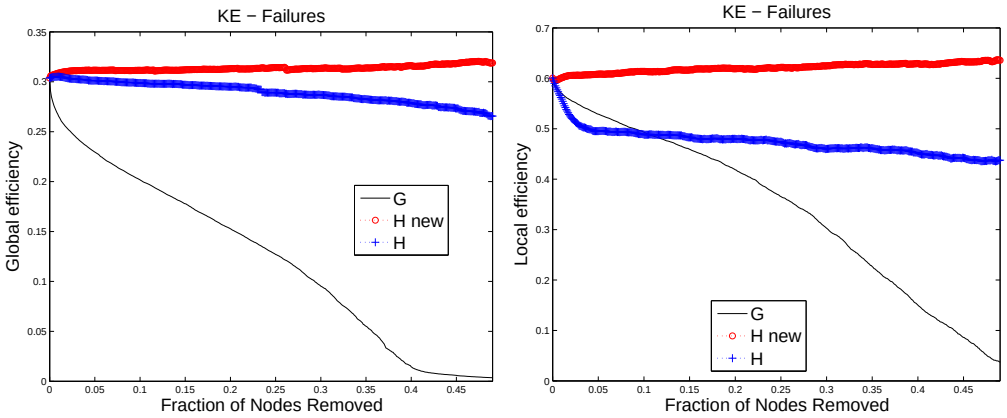


Figure 10: Performance of the adaptation process regarding global (E_{glob}) and local (E_{loc}) efficiencies — Failures in KE networks ($\gamma = 0.5$ and $d = 2$).

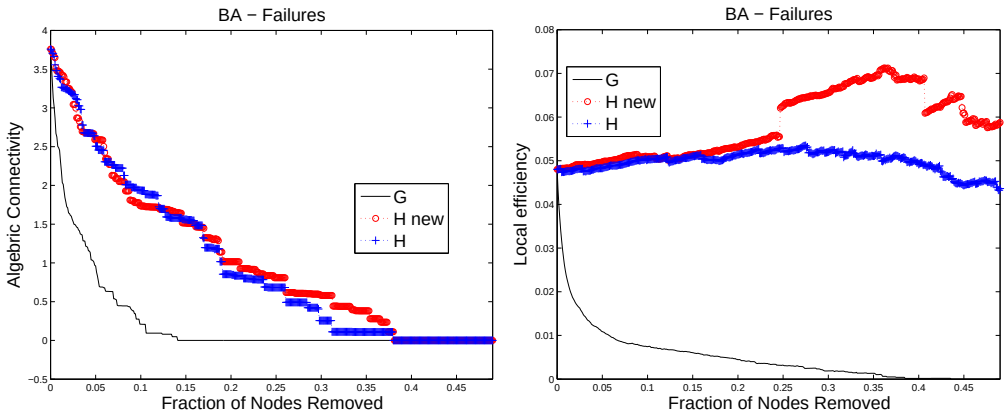


Figure 11: Performance of the adaptation process regarding local efficiency (E_{loc}) and algebraic connectivity — Failures in BA networks ($\gamma = 0.5$ and $d = 2$).

For the protein interaction network, the adaptation process considering new connections ($Hnew$) was able to increase somewhat the value of the local efficiency in the beginning of the process and maintain it until almost 20% of nodes were removed from the network (Figure 12). Regarding the global efficiency, a small improvement over the original network was evident. Notice that both global and local efficiencies scores are low.

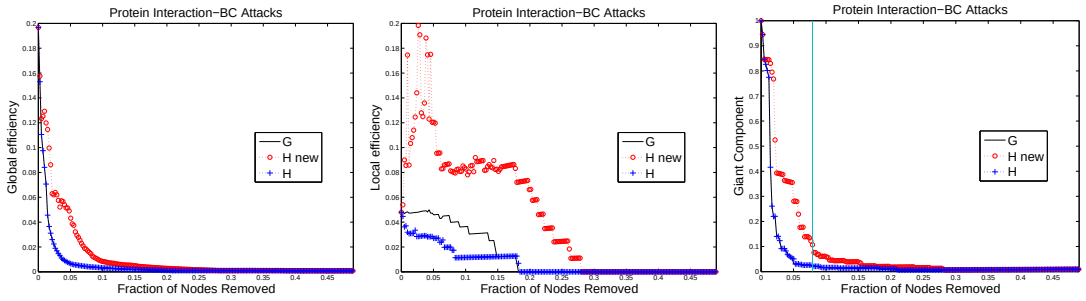


Figure 12: Values of global (E_{glob}) and local (E_{loc}) efficiencies and the giant component (S) for BC attacks - Protein Interaction network.

The results for the US transportation network show that with almost 10% of nodes dropped by BC attacks, the values of global and local efficiencies were kept - for the latter with an initial improvement (Figure 13). After this point, these properties were more affected, although without the adaptation process, at this point the global efficiency reached extremely low values.

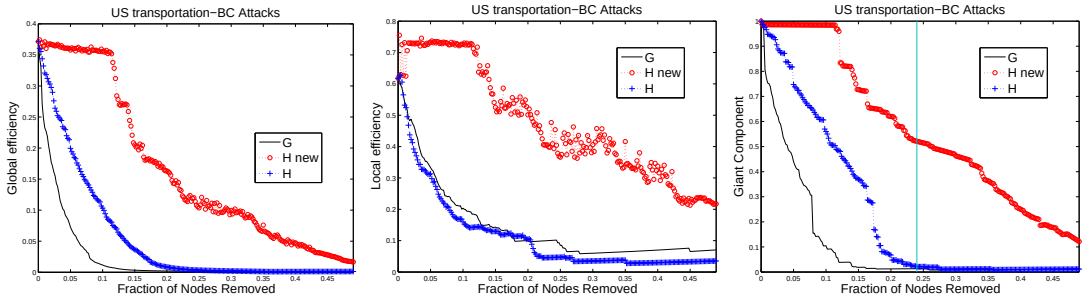


Figure 13: Values of global (E_{glob}) and local (E_{loc}) efficiencies and the giant component (S) for BC attacks - US transportation network.

The same pattern emerges from the adaptive process in the C. Elegans network. Remarkably, despite the relative size of the giant component decreasing more smoothly, the local and global efficiencies exhibit some peaks around 17% and between 27% and 33% of nodes removed from the network. Notably, the local efficiency decreases steeply, reaching a performance equal to or even lower than the original network G (Figure 14). Further investigations are necessary to understand such behavior.

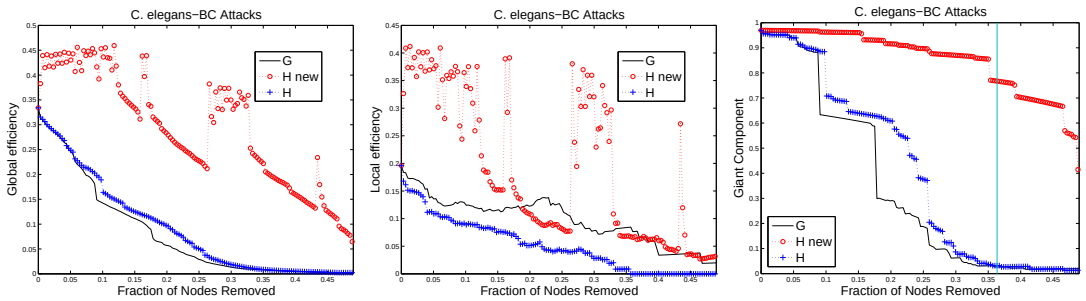


Figure 14: Values of global (E_{glob}) and local (E_{loc}) efficiencies and the giant component (S) for BC attacks - Celegans network.

Figure 15 illustrates that for failures, the local efficiency also significantly enhances, as for the artificial models (see Figure 11).

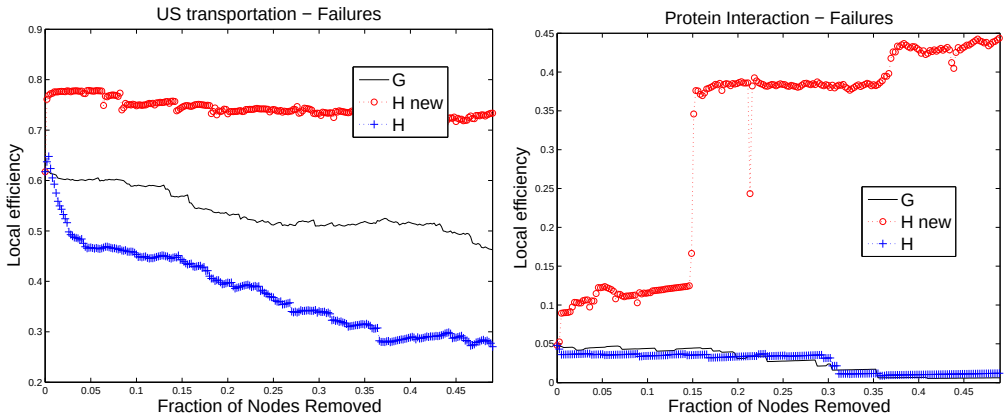


Figure 15: Performance of the adaptation process regarding local efficiency for failures - US Transportation and Protein Interaction networks.

4.3 Comparison with a self-regenerating network

We finish this section on results considering a setting where the network under consideration does have a random self-regenerating mechanism. The comparison to be considered is against the vulnerability-based adaptation mechanism with a regenerating capability ($Hnew$).

The critical point is to define a fair protocol to drive such comparative analysis, considering questions such as how many links should be created in a completely random process or which criteria should be adopted to define the distance between two nodes to be connected. If these parameters (e.g. number of new links and distance) are not well stated, the random process

performance can be over or underestimated. Note that, in the vulnerability-based adaptation model ($Hnew$), both are defined by the process itself.

Taking into account the above issues, we set the following protocol: the original network G undergoes successive BC attacks, without any active adaptive mechanism. The fraction of nodes disconnected from the G network was set to 0.05. The resulting G' network is input to a parallel adaptation process that generates two new networks: $Hnew$, which is a result of one iteration of the same process defined in Section 4.1 with $\gamma = 0.5$ and $d = 2$, and a network resulting from an average of five realizations of the random self-regenerating mechanism. For the latter, two strategies were implemented. The first, namely $Hrandom$, creates exactly the same number of connections (η) between nodes at the same distance that those added to $Hnew$, but the nodes are picked up at random, no matter their status of vulnerability and considering only the distance between them. In this case, the number of nodes and the distance are given by the vulnerability-based adaptation process. For the second strategy ($Hrandom'$), the number of new connections is given by a random value ranging between 1 and the number of connections established for $Hnew$ [1..η]. A pair of nodes is chosen for each new connection, regardless of the distance between them.

Figures 16 and 17 illustrate the average property values for G, G', $Hnew$, $Hrandom$ and $Hrandom'$ for KE networks with $N = 1000$ (see table 1), and $N = 5000$ ($\mid E \mid$ = 33965, $\langle k \rangle$ = 13.58, E_{glob} = 0.27, E_{loc} = 0.56, D=6.8 and λ = 0.54), respectively.

The results demonstrate that, for all scenarios, $Hnew$ performs better for global efficiency and maintains the value for the local efficiency. In turn, the algebraic connectivity exhibits an improvement for $Hrandom'$. Table 4.3 shows the average for the number of new connections and the previous shortest distance between the nodes which established these connections. The number of adaptations for $Hnew/Hrandom$ is almost twice than for $Hrandom'$. On the other hand, the average distance is more than doubled in the latter, and this is probably the reason for the increase of the algebraic connectivity. It is important to notice that this analysis involves a single iteration, and that the distance tends to be higher at the beginning of the process and to decrease to the parameter value as iterations take place, although the distance usually does not add more than a single hop to the value set for d.

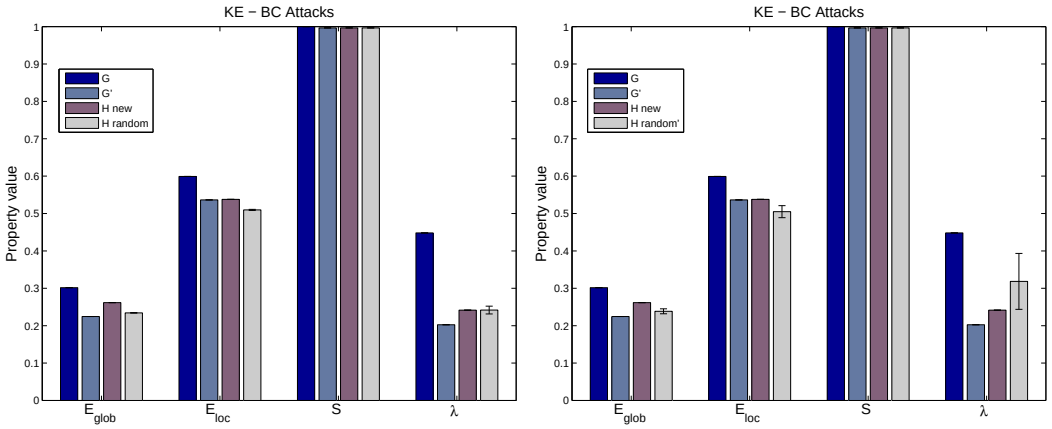


Figure 16: Comparative performance of $Hnew$ with $Hrandom$ (left) and $Hrandom'$ (right) for global (E_{glob}) and local (E_{loc}) efficiencies, the algebraic connectivity (λ), and the giant component (S), considering BC attacks in KE networks (N=1000, $\gamma = 0.5$, $d = 2$).

