A decentralised topology control to regulate global properties of complex networks

Sabato Manfredi\textsuperscript{1,2,}\textsuperscript{a} and Edmondo Di Tucci\textsuperscript{1}

\textsuperscript{1} Department of Electrical Engineering and Information Technology, University of Naples Federico II, Naples, Italy
\textsuperscript{2} Control and Power Group, Electrical and Electronic Engineering Department, Imperial College, London, UK

Received 3 October 2016 / Received in final form 29 January 2017
Published online 3 April 2017
© The Author(s) 2017. This article is published with open access at Springerlink.com

Abstract. In the last decade much research effort has been devoted to the investigation of the interplay between properties (i.e. synchronization, clustering, resilience to node fault) and topology of complex networks. Many algorithms have been proposed to construct a network topology with a given properties or to optimize them. These algorithms are static, off-line implemented and may require global network knowledge. In this paper we propose a simple decentralized topology control algorithm that by local actions carried out at the node allows to regulate network global properties. Additionally the algorithm is dynamic coping with both node and link faults and can be on-line implemented.

1 Introduction

During recent years a really active research on complex networks and their topological properties has been carried out \cite{1,2}. A network topology is represented by a graph composed of nodes (e.g. a person, an agent, a router) and links connecting them (e.g. a friendship relation, communication channel, roadway). Many indexes have been introduced in the literature to define and capture the essential features and properties of the topology of a complex network, such as diameter \cite{3,4}, the clustering coefficient \cite{5}, the path sum \cite{6,7} (just to cite a few). One of the main topological index is the network connectivity defined as the second smallest Laplacian eigenvalue $\lambda_2$. It is well known that its value characterizes how much quickly:
(i) the information spread through the network \cite{8,9}, and
(ii) the dynamical systems at nodes synchronize \cite{10,11}. Also the largest Laplacian eigenvalue $\lambda_N$ is of interest as it gives a measure of network robustness to uncertainties, time delays or fault propagation \cite{8,12}. Many artificial and natural complex networks, the topological indexes strongly affect the performance. For instance in computer networks, a specific value of path sum affects how fast the spreading of the information is, while in finance networks to set a sufficiently high cluster coefficient may confine the fault propagation to a small area closed to the faulty bank \cite{8}. Therefore it is of interest to formulate algorithms to assess a topology with a desired index value. In the literature static and dynamic algorithms have been proposed to both mime real networks (i.e. neural brains, social network, Internet network) \cite{9,13,14} and construct networks with assigned features \cite{15}. Herein with the term \textit{static algorithm} refers to a method that is applied one-shot to construct a topology with a given metric (degree distribution, connectivity). Differently, a \textit{dynamic algorithm} may act during time on the network to assess a given aim (i.e. guarantee network connectivity). For example, the well known Barabasi-Albert (BA) \cite{16} model may be seen as an algorithm for generating random scale-free networks by the preferential attachment mechanism to build a network exhibiting certain degree distribution, average path length and clustering coefficient. Other algorithms may be used to statically build networks according to the small world \cite{5} or pseudo random \cite{17} model. In \cite{15} a static algorithm has been proposed to build an efficient network topology with some connectivity properties with uniform degree, small clustering and network connectivity as large as possible. All the above algorithms are static and off-line implemented. Therefore they are inefficient if the topology changes during time (i.e. due to the fault of a link or node).

Dynamic topology control algorithms have been proposed in the literature to maximize network connectivity but they have the drawback to be centralized in the sense that they need of a global network knowledge to make a decision (i.e. add or remove a link). Moreover they try to maximize the value of the index (i.e. $\lambda_2$) without the possibility to regulate it to a predefined desired value.