Table 3: Average distance and number of adaptations

		$Hnew/Hrandom$		$Hrandom'$	
Network	n	distance	adaptation	distance	adaptation
KE	1000	2.23	208.1	4.74	107.5
KE	5000	2.10	1045.6	4.66	564.92

For real network topologies, the vulnerability-based adaptation model performed better than the random self-regenerating mechanism (see Figures 18 and 19). For the C.Elegans network, the vulnerability-based adaptation model was able to recover the global efficiency to its initial value and to improve the local efficiency. The algebraic connectivity remained with very low values in all cases.

5 Community-based model for vulnerability management

This section explores the assumption that community structure can be worthwhile to support the evaluation and mitigation of vulnerable topological states. Consider that networks target of analysis have their nodes classified according to the community they belong. The approach to find and update communities adopted in this work is presented in Section 4.

For assessing the relevance of community structure information to detect and mitigate vulnerable topological network configurations, a protocol for attacks and failures concerning the

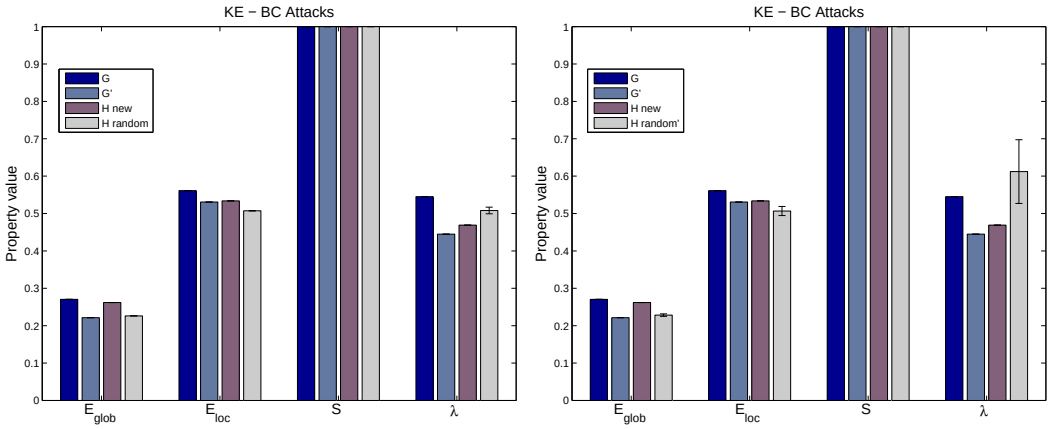


Figure 17: Comparative performance of $Hnew$ with $Hrandom$ (left) and $Hrandom'$ (right) for global (E_{glob}) and local (E_{loc}) efficiencies, the algebraic connectivity (λ), and the giant component (S), considering BC attacks in KE networks (N=5000, $\gamma = 0.5$, $d = 2$).

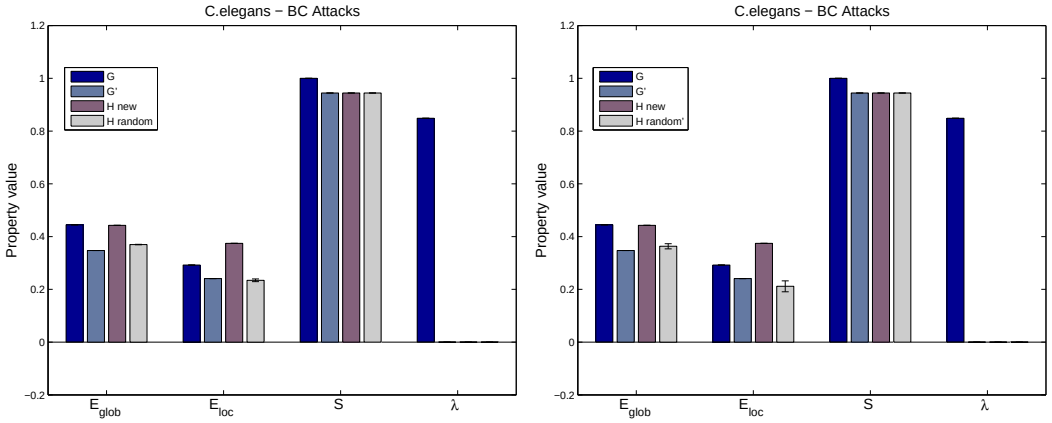


Figure 18: Comparative performance of $Hnew$ with $Hrandom$ (left) and $Hrandom'$ (right) for global (E_{glob}) and local (E_{loc}) efficiencies, the algebraic connectivity (λ), and the giant component (S), considering BC attacks in C.Elegans network - ($\gamma = 0.5$, d=2).

role of central nodes in the community structures were applied. For simulating failures, nodes are considered autonomous agents that can leave the network at random with a uniform probability distribution. On the other hand, to reproduce a possible scenario of attacks, the Betweenness Centrality ranking was used to define the order that nodes must be removed from the network (see eq. 6).

The failure and attack protocol encompasses: 1) ranking nodes according to BC or random criteria; 2) removing links of the most central node from the network considering their role in the node community: inside, outside or both; and 3) computing the target properties values. At each

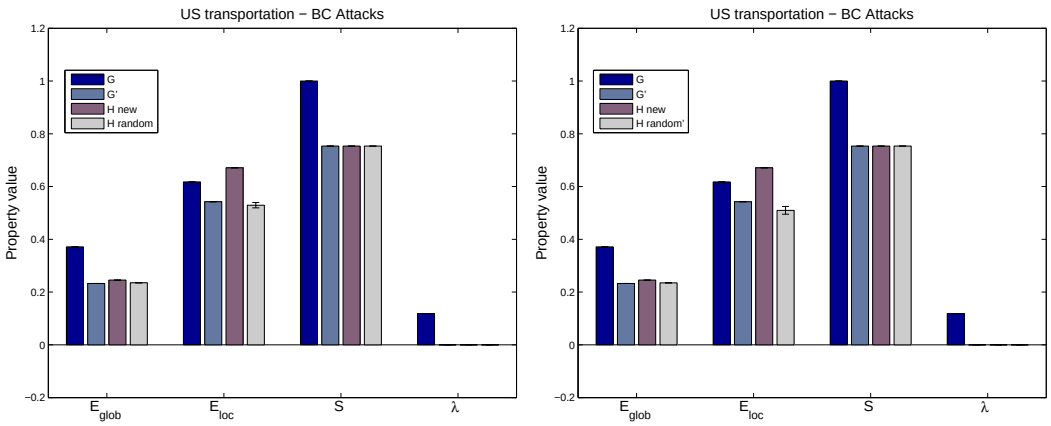


Figure 19: Comparative performance of $Hnew$ with $Hrandom$ (left) and $Hrandom'$ (right) for global (E_{glob}) and local (E_{loc}) efficiencies, the algebraic connectivity (λ), and the giant component (S), considering BC-attacks in US transportation network - ($\gamma = 0.5$, $d = 2$).

iteration, the node ranking is updated until a previously defined fraction (f) of nodes become disconnected from the network.

The adaptive mechanism must compensate the central node failures with addition of new links. Thus, for its performance evaluation the most central nodes are completely removed from the network. For validation purposes, three heuristics were defined considering the constraints of creating new links according to their roles: *inside* or *outside* the community, or *both*. For model-based networks, the results were averaged over five realizations.

For assessing the role that central elements play in the community structure concerning robustness to failures and attacks, the protocol for link removals was applied. The results are depicted using blue, red, and green lines representing the removal of node's link(s) according to inside, outside and both (inside and outside) criteria, respectively.

Figures 20 to 23 show the evolution of global efficiency and the giant component (y-axis) during the process of attacks, represented by the fraction of nodes removed from the network (x-axis). The results stress the importance of links between communities, emphasizing that losing channels of communication between communities may be potentially harmful to the network connectivity. It means that those nodes responsible for linking communities may be the key elements for evaluating and mitigating topological states of vulnerability.

Figure 24 illustrates the classification of links at each network state during the perturbation

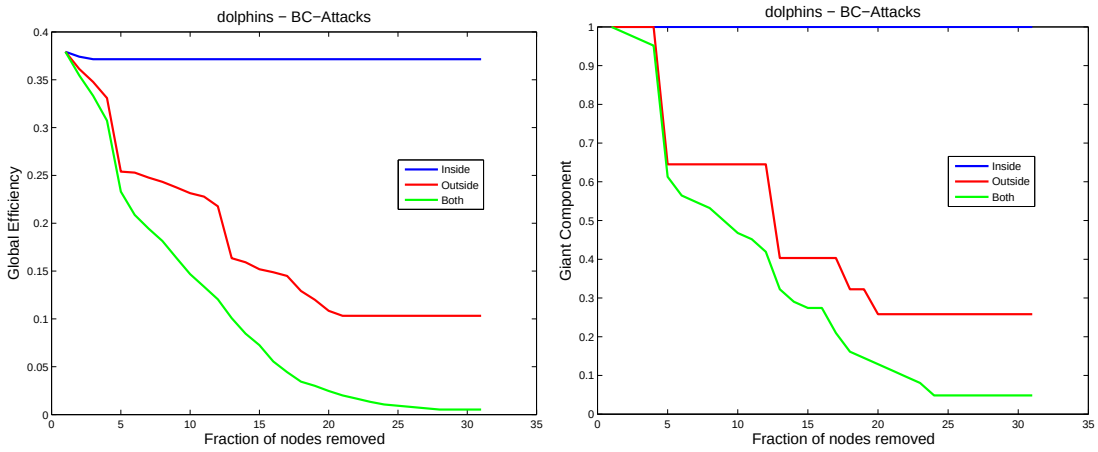


Figure 20: Global efficiency and giant component – BC attacks for dolphins network.

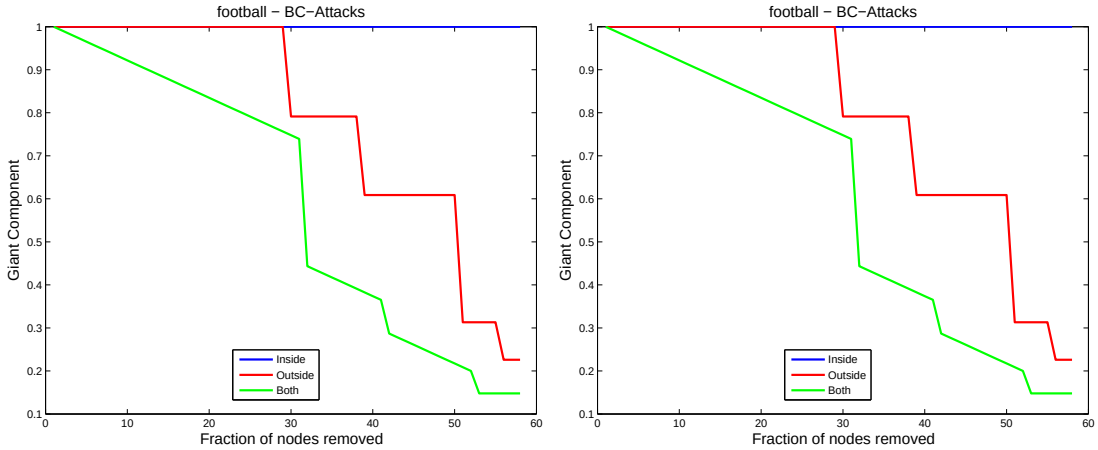


Figure 21: Local efficiency and giant component — BC attacks for football network.

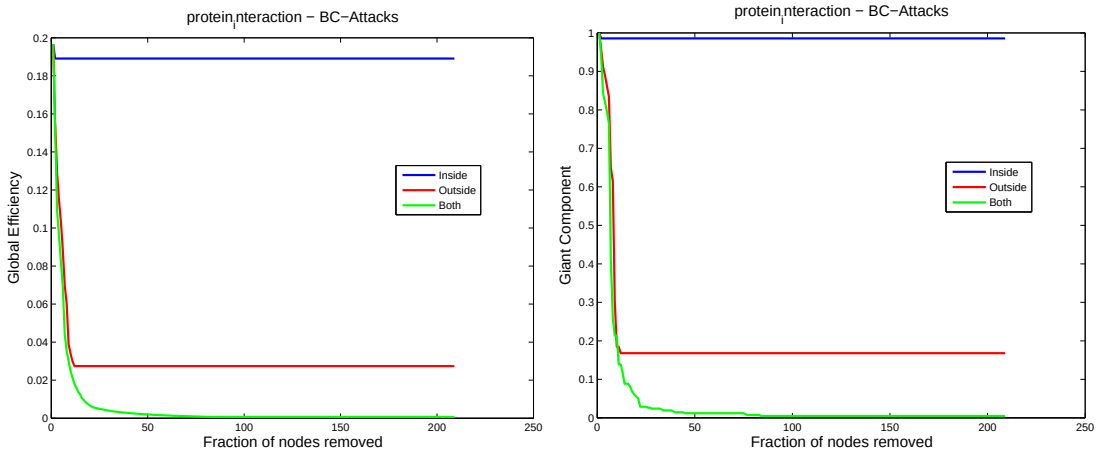


Figure 22: Global efficiency and giant component — BC attacks for Protein Interaction network.

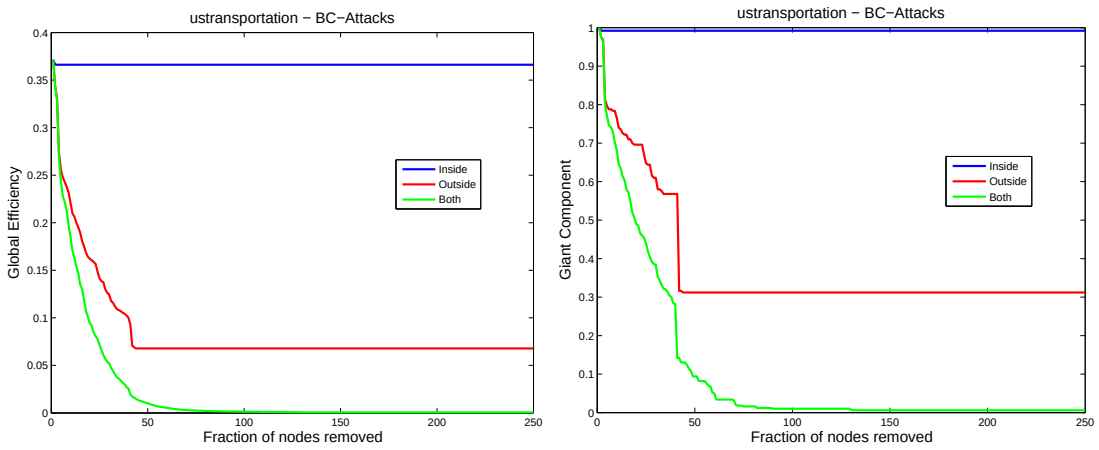


Figure 23: Global efficiency and giant component — BC attacks for UStransportation network.

process considering the removal of both links (inside and outside) for the Dolphins network. The inter-communities bars represent the fraction of nodes that are connecting nodes from different communities. In turn, the intra-community bars represent those links that are connecting nodes belonging to the same community. They are computed taking into account the entire network (on the left) and the links that were removed from the network (on the right). Notice that at the beginning of the perturbation process, despite the fraction of intra-communities links considering the entire network is around 0.3, they were the majority of links lost. Furthermore, they were those which more severely affect the network connectivity (see Figure 20), highlighting the importance of these links to the network connectivity.

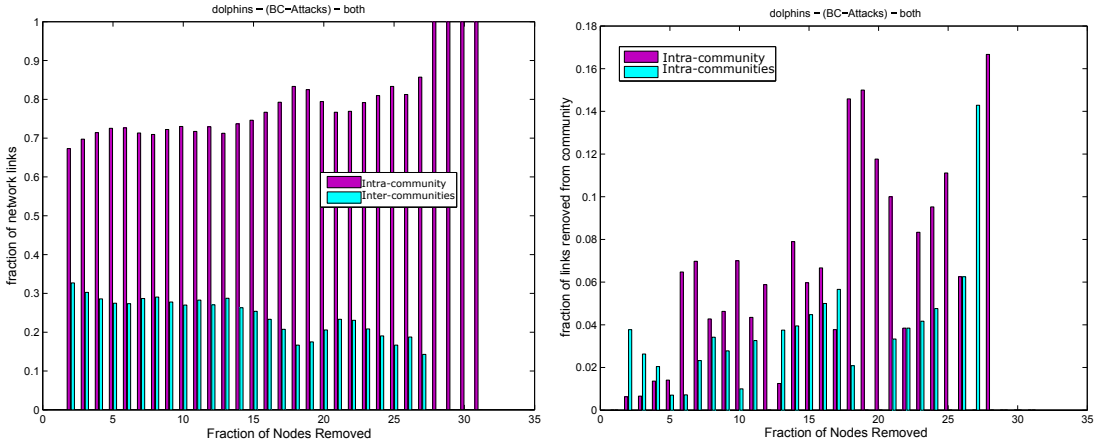


Figure 24: Link statistics — BC attacks for Dolphin network.

Figure 25 shows the results for BA and KE networks.

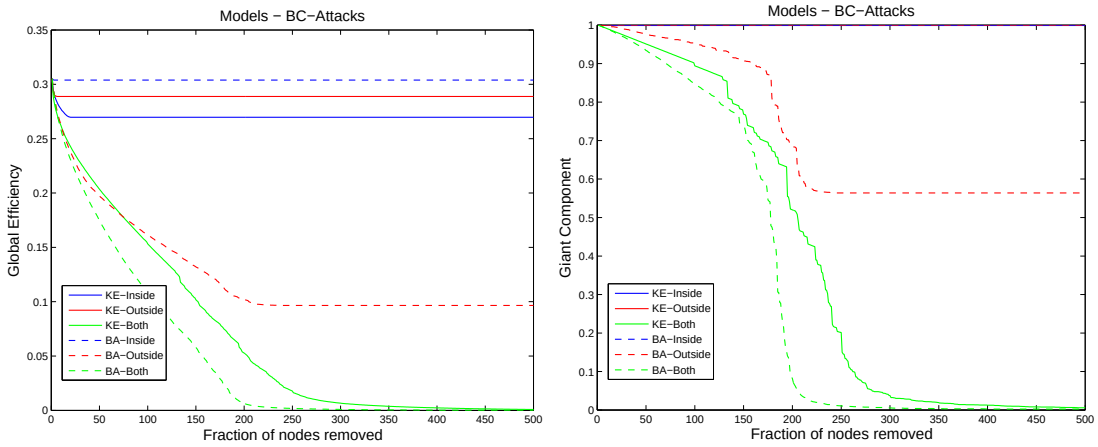


Figure 25: Global efficiency and giant component — BC attacks for BA and KE networks.

The property values of BA networks (dashed lines) showed the same pattern that real network topologies. On the other hand, for the KE networks the removal of both types of links were necessary to actually impact the network connectivity, thus deviating from most of the results achieved for inside and outside strategies, even for some other real networks not presented in this report.

5.1 Adaptive mechanism

The approach proposed here is based on the previous work presented in [19]. It considers nodes as the main agents for controlling the necessary information and procedures responsible for evaluating and promoting changes in the network topology. For that, some nodes are assumed to be more likely to affect the network connectivity according to some likelihood status, and thereby, nodes in their neighborhood can be defined as in a vulnerable state. Here, the unit of analysis changes from nodes to a higher hierarchic structure according to the graph partition generated by the community detection technique [8]. A partition P is a division of a graph into clusters, such that each vertex is assigned to one and only one cluster. Even though some approaches consider vertices belonging to two or more clusters simultaneously [17], here each node takes part of a single community/cluster.

A community C is classified as in a vulnerable state if some specific property value is lower

than expected. Two states are thus defined: vulnerable ($V_{C,t} = 1$) and not vulnerable ($V_{C,t} = 0$), according to:

$$V_{C,t} = \begin{cases} 1 & \text{if } \delta_{C,t} >= \gamma \\ 0 & \text{if } \delta_{C,t} < \gamma \end{cases}$$

where $\delta_{C,t}$ represents the target property value for community C at a specific time t. The threshold to set a community as vulnerable is given by the parameter γ. Both the target property and the vulnerability threshold can be set out as convenient for the vulnerability problem being handled.

The adaptation process is straightforward. It encompasses two main functionalities: the vulnerability assessment and the creation of new links. In compliance with the attacks and failures protocol, after each node removal, every community C assesses its vulnerability state. If applicable (*i.e*, when $\delta_{C,t} < \gamma$) new links are added in the network to try to reverse or minimize the adverse effects of the resulting topological configurations. For validation purposes three strategies were implemented:

- *inside*: adding connections between nodes belonging to the same community,
- *outside*: creating link(s) between node(s) from the vulnerable community to other(s) neighboring community(ies),
- *both*: the combination of inside and outside strategies.

The criteria for the definition of new connections are tied to the vulnerability property. According to the results discussed in [19], the local efficiency is a potentially good estimator for detecting and mitigating vulnerable states. Consider then the concept of local efficiency (4) at the community level:

$$E_{loc}(C_i) = \frac{1}{k_{in}(C_i)(k_{in}(C_i) - 1)} \sum_{l \neq m \in C_i} \frac{1}{d_{lm}}. \tag{11}$$

where $k_{in}(C_i)$ is the number of nodes belonging to community C_i.

For new inside links, the non-connected nodes exhibiting the lowest and the highest local efficiency are connected. As the probability of sharing common neighbors is higher inside the community, this new connection tends to enhance the local community robustness.

The *outside* strategy considers that each vulnerable community (source community) should reinforce its connection with the neighboring communities with which it is weakly connected. Considering C as the set of communities in G and C_i the set of nodes belonging to community i, the neighboring of community C_i is $N(C_i) = \{(C_j \in C | e_{v,u} \in E \wedge v \in C_i \wedge u \in C_j\}$ and $k_{out}(C_{i,j})$ the number of times a community C_j appears in $N(C_i)$. For a vulnerable community C_i, the lowest community degree value $min(k_{out}(C_{i,j})|C_j \in N(C_i))$ is the threshold to define the neighbor community(ies) to create a connection. It means that those neighboring communities with fewer connections are the targets for new connections, thus creating an alternative path between them.

The strategy to identify which nodes will receive new connections in both source and target communities is the same: the priority is for choosing nodes without any link with other communities. In the case of absence of nodes showing this feature, those nodes without connections with the target community are selected.

The *both* strategy combine the inside and outside procedures.

5.2 Results

For performance evaluation, the vulnerability threshold was set to $\gamma = E_{loc}(G) * 0.5$. This definition relies on the assumption that communities with local efficiency below the network local efficiency (see (3)) are more likely to be vulnerable.

Figures 26 to 31 present the adaptive mechanisms performance. Each line shows the evolution of global efficiency (on the left) and size of the giant component (on the right) during the process of attacks regarding different adaptation strategies: H is the original heuristic [19], $E_{loc}(outside)$, $E_{loc}(inside)$ and $E_{loc}(both)$ are for the *outside*, *inside* and *both* strategies, respectively. For benchmarking, G depicts networks without any running adaptive mechanism.

As expected, the improvements accomplished by the $E_{loc}(inside)$ strategy were irrelevant.

The results for the original strategy demonstrate that its performance is related to the network local efficiency, mainly because the creation of links depends on the existence of non-vulnerable nodes. It means that vulnerable states can be detected, but the requirement to add links is not fulfilled. The evolution of both global efficiency and size of the giant component for Football, UStransportation and KE networks, which exhibit the higher scores for local efficiency (see Tables 2 and 1), demonstrate that.

On the other hand, $E_{loc}(outside)$ and $E_{loc}(both)$ strategies produced significant results for all networks evaluated and were able to maintain the majority of nodes connected to the giant component. It is important to notice the influence of the initial network configuration regarding its sensitivity to attacks. For instance, the Football and Dolphins networks are less affected by attacks, so the adaptive community-based mechanisms were able to maintain the global efficiency and nodes in the giant component for most iterations, with the addition of a few links (see Figure 32). In turn, for more sensitive topologies, such as Protein Interaction and UStransportation networks, a small fraction of nodes was not able to be maintained in the giant component, despite the number of links created in the beginning of the adaptation process. Therefore, considering these networks sensitivity, the community-based heuristic improved the network robustness.

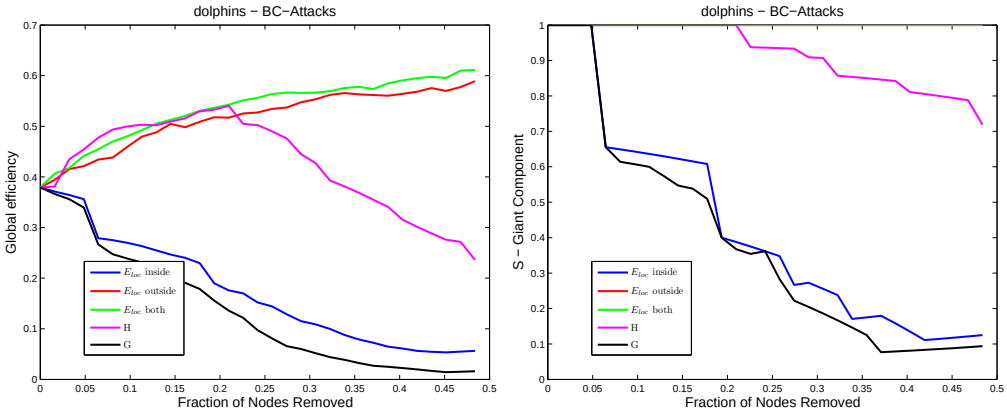


Figure 26: Global efficiency and giant component — Adaptation - Dolphin network.

Figure 32 shows the proportion of new links created at each iteration. Notice that for the Football network a few nodes were added to the network considering the community-based heuristic. Regarding the Protein Interaction network, the proportion of new links for *outside* and

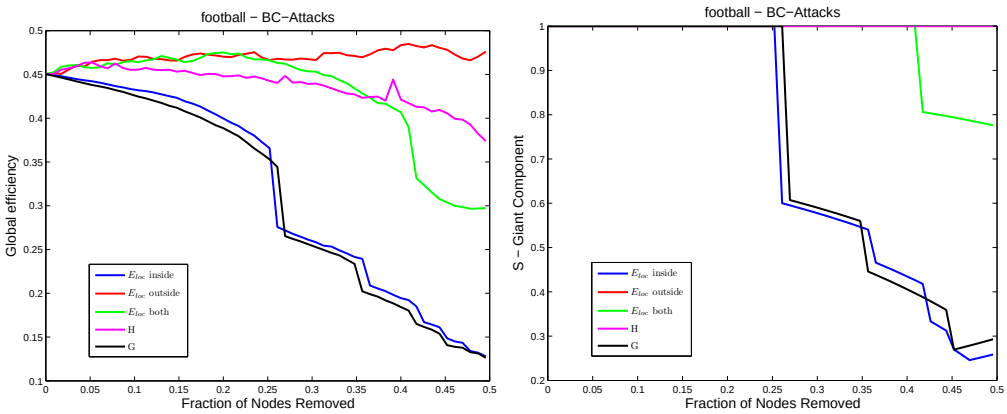


Figure 27: Global efficiency and giant component — Adaptation - Football network.