In order to dynamic regulate the network topology index in the presence of random link/node fault, herein we propose on-line and decentralizable topology control algorithm. Although the proposed algorithm is locally implemented at each node, it is effective in controlling the global
network indexes to a desired value. The main differences of the algorithm with respect to existing ones in the literature are summarised in the following:

1. differently from the algorithms for the static network topology construction, the proposed algorithm is dynamic by adding and removing links during time to regulate a network index or feature to a desired set point (or target) value [11,18];
2. differently than the dynamic algorithms that try to optimize network connectivity, herein we regulate λ2 value to a desired setpoint value. This is an interesting feature because in the case of network connectivity maximization, a trivial choice is to add as many links as possible assessing a full connected network [19]: this could be expensive in terms of number of links to add;
3. the proposed algorithm is based on a random local action, resulting robust to link and node failure [18];
4. the proposed algorithm may be easily implemented by a decentralised way by using the well known consensus algorithm, differently most of existing decentralised algorithms proposed in the literature (i.e. [20]).

2 Topology control algorithm

We consider a network modelled as a graph $G(V,E,A)$ with the set of node index $V = \{1, \ldots, n\}$, set of edges $E \subseteq V \times V$, and a $n \times n$ adjacency matrix $A = [a_{ij}]$ with elements $a_{ij}$:

$$a_{ij} = \begin{cases} 1 & \text{if } j \in \mathcal{N}_i \\ 0 & \text{otherwise} \end{cases}$$

(1)

where $\mathcal{N}_i = \{ j : a_{ij} > 0 \}$ is the set of neighbours of node $i$th. $d_i = \sum_{i=1}^{n} a_{ij}$ is the node degree while the Laplacian $L$ is the matrix of elements $l_{ij}$ such that $l_{ii} = \sum_{j=1, j \neq i}^{N} a_{ij}$ and $l_{ij} = -a_{ij}$ for $i \neq j$.

The proposed algorithm defines the probability $p_{ij}$ of existence of a link between nodes $i$ and $j$ according to the following topology control law:

$$p_{ij}(k+1) = p_{ij}(k) + (1-p_{ij}(k))\xi_l(d_i, d_j)\delta(S(k) > \bar{S}) - p_{ij}(k)\xi_D(d_i, d_j)\delta(S(k) < \bar{S})$$

(2)

where $\xi_l$ and $\xi_D$ are two tunable functions depending on the network degree, $S$ is the network index to regulate and $\bar{S}$ its desired set-point value. In (2) the notation $\delta(\cdot)$ denotes a function that is one (resp. zero) if its argument is positive (resp. negative). Therefore $\delta(S(k) > \bar{S}) = 1$ (resp. $\delta(S(k) < \bar{S}) = 0$) if $S(k) > \bar{S}$ (resp. $S(k) < \bar{S}$). Similarly meaning for the term $\delta(S(k) < \bar{S})$. The algorithm defines the probability of existence $p_{ij}$ for all $(i,j) \in E$ and generate a $n \times n$ probability matrix $P$. The new network adjacency matrix $A^P$ is then generated according to a Bernoulli distribution function as $A^P = Bern(P)$ and has elements:

$$a^P_{ij} = \begin{cases} 1 & \text{with probability } p_{ij} \\ 0 & \text{with probability } 1 - p_{ij}. \end{cases}$$

(3)

Let the index $S$ be controlled to a desired value $\bar{S}$, the idea of the algorithm is to increase (or decrease) the probability of existence of a link if the value of the index $S > \bar{S}$ (or viceversa). This allows to regulate $S$ close to $\bar{S}$. In order to implement the control law $S$, each node needs to evaluate a measure (or its estimation) of the index value $S(k)$, that can be carried out in a decentralised way. In the following we will detail examples of topology indexes or properties $S$ controllable by the proposed algorithm.

1. Diameter (later shortly $DF$) defined as $S = \max_{i,j} l_{ij}$ where $l_{ij}$ is the distance between nodes $i$ and $j$. We may evaluate the value of $S(k)$ in a centralized way by knowing the adjacency matrix $A$ or in a decentralized way by using Dijkstra’s algorithm (commonly used in the routing algorithm).

2. Path sum (later shortly $PS$) defined as $S = \sum_{(i,j) \in E} l_{ij}$. It can be evaluated in a decentralized way by using a consensus algorithm on the local variable $l_{ij}$ [21,22].

3. Global efficiency (later shortly $GE$) defined as $S = \sum_{(i,j) \in E} (1/d_{ij})$. It may be evaluated in a distributed way by the average consensus algorithm on the variable $1/d_{ij}$ [23,24].