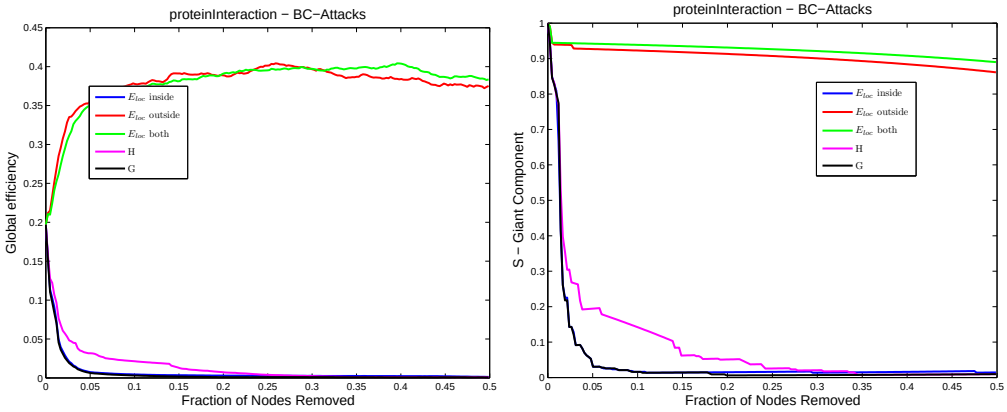


Figure 28: Global efficiency and giant component — Adaptation - Protein Interaction network.

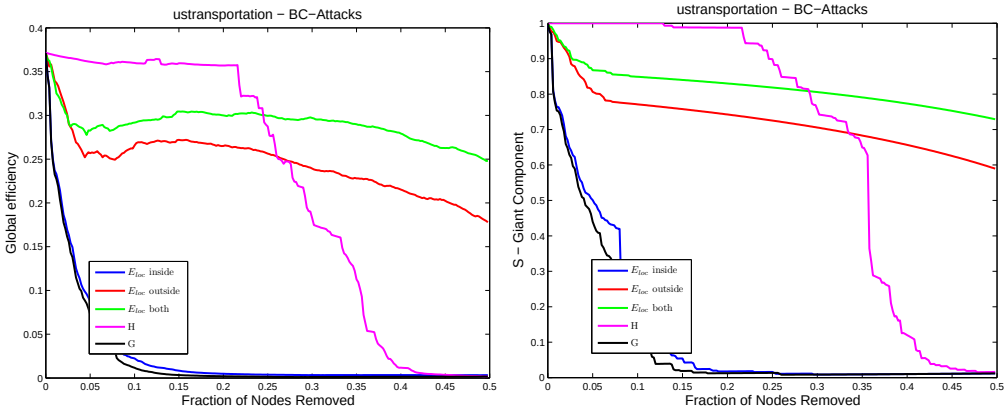


Figure 29: Global efficiency and giant component — Adaptation - USTransportation network.

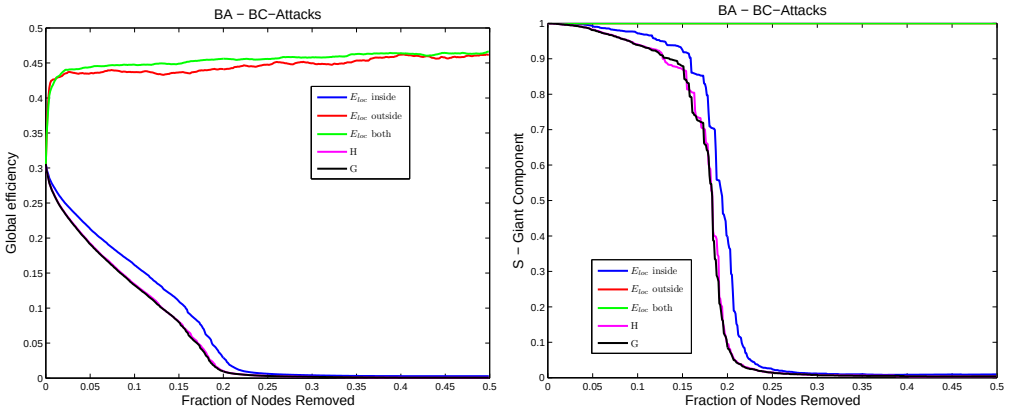


Figure 30: Global efficiency and giant component — Adaptation - BA networks.

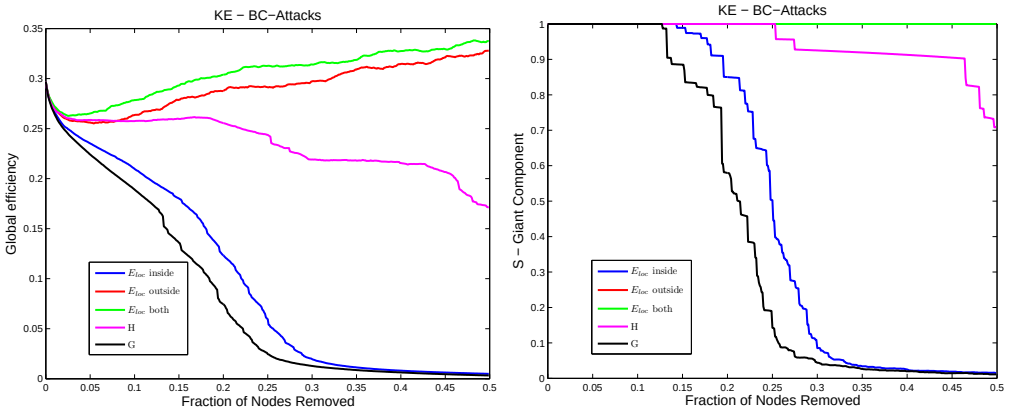


Figure 31: Global efficiency and giant component — Adaptation - KE network.

both strategies at the beginning of process are around 0.40 of the total number of links in the network.

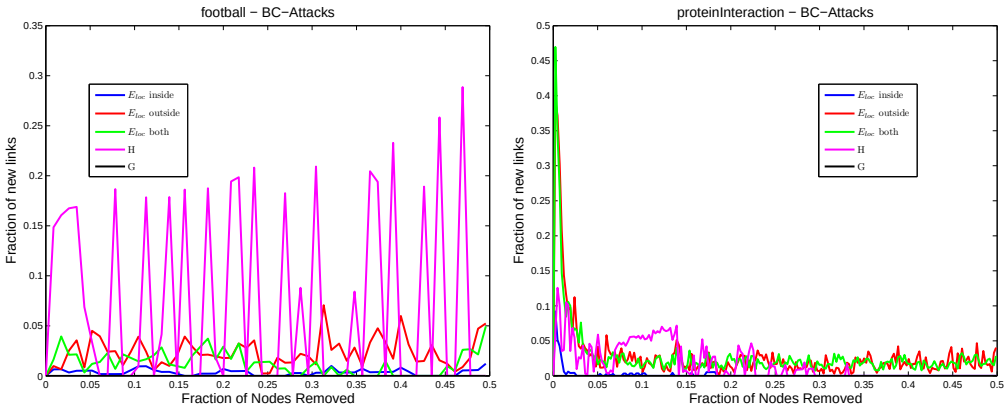


Figure 32: Global efficiency and giant component — Adaptation - USTransportation network.

As highlighted before, this network topology is quite sparse, exhibiting low scores for efficiencies and average degree, so it is necessary to create more links to provide a more robust network. However, after the initial adjustments, the network was able to accommodate perturbations and to maintain its global efficiency and the size of the giant component.

The results present in this Section highlights that central nodes are probably those connecting communities and, therefore, information about the community structure can be worthwhile to design networks that are more resilient to failures and attacks. Besides, the community-based heuristics showed to be a good prospect towards robust mechanisms to deal with the vulnerable topological configurations w.r.t. network robustness to attacks.

6 Vulnerability management in mobile multi-robot systems

Network services are nowadays becoming pervasive, and network infrastructure is widely available: this makes it possible to access the internet, cloud services, and effective communication services in any circumstance. Frequently, the request for these services is triggered and discarded spontaneously as devices (e.g., smartphones, sensors, notebooks) respectively connect to or disconnect from the application service. Examples are collaboration, social, communication and sensor networks.

The downside of this is that the availability of network services is becoming a prerequisite for almost any kind of activity. Sometimes, this availability should take place in dynamic and unpredictable environments, in which the deployment of a network infrastructure is not possible, or not practical, both from an economic and a technological point of view. For instance, very often search and rescue activities are performed in environments where network infrastructure is not available, either because it was damaged during the disaster or because it does not exist at all [39]. Exploiting a sufficient number of mobile robots, it is possible to create a mobile adaptive infrastructure to provide rescuers with network services.

Consider now areas in the world where communication and network infrastructure are not available. In this direction, some pioneering projects[1] have been recently started for using autonomous flying systems (in particular, balloons) to bring Internet services to areas where they are not available. In this situation, the objective is to provide Internet services to clients (e.g. mobile phones, laptops, tablets, etc.). While some historical data may be available on the clients, their motion has to be generally considered as unpredictable and uncontrollable.

Both application examples highlight the necessity of bringing efficient network solutions in unstructured environments. Figure 33 illustrates a scenario that fits into several of such contexts of applications. In this example, *mobile robots* are controllable entities, that can communicate with each other, and whose position can change in order to provide a network services to a group of *mobile clients*, which refers to generic uncontrolled mobile entities, that access the network service provided by the mobile robots.

Mobile robots, if equipped with appropriate communications devices, can be exploited to create an infrastructure network to provide communication services in such environments. Providing network services for unstructured environments through interconnected mobile robotic systems involves several issues, from deployment and communication efficiency to topology control.

This Section focuses on the mobile robot networks, more specifically it addresses how to achieve a robust network concerning the connectivity maintenance. A robust network implies that, despite robot failures, most of its elements are still connected, being able to maintain a

[1] See for instance: http://www.google.com/loon/ (Loon project), http://altave.com.br (Conectar project)

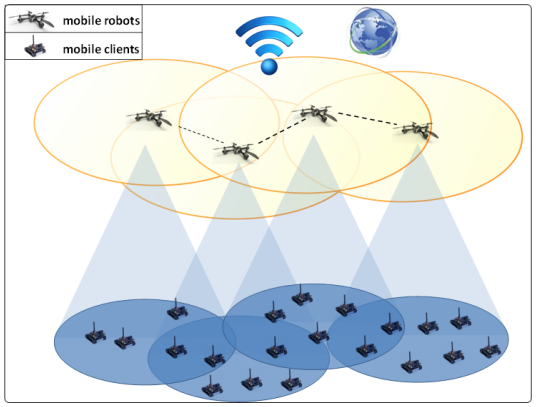


Figure 33: Application scenario

certain level of service. On the other hand, it is necessary to guarantee that the interconnection topology defines a connected graph, thus ensuring the possibility of exchanging data among all the nodes in the network.

6.1 System model

The initial model takes into account both the feasibility of manipulated data to be obtained from real applications and the model requirements, such as local information processing, agent's position controllability and disk communication model.

Consider a multi-robot system composed of N robots, which know their position and are able to communicate with other robots in their communication radius R, forming a network. The communication network topology can be modeled as an undirected graph $\mathcal{G} = (V, E)$, defined by a set of nodes $V = \{1, 2, ..., v\}$ and a set of links $E \subseteq VXV$, representing the robots and the existence of a direct connection between them, respectively.

Regarding real systems, there is a communication cost associated with each link. Taking into account that the communication between nodes is mostly through the shortest path, the communication cost can be based on the Euclidean distance. Consider the i-th robot's position as a point x_i in a Euclidean space R^m. Thus, the distance between nodes i and j is given by:

$$dist_{i,j} = \|x_i - x_j\|, \tag{12}$$

and the communication cost function as:

$$cost_{i,j} = 1 - \exp(-dist_{i,j}). \tag{13}$$

Thus, the robot network is represented by two NXN square matrix: a boolean adjacency matrix A indicating the existence of a direct communication channel between nodes, so if there is a 1-hop path between node i and j then $A(i,j) = 1$; and a weight matrix w with the communication cost between nodes i and j:

$$a_{i,j} = \begin{cases} 1 & \text{if } d_{(\overline{p}(i),\overline{p}(j))} <= R \\ 0 & \text{otherwise} \end{cases}$$

$$w_{i,j} = \begin{cases} cost_{(\overline{p}(i),\overline{p}(j))} & \text{if } a(i,j) \\ 0 & \text{otherwise} \end{cases}$$

where $\overline{p}$ is the node position vector.

Now, let $x_i \in \mathbb{R}^m$ be the state of the i-th robot. Hereafter it is assumed that each robot's state is the position of the robot itself and that the velocity of the robots can be controlled. Hence, we model each robot according to the following discrete time kinematics:

$$x_i\,(t+T) = x_i\,(t) + T\,\mu_i\,(t)\,. \tag{14}$$

where $T > 0$ is the sampling time, and $\mu_i\,(t) \in \mathbb{R}^m$ is the control input computed at time t.

6.2 Robustness model

This section introduces the local heuristic for estimating the probability of a node being in a vulnerable configuration and a control strategy to improve possible harmful configurations.

6.2.1 Robustness assessment

Despite the previous approach presented in [20], a new local heuristic was developed based on the application features. It is assumed that nodes can acquire the position from their 1-hop and

2-hops neighbors. Now, consider $d(v, u)$ as the minimal number of hops that connect nodes v and u. Subsequently, define $\Pi(v)$ as the neighborhood of node v, that is the set of nodes from which v can acquire information, namely

$$\Pi(v) = \{u \in V(G) : d(v, u) \leq 2\}$$

Moreover, let $|\Pi(v)|$ be the number of elements of $\Pi(v)$. In addition, define $\Pi_2(v) \subseteq \Pi(v)$ as the set of the 2-hop neighbors of v, that comprises only nodes whose shortest path from v is exactly equal to 2 hops, namely

$$\Pi_2(v) = \{u \in V(G) : d(v, u) = 2\}$$

Let $L(v, u)$ be the *number of paths* between nodes v and u. Subsequently, define $Path_\beta(v) \subseteq \Pi_2(v)$ as the set of v's 2-hop neighbors that are reachable through at most β paths, namely

$$Path_\beta(v) = \{u \in \Pi_2(v) : L(v, u) \leq \beta\}$$

Moreover, let $|Path_\beta(v)|$ be the number of elements of $Path_\beta(v)$.

As an example, setting $\beta = 3$ implies that $Path_\beta(v)$ contains all the 2-hop neighbors u of node v for which no more than 3 different paths exist that connect v to u. Therefore, using a low value for β, is it possible to identify the most weakly connected 2-hop neighbors. Hence, the value of $|Path_\beta(v)|$ is an indicator of the magnitude of node fragility w.r.t. connectivity.

Hereafter, to identify 2-hop neighbors that are connected by a single path, the value used for β is 1, that represents a critical situation for the network connectivity.

On these lines, the probability of a node being vulnerable to failures $P_\theta(v) \in (0, 1)$ is defined as:

$$P_\theta(v) = \frac{|Path_\beta(v)|}{|\Pi(v)|} \tag{15}$$

Notice that $P_\theta(v)$ can be locally computed by each node, relying on information regarding

Table 4: Neighborhood parameters for the network shown in Fig. 34

v	$\Pi(v)$	$\Pi_2(v)$	$Path_\beta(v)$	$P_\theta(v)$
1	{2,3,5,7,9}	{2,3,5,7}	{2,3,5,7}	0.800
2	{1,3,4,5,6,7,9}	{1,3,5}	{1,5}	0.285
3	{1,2,4,5,6,7,8,9,10}	{1,2,7,8}	{1,7}	0.222
4	{2,3,5,6,7,9,10}	{5,6,7,9,10}	{5,7,10}	0.428
5	{1,2,3,4,6,7,10,8,9}	{1,2,4,6,7}	{1,2,4,6,7}	0.555
6	{2,3,4,5,7,9,10}	{5,7,9,10}	{5,7,10}	0.428
7	{1,2,3,4,5,6,9}	{1,3,4,5,6}	{1,3,4,5,6}	0.714
8	{3,5,9,10}	{3,9}	{9}	0.250
9	{1,2,3,4,6,5,7,8,10}	{4,6,8,10}	{8}	0.125
10	{3,4,6,5,8,9}	{4,6,9}	{4,6,9}	0.333

1-hop and 2-hop neighbors.

Consider the network in Figure 34 with the instantiation showed in Table 4. It is possible to notice that node 1 is clearly vulnerable because it is relying only on node 9 to communicate with the entire network. The probability assigned to node 1 is $P_\theta(1) = 0.800$. In contrast, node 8 has a single path to node 9, however $|\Pi(8)| = 4$, indicating that alternative paths, with more than 2-hops, can exist, as it is the case. So, its probability of being vulnerable is 0.2500.

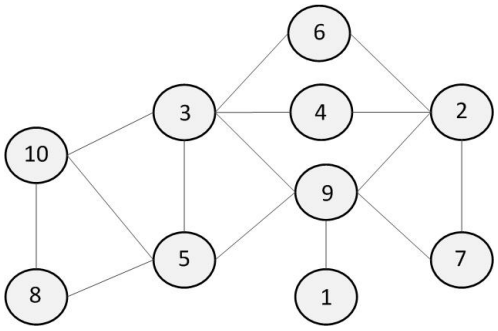


Figure 34: Example of a network with 10 nodes

6.2.2 Control strategy for robustness improvement

This Section introduces a control strategy that, based on the local robustness evaluation procedure previously defined, aims at improving the robustness of the network.

Assume the i-th robot identifies itself as vulnerable. In this case, the aim of the control strategy is to increase the number of links towards its 2-hop neighbors that are in $Path_\beta(i)$, for

Table 5: Model settings for simulation

N	Area $\mathcal{A}$	Range R	# Iterations	# Experiments
20	50	18	10	200
100	400	90	50	100

a given value of β. Hence, define $x^i_\beta \in \mathbb{R}^m$ as the barycenter of the positions of the robots in $Path_\beta(i)$, namely

$$x^i_\beta = \frac{1}{|Path_\beta(i)|} \sum_{j \in Path_\beta(i)} x_j \tag{16}$$

The control law μ_i in (14) is then defined as follows:

$$\mu_i = \frac{x^i_\beta - x_i}{\left\| x^i_\beta - x_i \right\|} \alpha \tag{17}$$

where $\alpha \in \mathbb{R}$ is the linear velocity of the robots that, for the sake of simplicity, is assumed to be constant.

This control law drives vulnerable robots towards the barycenter of the positions of robots in $Path_\beta(i)$, thus decreasing the distance to those robots and eventually creating new edges in the communication graph.

The control law in (17) is applied in a probabilistic manner, with higher probability for those robots i whose value $P_\theta(i)$ is high. This is obtained comparing $P_\theta(i)$ with a random number $r \in (0, 1)$: if $P_\theta(i) > r$, then the i-th robot considers itself as vulnerable, and applies the control law in (17).

6.2.3 Simulation model

The proposed control strategy was validated using a simulation environment developed in Matlab. This simulation environment allows to randomly positioning a variable number N of robots in $\mathbb{R}^2$, over a bounded area of size $\mathcal{A}$. Given a communication radius R, the communication network is then generated, based on the relative positions among the robots. Simulations were performed using the parameter set defined in Table 5, with $\beta = 1$, $\alpha = 0.25R$.

The evaluation of robustness improvement is based on the assumption that failures of the

most central nodes are more harmful to network connectivity. Consider a graph $\mathcal{G}$ with N nodes. Let $[v_1, \ldots, v_N]$ be the list of nodes ordered by descending value of BC. Let $\varphi < N$ be the minimum index $i \in [1, \ldots, N]$ such that, removing nodes $[v_1, \ldots, v_i]$ leads to disconnecting the graph, that is the graph including only nodes $[v_{\varphi+1}, \ldots, v_N]$ is disconnected. Then, the *level of network robustness* of $\mathcal{G}$ is defined as:

$$\Theta(\mathcal{G}) = \frac{\varphi}{N}, \tag{18}$$

Namely, the level of robustness defines the fraction of central nodes that need to be removed from the network to obtain a disconnected network, i.e. $S(\mathcal{G}) < 1$. Small values of $\Theta(\mathcal{G})$ imply that the graph may lose connectivity in case of failure of a small fraction of its nodes. Therefore, increasing this value increases the robustness of the network.

Notice that $\Theta(\mathcal{G})$ is only an estimate of how far the network is from getting disconnected w.r.t. fraction of nodes removed. In fact, it might be the case that different orderings of nodes with the same BC produce different values of $\Theta(\mathcal{G})$.

Repeated experiments were carried out in order to ensure the numerical accuracy (see $\#Experiments$ in Table 5). Starting from random positioning, discrete time iterations were performed. At each iteration, the following procedure was executed:

1. **network adaptation**: based on the local evaluation of robustness (15), control law (17) is applied to the fraction $\zeta \in (0, 1)$ of robots that identified themselves as vulnerable,

2. **robustness evaluation**: computes the number of central nodes required to be removed from the network to achieve a disconnected topology.

6.2.4 Simulation results

Results of simulations for $N = 20$ and $N = 100$ robots are in Figures 35 and 36, respectively. Each color depicts one iteration result. For both network sizes, the adaptive mechanism enhanced the initial robustness level more than three times. Also, in general, at each iteration, as the robustness level increases, the number of adaptations performed decreases, demonstrating

the effectiveness of the process. The trend is however not monotonic, due to fact that each robot estimates its vulnerability state in a probabilistic manner.

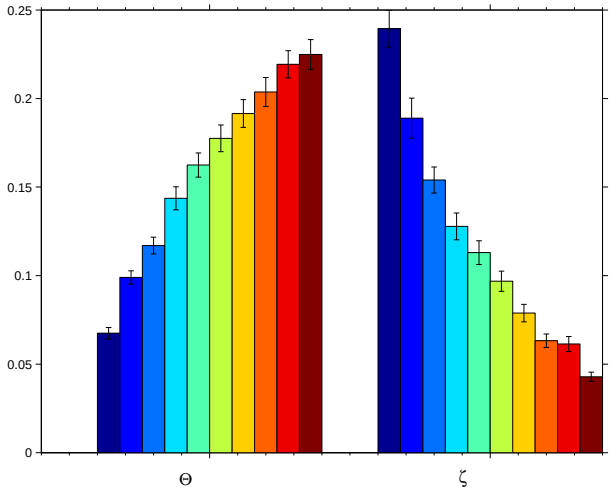


Figure 35: Adaptive mechanism performance, N=20

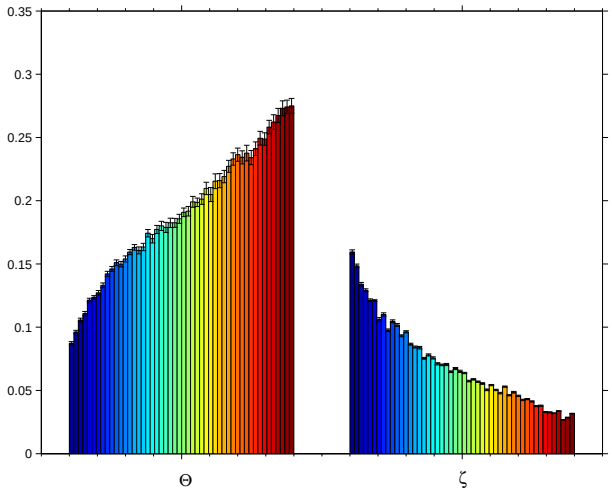


Figure 36: Adaptive mechanism performance, N=100

Figures 37 to 39 present the evolution of the network illustrated in Figure 1 as a result of the adaptive mechanism. The vulnerable nodes are identified as blue. The red nodes are those that connect vulnerable nodes to their 2-hops neighbors. As vulnerable nodes rely on these nodes, their loss can be potentially harmful to the network connectivity. Notice that such harmful nodes can also hold a high probability of being vulnerable. Notably, in seven iterations (the sixth is not shown), the topology has evolved to a more robust one.

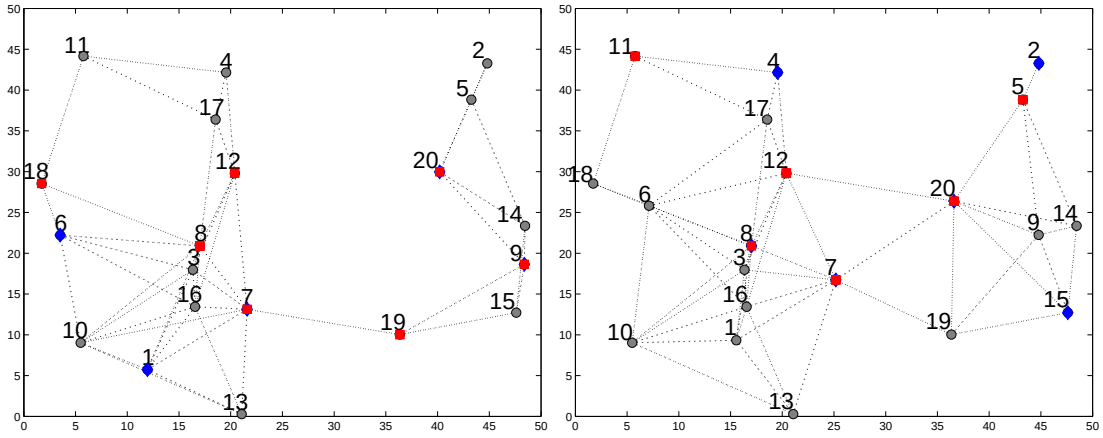


Figure 37: Example of the adaptive mechanism performance - Iterations 1 and 2, N=20

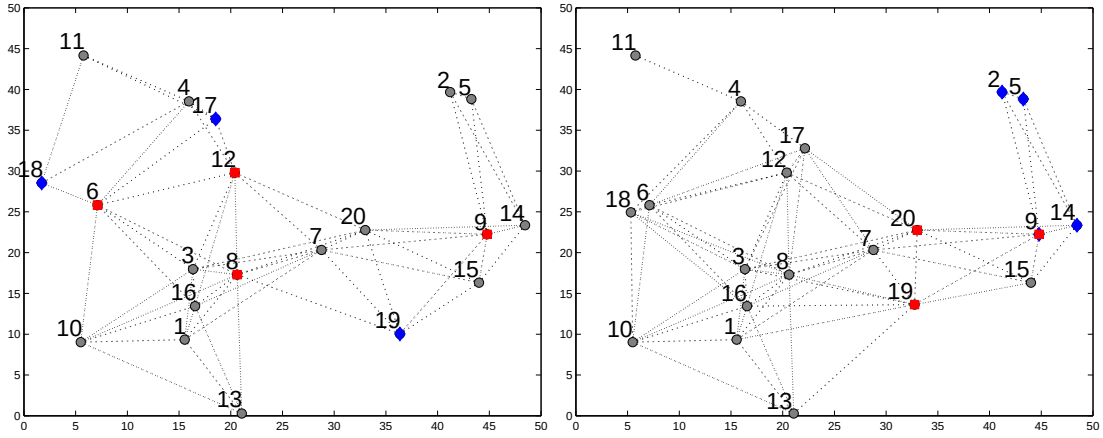


Figure 38: Example of the adaptive mechanism performance - Iterations 3 and 4, N=20

Figure 40 presents the evolution of the network introduced in Section 3 (see Figure 2). Despite the number of nodes in the network, it is possible to notice how the critical configuration could be reverted after 22 iterations.

Some additional examples can be freely viewed online[2].