4. Network connectivity ($\lambda_2$). The evaluation of $\lambda_2$ is centralized but there are several algorithms to estimate it in a decentralized way [25–28].

5. Clustering coefficient (later shortly $CC$) defined as $S = \frac{1}{N}\sum_{i} C_i$ [5] with $C_i = \frac{\sum_{c_{i,j}} e_{i,j}}{k_i(k_i-1)/2}$, $c_{i,j}$ is the edge between nodes $i$ and $j$ with vertex $v_i$ and $v_j$ respectively, $k_i$ is the number of neighbours of $i$. Also in this case an average consensus algorithm can be used to control the value of $S(k)$ in a decentralised way at node $i$.

The probability of existence of a link also depends on the functions $\xi_l$ and $\xi_D$ that can be opportunely tuned according to different desired algorithm behaviour and specifications. It is well known that more efficient networks are characterized by a graph with uniform node degrees with few hubs or bottleneck nodes [20,29]. This means that an efficient rewriting rule has to remove links on the nodes with high degree and add links to the ones with low degree. In this respect we specifically consider the following functions:

$$\xi_l = \frac{k_i}{d_i + d_j}$$

(4a)

$$\xi_D = k_D(d_i + d_j)$$

(4b)

where $k_l$ and $k_D$ are tunable gains. In this way if $S(k) > \bar{S}$ the algorithm increases the number of connections between nodes in order to decrease $S(k)$. This is assessed by the function $\xi_l(d_i, d_j) = k_i/(d_i + d_j)$ in (2) that increases the probability of existence of a link between nodes with low degree. Conversely if $S(k) < \bar{S}$, the algorithm decreases the number of connections between nodes with higher degree by mean of the function $\xi_D(d_i, d_j) = k_D(d_i + d_j)$. In this way the index $S$ is regulated close to the set point value $\bar{S}$ and, as side effect, the network topology presents
uniform distribution of nodes degree. In other words, the idea of the algorithm is to mime the Robin Hood behaviour by “robbing from the rich node with high degree and giving to the poor node with low degree”. If an opposite-type control behaviour is required (e.g. “rich get richer”), we just need to invert the sign of the addends in (2): this is the case when GE or CC are the indexes to control. Notice that the algorithm (2) allows by local actions at each node \( i \) to control the global network index \( S \) to a desired value \( S \). This will be validated by representative simulation scenarios, and it will be subject of the next section.

### 3 Algorithm validation and tuning

In what follows we will show a variety of simulations validating the effectiveness of the proposed algorithm in several scenarios. Firstly we compare the proposed dynamic algorithm with respect to a static network topology construction proposed in [15]. Then we will validate the algorithm for larger network and in the presence of node/link fault. In this scenario we make a performance comparison than a random centralised algorithm. Finally we evaluate the algorithm responsiveness (later define as convergence time) under variations of algorithm parameters. The results may be used to tune the algorithm to cope with several scenarios. Interestingly, the final network configuration presents the same performance in terms of Entropy value independently on the controlled index. This means that the algorithm is able to guarantee the same “network navigability” property despite the controlled index.

In the performance evaluation we will consider the metrics introduced in the previous section and, in addition, the following:

- Degree (later shortly \( DE \)): the average number of links at network nodes.
- \( \lambda_2/\lambda_N \): the eigenratio of the Laplacian matrix.
- Number of links (later shortly \( NL \)): the total number of links in the network.
- Entropy (later shortly \( h \)): the entropy of the network as defined in [30].

For performance comparison purpose, we define the relative variation of an index \( I_{\text{pro}} \) obtained under the proposed topology control than the value \( I_{\text{ref}} \) obtained under a reference algorithm (i.e. static construction or random algorithm), namely: \( \Delta I = 100 \cdot \frac{I_{\text{pro}} - I_{\text{ref}}}{I_{\text{ref}}} \) in \%. In the following we will refer to the index \( \Delta I \) as a comparison metric while the proposed algorithm will be denoted in tables by the label pro.