6.3 Utility-based approach

The control strategy defined in Section 6.2.2 considers that a vulnerable node moves toward the average position of those nodes for which it relies on a single path. In the utility-based approach, the vulnerable node will move towards that node which mostly improves its robustness. The

[2] $http://www.arscontrol.unimore.it/syroco15\_networks/$

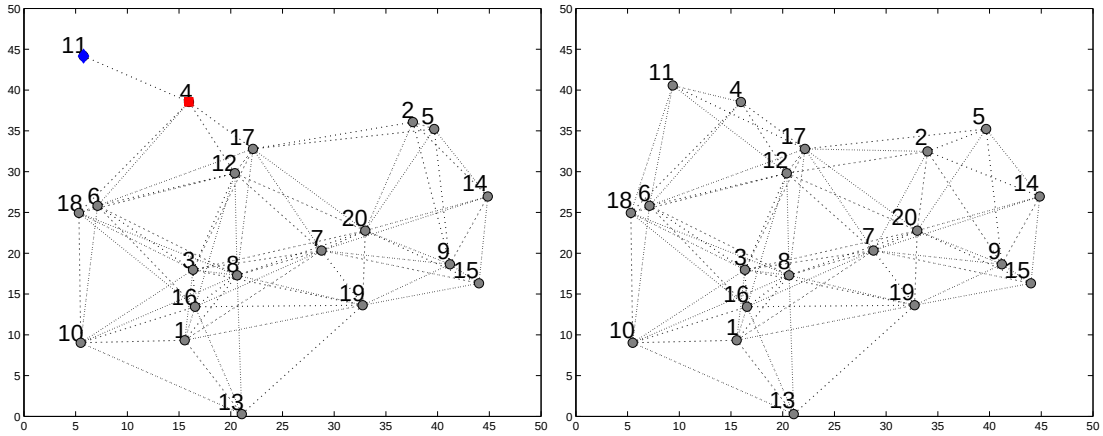


Figure 39: Example of the adaptive mechanism performance - Iterations 5 and 7, N=20

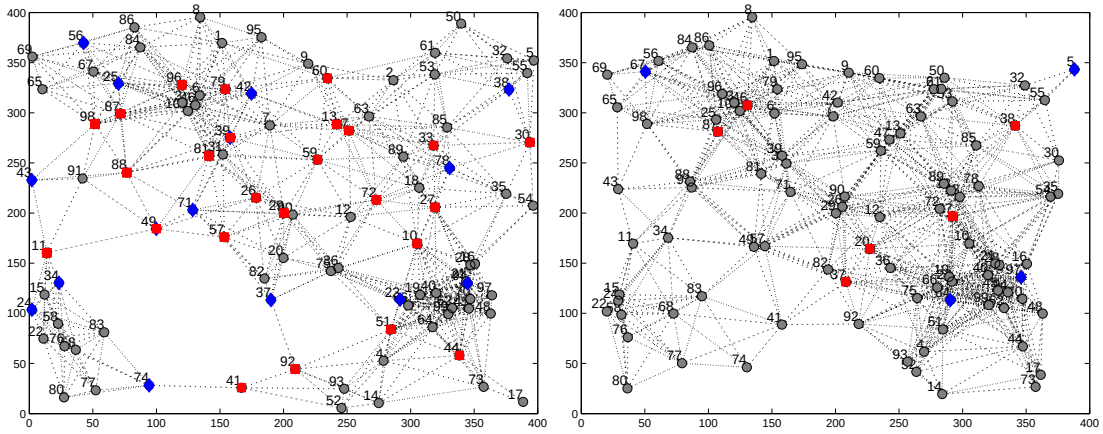


Figure 40: Example of the adaptive mechanism performance - Iterations 1 and 22, N=100

utility of a node i that moves towards a node j is given by:

$$U_j(i) = P_\theta(i') - P_\theta(i) \tag{19}$$

where $P_\theta(i')$ is the probability of i being vulnerable if it gets closer to j. Thus, the new goal state is given by j which maximizes the robustness of i:

$$x^i_{\beta_{\max}} = x_j : U_j(i) > 0 \wedge U_j(i) \in \max(U(i)), \tag{20}$$

where $U(i) = \{U_j(i) \forall\, j \in Path_\beta(i)\}$.

Figure 41 and 42 show the results comparing x^i_β , $x^i_{\beta_{\max}}$ and $x^i_{\beta_{both}}$, a strategy that considers

x^i_β when $x^i_{\beta_{\max}} = \emptyset$. The strategy of always moving toward an average position performed better for both network sizes. The policy of only moving in the direction of the node that produces the higher utility did not achieve the best performance, even when combined with the possibility of moving to the average position, which is the case when there is no position that can improve the node utility. Probably, moving to the position with the higher probability of increasing the node robustness can also increase the probability of losing connection to the poorly connected nodes. Notice that nodes move without knowing the action of other nodes in their neighborhood. Also, in general, the number of adaptations performed for x^i_β were fewer than for the $x^i_{\beta_{both}}$.

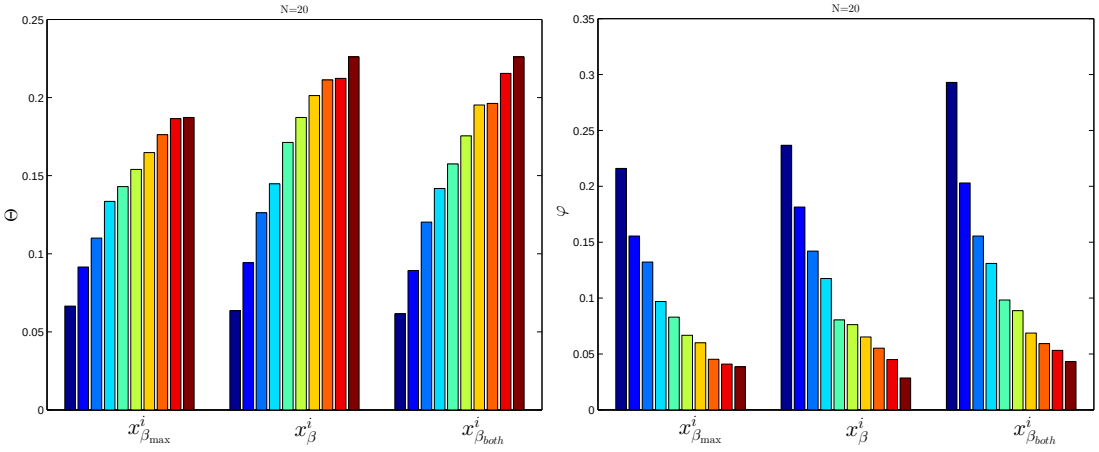


Figure 41: Adaptive mechanism performance N=20

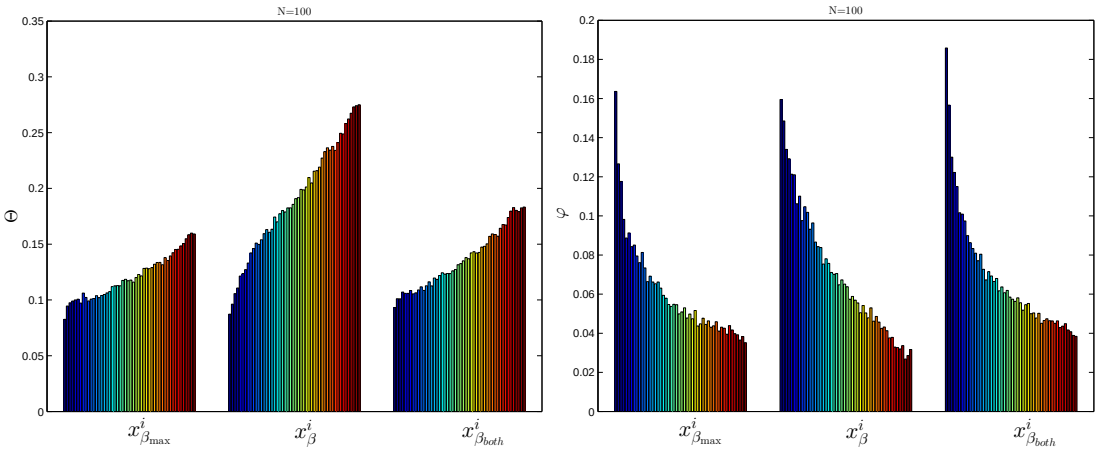


Figure 42: Adaptive mechanism performance N=100

6.4 Algebraic connectivity model

The robustness control improves the network robustness to attacks, but does not ensure that all the robots will remain connected in the network in case of attacks. Sabattini et. al. [35], proposed a control algorithm to solve the global connectivity problem in a decentralized manner. It is based on the algebraic connectivity property. As is well known, the algebraic connectivity approaches 0 when the network is poorly connected. Then, the control strategy ensures the network connectivity maintenance through a decentralized algebraic connectivity estimation and a strategy to improve it.

Maintaining the connectivity entails designing a controller that ensures that, if the graph is connected at time $t = 0$, then it will remain connected $\forall t \geq 0$.

Let A be the adjacency as defined in Section 6.1 and $D = \text{diag}(\{d_i\})$ be the degree matrix of the graph, where d_i is the degree of the i–th node of the graph, that is $d_i = \sum_{j=1}^{N} a_{ij}$. The (weighted) Laplacian matrix of the graph is defined as $L = D - A$. The Laplacian matrix exhibits some remarkable properties:

1. Let $\mathbf{1}$ be the column vector of all ones. Then, $L\mathbf{1} = \mathbf{0}$.

2. Let λ_i, $i = 1, \ldots, N$ be the eigenvalues of the Laplacian matrix.

   - The eigenvalues can be ordered such that

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_N \tag{21}$$

   - $\lambda_2 > 0$ if and only if the graph is connected. Then, λ_2 is defined as the algebraic connectivity of the graph.
   - Considering a weighted graph, λ_2 is a non–decreasing function of each edge weight.

Let L be the Laplacian matrix of the communication graph representing a group of N single–integrator robots:

$$\dot{p}_i = u_i^c \tag{22}$$

where $p_i \in \mathbb{R}^m$ is the position of the i–th robot, and $u_i^c \in \mathbb{R}^m$ is the control input. Let $p = \left[p_1^T \dots p_N^T\right]^T \in \mathbb{R}^{Nm}$ be the state vector of the multi–robot system. Then, the connectivity is guaranteed if the second smallest eigenvalue of L (that, hereafter, will be referred to as λ_2) is strictly greater than zero.

Define now $\epsilon > 0$ to be the desired lower–bound for the value of λ_2. The control strategy will then be designed to ensure that the value λ_2 never goes below ϵ. An *energy function* is applied for generating the decentralized connectivity maintenance control strategy.

Definition 6.1. *An energy function*

$$V(\lambda_2) = V(\lambda_2(p)) : \mathbb{R}^{Nm} \mapsto \mathbb{R}$$

exhibits the following properties:

(P1) It is continuously differentiable $\forall \lambda_2 > \epsilon$.

(P2) It is non–negative.

(P3) It is non–increasing with respect to λ_2, $\forall \lambda_2 > \epsilon$.

(P4) It approaches a constant value, as λ_2 *increases.*

(P5) It suddenly increases, as λ_2 *approaches* $\epsilon > 0$, *namely*

$$\lim_{\lambda_2 \to \epsilon} V(\lambda_2(p)) = \infty$$

The control design essentially drives the robots to perform a gradient descent of $V(\cdot)$, in order to ensure connectivity maintenance. Since λ_2 and its gradient are global quantities, the following centralized connectivity maintenance control law may be defined:

$$\underline{u}_i^c = -\frac{\partial V(\lambda_2(p))}{\partial p_i} = -\frac{\partial V(\lambda_2(p))}{\partial \lambda_2}\frac{\partial \lambda_2}{\partial p_i} \tag{23}$$

Assuming $\tilde{\lambda}_2$ as the estimated value for λ_2 computed in a decentralized manner, the maintenance control law introduced in Eq. (23) can be implemented in a decentralized manner:

$$u_i^c = -\frac{\partial V\left(\lambda_2^i\left(p\right)\right)}{\partial \lambda_2^i}\frac{\partial \tilde{\lambda}_2}{\partial p_i} \tag{24}$$

where the energy function $V\left(\lambda_2^i\left(p\right)\right)$ is defined using the lower–bound $\tilde{\epsilon}$ defined as follows

$$\tilde{\epsilon} = \epsilon + \Psi \tag{25}$$

where Ψ is the upperbound on the estimation error.

In [34] the energy function has been defined as follows (Fig. 43):

$$V\left(\lambda_2^i\right) = \begin{cases} \coth\left(\lambda_2^i - \tilde{\epsilon}\right) & \text{if } \lambda_2^i > \tilde{\epsilon} \\ 0 & \text{otherwise} \end{cases} \tag{26}$$

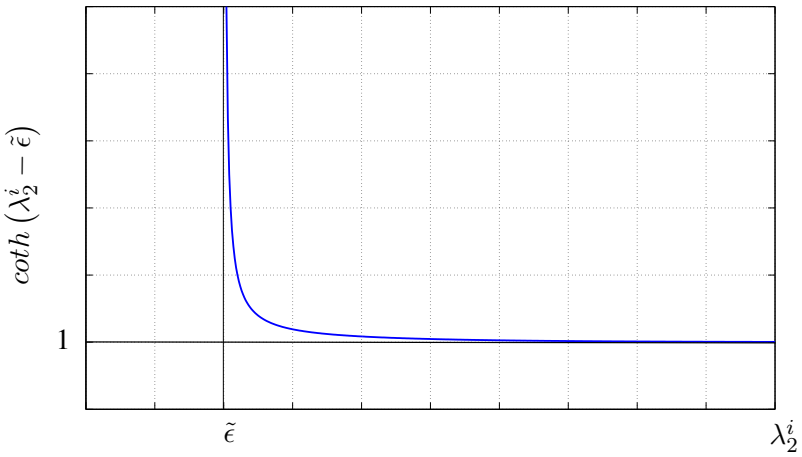


Figure 43: Energy function, defined in Eq. (26)

The following theorem proves that, assuming the initial value of λ_2 is sufficiently large (i.e. the graph is connected above a certain threshold), then the control law ensures the connectivity maintenance. Satisfying this condition simply implies that the robots are *close enough* to each other, in the initial configuration.

Theorem 1. *Consider the dynamical system described by Eqs.* (22), (24). *Then,* $\exists \epsilon, \tilde{\epsilon}$ *defined according to Eq.* (25) *such that, if the initial value* $\tilde{\lambda}_2\left(0\right) > \tilde{\epsilon}$, *then the value of* λ_2 *never goes below* ϵ.

For details about the estimation procedure, proofs and simulation results see [34, 35, 46].

6.5 Robustness improvement and connectivity maintenance model

This section describes the unified model that ensures both connectivity and robustness when it is possible. Let u_i^c represents the connectivity maintenance model and u_i^r the robustness model, so the combined model is given by:

$$\dot{p}_i = u_i^c \sigma + u_i^r \psi \tag{27}$$

where σ and ψ set the desired gain for the connectivity controller and the robustness controller, respectively.

The connectivity controller was implemented in Matlab, as defined in [35]. On the other hand, as the first model of robustness controller was developed for validation purposes, it had to be adapted to support a continuous time model. Having both controllers implemented, the primary focus was to evaluate the combined model performance and its sensibility to the parametrization settings. Thus, for the sake of analysis simplification the computational performance were not analyzed, as well as the algebraic connectivity was computed using global information.

The parametrization of the system model is quite different in the two models, such as the area, the number of robots and the definition of link weights. So, the model defined in Section 6.1 was applied for this initial analysis. At first, the main goal was to evaluate the impact of gains definition in the integrated model performance. Notice that the gains represents the weight of each controller. In the future, it is expected that the gain parameters will be set according to the system state.

The network model applied here is that defined in Table 5 and the link weights as defined in Section 6.1. The u_i^r adopted considers the average position, so $u_i^r = x_\beta^i$. For the continuous time model, the parameters t_0, t_i, and t_f, which means respectively, the initial, the interval and the total simulation time were added. Thus, the simulation protocol encompasses:

1. **robustness evaluation**: computes the number of central nodes required to be removed from the network to achieve a disconnected topology.

2. **network adaptation**: based on the local evaluation of robustness defined in Section 2.4, the integrated control law (27) is applied for a time interval from t_0 to t_i, for the fraction $\zeta \in (0, 1)$ of robots that identified themselves as vulnerable.

Table 6 presents the assessed values for σ and ψ. The purpose is to assess the behavior of the combined control law model. For instance, for $\sigma = 0$ and $\psi = 1$ the connectivity controller is not active, for $\sigma = 1$ and $\psi = 1$ both controller are active and exhibit the same weight. On the other hand, for $\sigma = 1$ and $\psi = 2$ the robustness controller is more relevant. Also, for these experiments the time interval parameters were defined as $t_0 = 0$, $t_i = 5$ and $t_f = 50$.

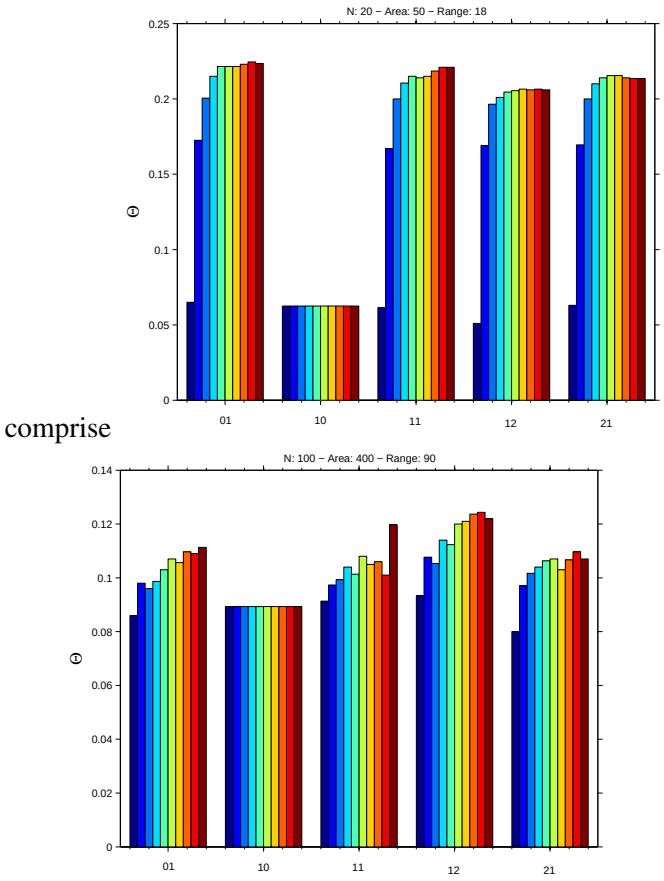
Table 6: Integrated model parametrization

id	σ	ψ
01	0	1
10	1	0
11	1	1
12	1	2
21	2	1

Figure 44 presents the robustness improvement for every combination of parameters σ and ψ. For networks with 20 nodes, the robustness had improved when the robustness control was active, that is for $\psi > 0$. Besides, the achieved level of robustness was similar to those presents in Figure 35, which only considers the robustness control. On the other hand, for networks with 100 nodes, despite the integrated control law had produced more robust networks, the level of robustness did not achieve the same results as those presented in Figure 36.

As these results are preliminaries, it is necessary to conduct additional experimental analysis in order to evaluate the integrated control model performance. Initially, the network model parametrization (e.g. number of nodes, range, area) and the values for σ and ψ must be extended. Probably, new mechanisms to assess the integrated control law performance need to be designed.

Figures 45 and 46 illustrate the result for the algebraic connectivity, the global efficiency, and the size of the giant component for networks with 20 and 100 nodes, respectively. Notice that when only the algebraic connectivity control law is active (10) the network property values are maintained. The robustness control law always improves the algebraic connectivity and



comprise

Figure 44: Network robustness evolution for N=20 and N=100 - Integrated Model

the global efficiency. The impact of such feature is one of the aspects that can be aggregated to the model in the future, through a model able to consider the trade-off between robustness and efficiency. Another interesting aspect is that the property values exhibit different scales for networks with 20 and 100 nodes. As pointed out before, this aspect need to be analyzed in details in the extension of the experimental set up.

7 Conclusions and future works

This report presents the activities carried out during the regular postdoctoral scholarship and the Research Internships Abroad program (BEPE). The overall purpose was to propose mechanisms to improve the robustness of complex networks to failure and attacks. The designed mechanisms concerning the detection and mitigation of vulnerable topological configuration.

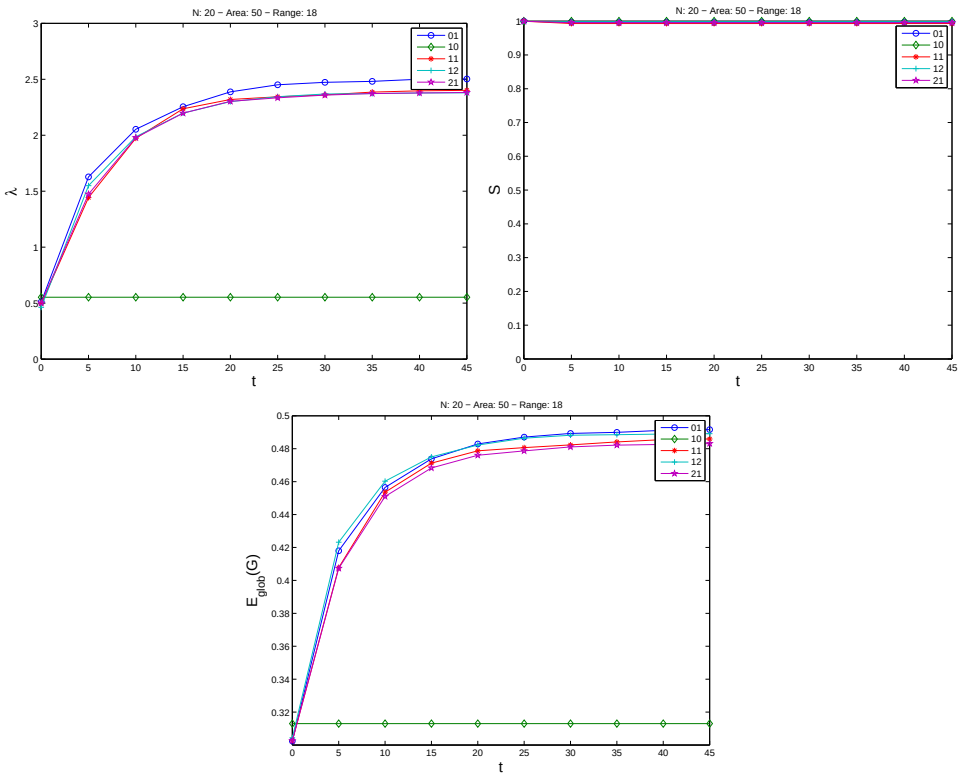


Figure 45: Property values evolution N=20 - Integrated Model

In general, we can emphasize as our main achievements the validation of the model proposed in [20], the development of a community-based approach [22], and the proposal of a mechanism to improve the robustness in the context of multi-robot networks.

Regarding the initial model validation, larger network models and additional real complex network topologies were considered. Additionally, a comparison between the performance of a random self-regenerating and the vulnerability-based approach were discussed, demonstrating the importance of the proposed mechanism. This work yielded a publication in the Advances in Complex Systems Journal (Qualis $B1$) [21]. An adaptive parametrization in agreement with the network characteristics and the addition of dynamical issues, such as temporal topology variation from node drop off and/or mobility, are some of the extensions that can be pursued.

This approach was also applied to the development of a hierarchical solution based on the community structure instead of nodes. The first aspect highlighted was that central nodes are probably those connecting communities and, therefore, information about the community struc-

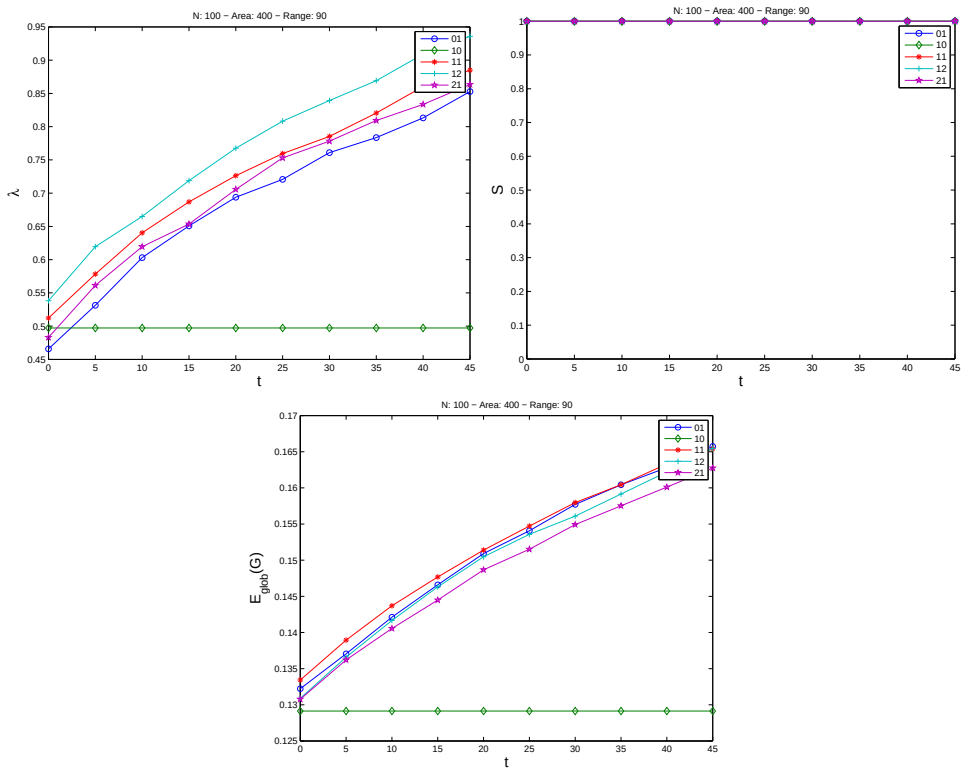


Figure 46: Property values evolution N=100 - Integrated Model

ture can be worthwhile to design more resilient networks, w.r.t. failures and attacks tolerance. Taking this premise into account, the solution proposed encompasses three main components: community identification, vulnerability detection and vulnerability mitigation. For the first component, a well-established method was applied [8]. For the second one, those communities exhibiting a local efficiency below the local efficiency of the network are considered in a vulnerable state.

Finally, the proposed heuristic to mitigate possibly vulnerable states relies on the creation of additional links between communities. The relative importance of the community structure, we evaluated considering creating links inside or outside the community, or both. The outside and both community-based heuristics outperformed both the inside strategy and the original method based on node information. Furthermore, they showed less sensitivity to the initial network topology. Thus, the community-based heuristics showed a good prospect towards mechanisms to deal with vulnerable topological configurations. This work was presented at PESARO - The

Fifth International Conference on Performance [22]. Further research may comprise the evaluation of local mechanisms for communities detection and the parameter estimation.

The project in collaboration with Unimore started as a concept that aimed at integrating two existing approaches regarding network connectivity: connectivity maintenance and robustness to failures. As each of them considers specific models, we defined an application environment that is feasible to be the requirement for most of the new applications and, as a consequence, with several open issues.