#### 3.1 Building networks with desired properties

First of all we compare the proposed dynamic control than a static off-line algorithm construction proposed in [15]. Specifically, therein it has been considered ad-hoc optimal construction of small network of 30 nodes characterized by high value of different indexes introduced above and reported for comparison in Tables 1–3. We apply the proposed algorithm to control several indexes (i.e. path sum \( PS \), diameter \( DI \), \( \lambda_2 \) with a setpoint value \( \bar{S} \) as indicated in the original reference [15]. In Tables 1–3 are reported the metrics of the topology obtained under both the proposed and static construction algorithm when the setpoint value is: \( \bar{S} = \bar{PS} = 3 \) (Tab. 1), \( \bar{S} = \bar{DI} = 4 \) (Tab. 2) and \( \bar{S} = \bar{NL} = 1 \) (Tab. 3). The metrics are evaluated at steady state (i.e. the dynamic network control process has reached the steady state condition) by carrying out an average value over 30 realizations.

It appears from Table 1 that the proposed algorithm is able to dynamically control the \( PS \) index close to the desired value. The other (uncontrolled) indexes (i.e. degree, network connectivity) present an improved value than the initial value at \( t = 0 \). Moreover, the values of \( \lambda_2 \) and \( \lambda_2/\lambda_N \) outperform those obtained with the static algorithm [15]. This is due to the intrinsic ability of the algorithm to balance and uniform the network nodes degree by the control functions (4) with beneficial effect on most of uncontrolled network metrics. Indeed, the variance of nodes degree obtained under the proposed algorithm is low, being \( \sigma(DE) = 0.1766 \). This is made at the cost of using a larger number of links (\( NL \) in Tabs. 1–3).

#### Table 1. Control and construction of a network of 30 nodes with a setpoint index \( \bar{PS} = 3 \). First column in table presents the indexes at time \( t = 0 \); second column in table shows the indexes values obtained under the proposed algorithm \( pro \); third column in table shows the metrics obtained under the static construction procedure proposed in [15]. Last column in table shows the comparison metric \( \Delta I \) of the proposed topology control than the static network construction algorithm.

|       | \( t = 0 \) | \( \text{pro} \) | Reference [15] | \( \Delta I(\%) \) |
|-------|----------|---------------|----------------|-----------------|
| \( PS \) | 3.29690  | 2.9768        | 3              | -9.37 |
| \( DE \) | 3.1333   | 2.4843        | 2.766          | 89.74 |
| \( \lambda_2 \) | 0.0990  | 1.0837        | 1              | 8.37 |
| \( \lambda_2/\lambda_N \) | 0.0106 | 0.1184        | 0.167          | -29.10 |
| \( DI \) | 8        | 4.5425        | 4              | 13.56 |
| \( NL \) | 46.999   | 78.724        | 45             | 74.94 |
| \( h(\alpha = 1) \) | 2.717  | 1.878         | 1.099          | 70.88 |

#### Table 2. Control and construction of a network of 30 nodes with a setpoint index \( \bar{DI} = 4 \). First column in table presents the indexes at time \( t = 0 \); second column in table shows the indexes values obtained under the proposed algorithm \( pro \); third column in table shows the metrics obtained under the static construction procedure proposed in [15]. Last column in table shows the comparison metric \( \Delta I \) of the proposed topology control than the static network construction algorithm.

|       | \( t = 0 \) | \( \text{pro} \) | Reference [15] | \( \Delta I(\%) \) |
|-------|----------|---------------|----------------|-----------------|
| \( PS \) | 3.29690  | 2.0457        | 3              | -31.81 |
| \( DE \) | 3.1333   | 5.9269        | 2.766          | 114.27 |
| \( \lambda_2 \) | 0.0990  | 1.3239        | 3              | 32.39 |
| \( \lambda_2/\lambda_N \) | 0.0106 | 0.1065        | 0.167          | -36.23 |
| \( DI \) | 8        | 3.6419        | 4              | -8.95 |
| \( NL \) | 46.999   | 88.903        | 45             | 97.56 |
| \( h(\alpha = 1) \) | 2.717  | 1.891         | 1.099          | 72.07 |
Similar considerations may be stated when the controlled index is the diameter $DI$ and network connectivity $\lambda_2$ as shown by the results in Tables 2 and 3, respectively.