Based on the application features, we proposed a new approach for robustness evaluation. The study of such problem in the mobile multi-robot systems is entirely new. The mechanisms for vulnerability detection and robustness improvement proved to be efficient. The main results were accepted for presentation at the 11th Symposium on Robot Control. In addition to the robustness mechanisms, a new measure and a protocol to evaluate the network robustness were developed.

Furthermore, the connectivity maintenance approach was subject to study and further implementation in the simulation environment. Having both control strategies implemented, integrating them is not a trivial task. Initially, some simple combinations of parametrization were assessed. However, we need to extend the experiment to several combinations of environment and models parametrization to evaluate the solution performance and to promote the necessary adjustments.

In addition to the parametrization and the fine-tuning of the integrated model, the trade-off between efficiency and connectivity maintenance, the collision avoidance, the addition of a cost function and the environment awareness mechanisms are some of the main points to be addressed. More specifically, we can emphasize as potential future works:

- The addition of a failure probability model: the vulnerability assessment may be improved by an estimator of the probability of nodes failure based on their status or on the network state or dynamic.

- The addition of a cost function to evaluate the trade-off between the adaptation gain and its cost.

- The design of an efficiency-based model. This means aggregating mechanisms to assess the impact of the integrated model actions to the network efficiency. This information can be used for decision making. Thus, the system will try not only to adjust the topology robustness and connectivity, but also to maintain the system efficiency at a desired level.
- The model implementation in a real environment setup. The possible candidates are the e-puck mobile robots and flying robots (e.g. quadcopters).
- Propose an adaptive control strategy based on a local optimization procedure.
- Improve the formalization of the robustness algorithm, in order to have the possibility of exploiting classical control system tools to formally demonstrate the desired properties.
- Obtain an *optimal control*-like formulation, that will lead to defining the best control action to guarantee connectivity and robustness, while providing the desired quality of service.
- Introduction of the clients' dynamics into the system.

8 Final remarks

Considering as the primary goal of this research the design of mechanisms for harmful topological configuration detection and mitigation, we believe that our work has been producing interesting results. Our study concerns a real and challenging problem for contemporary application requirements, with a very few approaches presented in the literature. Thus, we had to design or adapt most of the components from scratch.

Regarding specific goals, we can highlight the proposal of new approaches to deal with the vulnerability problem. Notice that each of them was published. These new proposals emerged during the design of more flexible mechanisms in terms of parametrization instantiation and the sensitivity to the network topology, but any of them is feasible to be applied in specific context. For instance, applications where the community structure is emergent, the community-based approach is able to produce more resilient networks.

As pointed out in the project proposal, the model performance dependence on the initial network state is one of the main challenges regarding our problem. It is quite difficult to design robust solutions for complex network applications: the emergent topology is not known in advance, may frequently change and is costly to evaluate. We also emphasized that, probably, our work would be targeted to a particular scope. In this sense, we are now focusing on the multi-robot systems applied to the context described in Section 6.1.

The main activities carried out so far are summarized in figure 8.

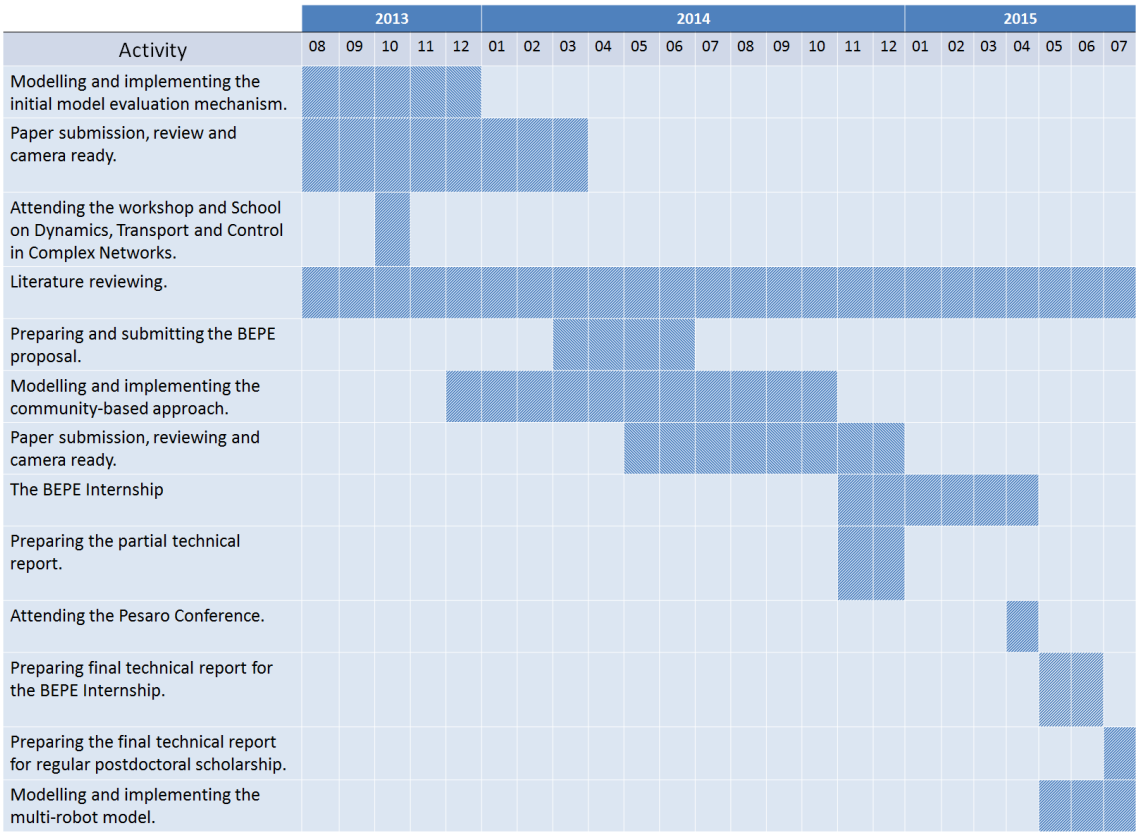


Figure 47: Activities carried out from August/2013 to July 2015.

As stressed in Section 7, there are several issues to be addressed as future work. At the moment, we are focusing on the combined model adjustment (see section 6.5). We intend to refine and to add new functionalities to the model in an incremental way.

References

[1] Réka Albert and Albert László Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47–97, January 2002.

[2] Reka Albert, Hawoong Jeong, and Albert-Laszlo Barabasi. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, July 2000.

[3] E. Almaas and A. L. Barabasi. Power laws in biological networks. In E. V. Kroonin, Y. I. Wolf, and G. P. Karev, editors, *Power laws, scale-free networks and genome biology*, pages 1–11. Springer Science, 2006.

[4] A.L. Barabási and Z. Toroczkai. Center for complex network research. online, January 2010. Network database.

[5] Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. *Dynamical processes on complex networks*. Cambridge University Press, 2008.

[6] Alireza Bigdeli, Ali Tizghadam, and Alberto Leon-Garcia. Comparison of network criticality, algebraic connectivity, and other graph metrics. In *Proceedings of the 1st Annual Workshop on Simplifying Complex Network for Practitioners*, SIMPLEX '09, pages 4:1–4:6, New York, NY, USA, 2009. ACM.

[7] Bartosz Biskupski, Jim Dowling, and Jan Sacha. Properties and mechanisms of self-organizing manet and p2p systems. *ACM Trans. Auton. Adapt. Syst.*, 2(1):1:1–1:34, 2007.

[8] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

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

[10] Pin-Yu Chen and Kwang-Cheng Chen. Information epidemics in complex networks with opportunistic links and dynamic topology. In *Global Telecommunications Conference (GLOBECOM)*, pages 1–6. IEEE, Dec 2010.

[11] Paolo Crucitti, Vito Latora, Massimo Marchiori, and Andrea Rapisarda. Efficiency of scale-free networks: error and attack tolerance. *Physica A: Statistical Mechanics and its Applications*, 320:622–642, Mar 2003.

[12] Luca Dall'Asta, Alain Barrat, Marc Barthelemy, and Alessandro Vespignani. Vulnerability of weighted networks. *Journal of Statistical Mechanics: Theory and Experiment*, April 2006:P04006, 2006.

[13] Peter S. Dodds, Roby Muhamad, and Duncan J. Watts. An experimental study of search in global social networks. *Science*, 301(5634):827–829, August 2003.

[14] P. Erdős and A. Rényi. On random graphs. *Publicationes Mathematicae Debrecen*, 6:290–297, 1959.

[15] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *In SIGCOMM*, pages 251–262, 1999.

[16] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973.

[17] Santo Fortunato. Community detection in graphs. *Physics Reports*, (3-5):75 – 174.

[18] Cinara Ghedini and Carlos H. C. Ribeiro. A framework for vulnerability management in complex networks. *Ultra Modern Telecommunications and Control Systems - The Workshop on Reliable Networks Design and Modeling International Conference on Ultra Modern Telecommunications and Control Systems*, pages 1–8, October 2009.

[19] Cinara Ghedini and Carlos H. C. Ribeiro. Improving resilience of complex networks facing attacks and failures through adaptive mechanisms. *Advances in Complex Systems*, 17(02):1450009, 2014.

[20] Cinara G. Ghedini. *A Proposal Towards Resilient Complex Networks Through Evaluation and Adaptation Mechanisms*. PhD thesis, Instituto Tecnológico de Aeronáutica - ITA, 2012.

[21] Cinara G. Ghedini and Carlos H. C. Ribeiro. Improving resilience of complex networks facing attacks and failures through adaptive mechanisms. *Advances in Complex Systems*, 17(02):1450009, 2014.

[22] Cinara G. Ghedini and Carlos H. C. Ribeiro. Using community structure information to improve complex networks robustness. In *PESARO - The Fifth International Conference on Performance*, pages 8–14. IARIA, April 2015.

[23] Alan Gibbons. *Algorithmic Graph Theory*. Cambridge University Press, July 1985.

[24] C. Godsil and G. Royle. *Algebraic Graph Theory*. Springer, 2001.

[25] James Holl and Mark S. H. An assessment of preferential attachment as a mechanism for human sexual network formation, 2003.

[26] Konstantin Klemm and Vctor M. Eguíluz. Growing scale-free networks with small-world behavior. *Physical Review E*, 65(5):057102, 2002.

[27] Dirk Koschützki, Katharina Anna Lehmann, Leon Peeters, Stefan Richter, Dagmar Tenfelde-Podehl, and Oliver Zlotowski. Centrality indices. In *Network analysis*, pages 16–61. Springer, 2005.

[28] Maciej Kurant, Patrick Thiran, and Patric Hagmann. Error and Attack Tolerance of Layered Complex Networks. *Phys. Rev. E*, 76(026103):026103, 2007.

[29] Vito Latora and Massimo Marchiori. Efficient behavior of small-world networks. *Phys. Rev. Lett.*, 87:198701, Oct 2001.

[30] Vito Latora and Massimo Marchiori. Economic small-world behavior in weighted networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 32:249–263, 2003.

[31] Keong Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim. A survey and comparison of peer-to-peer overlay network schemes. *Communications Surveys & Tutorials, IEEE*, 7:72–93, 2005.

[32] Massimo Marchiori and Vito Latora. Harmony in the small-world. *PHYSICA A*, 285:539, 2000.

[33] M. E. J. Newman. The structure and function of complex networks. *SIAM Review*, 5(2):167–256, 2003.

[34] L. Sabattini, N. Chopra, and C. Secchi. On decentralized connectivity maintenance for mobile robotic systems. In *Proceedings of the IEEE Conference on Decision and Control*, 2011.

[35] L. Sabattini, N. Chopra, and C. Secchi. Decentralized connectivity maintenance for cooperative control of mobile robotic systems. *The International Journal of Robotics Research (SAGE)*, 32(12):1411–1423, October 2013.

[36] L. Sabattini, C. Secchi, and N. Chopra. Decentralized estimation and control for preserving the strong connectivity of directed graphs. *IEEE Transactions on Cybernetics*, 2014.

[37] L. Sabattini, C. Secchi, N. Chopra, and A. Gasparri. Distributed control of multi-robot systems with global connectivity maintenance. *IEEE Transactions on Robotics*, 29(5):1326–1332, October 2013.

[38] C. Secchi, L. Sabattini, and C. Fantuzzi. Decentralized global connectivity maintenance for interconnected lagrangian systems in the presence of data corruption. *European Journal of Control*, 19(6):461–468, December 2013.

[39] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I.L. Grixa, F. Ruess, M. Suppa, and D. Burschka. Toward a fully autonomous uav: Research platform for indoor and outdoor urban search and rescue. *Robotics Automation Magazine, IEEE*, 19(3):46–56, Sept 2012.

[40] A. Vespignani V. Colizza, R. Pastor-Satorras. Reaction-diffusion processes and metapopulation models in heterogeneous networks. *Nature Physics*, (3):276–282, 2007.

[41] Fernando Vega-Redondo. *Complex Social Network*. Cambridge University Press, 2007.

[42] Stanley Wasserman, Katherine Faust, and Dawn Iacobucci. *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, November 1994.

[43] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.

[44] Duncan J. Watts. *Small Worlds : The Dynamics of Networks between Order and Randomness (Princeton Studies in Complexity)*. Princeton University Press, November 2003.

[45] Stefan Wuchty. Small worlds in rna structures. *Nucl. Acids Res.*, 31:1108–1117, 2003.

[46] P. Yang, R. A. Freeman, G. J. Gordon, K. M. Lynch, S. S. Srinivasa, and R. Sukthankar. Decentralized estimation and control of graph connectivity for mobile sensor networks. *Automatica*, 46:390–396, 2010.