Additionally it has been evaluated the comparison metric $\Delta I$ to compare the metrics obtained under the proposed algorithm with respect to the static algorithm in [15]. The results in Tables 1–3 highlight the good set point regulation of the controlled indexes (i.e. $PS$, $DI$, $\lambda_2$). For the uncontrolled indexes we may observe several trends. Specifically, the indexes strictly related to the connectivity (i.e. $\lambda_2$, $\lambda_2/\lambda_N$, $DE$) and entropy (i.e. $h$) are substantial better under the proposed algorithm; the number of links $NL$ is higher for the proposed topology control, but this effect is mitigated on larger network as it will pointed out in the next simulations; the rest of indexes present a slightly difference and behaviour (i.e. $|\Delta I| \leq 10\%$).

Finally we evaluate the navigability of the topology obtained by the proposed decentralized algorithm. We refer to the general formulation of the entropy for a network with random walks [30] defined as:

$$\pi_{ij} = \frac{a_{ij}d_i^\alpha}{\sum_j a_{ij}d_j^\alpha}$$  \hspace{1cm} (5a)

$$h = -\sum_{i,j} \pi_{ij} w^*_i \ln(\pi_{ij})$$  \hspace{1cm} (5b)

where $w^*_i$ is the stationary distribution of the Markov chain described by (5a) (see [30] for further details).

Entropy is a measure of a diffusion process on a network. High entropy rate indicates a large randomness, or easiness of propagating from one node to another, and can be related to an efficient spreading of the information over the network. By changing the value of the exponent $\alpha$ we can tune the dependence of the diffusion process on the node property degree $d_i$. When $\alpha \neq 0$ we are introducing in the random walk of the particle a bias towards neighbour nodes with high $d_i$ ($\alpha > 0$) or low $d_i$ ($\alpha < 0$). For $\alpha = 0$, the standard unbiased random walk is obtained.

In Figure 1 the entropy of the topology obtained controlling different indexes (i.e. $PS$, $DI$, $\lambda_2$) is shown under different values of $\alpha$. Interestingly, the entropy value is almost independent from the particular controlled index. In this way, the topology has the same navigability property independently from the controlled index and the strategy used to visit the network nodes.

### 3.2 Performance over large networks

We consider a more realistic scenario when the network is composed of 50 and 100 nodes. The results are shown in Table 4 and confirm the effectiveness of the algorithm to control the index to the desired setpoint value. Notice that due to the intrinsic stochastic nature of the proposed algorithm, better performance are obtained for larger networks. For instance the controlled index $PS$ is closer to the set point $PS$ for a network of 100 nodes than one of 50. Therefore the proposed algorithm may be suitable for network of practical interest (e.g. engineering network, social network) for which the size is large or may increase during time.

![Fig. 1](image-url)  

**Table 3.** Control and construction of a network of 30 nodes with a setpoint index $\lambda_2 = 1$. First column in table presents the indexes at time $t = 0$; second column in table shows the indexes values obtained under the proposed algorithm pro; third column in table shows the metrics obtained under the static construction procedure proposed in [15]. Last column in table shows the comparison metric $\Delta I$ of the proposed control than the static network construction algorithm.

| $t = 0$ | pro | Reference [15] | $\Delta I(\%)$ |
|---------|------|----------------|----------------|
| $PS$    | 3.2690 | 2.6582 | -11.39 |
| $DE$    | 3.1333 | 4.8219 | 74.33 |
| $\lambda_2$ | 0.0990 | 0.9813 | 1 - 1.87 |
| $\lambda_2/\lambda_N$ | 0.0106 | 0.1149 | 0.167 -31.20 |
| $DI$    | 8     | 3.9065 | -2.34 |
| $NL$    | 46.999 | 72.329 | 45 60.73 |
| $h(\alpha = 1)$ | 2.717 | 1.864 | 1.099 69.61 |

**Table 4.** Control and construction of a network of 50 and 100 nodes respectively with a setpoint index $PS = 2.8$. First and third columns in table presents the indexes values obtained under the proposed algorithm pro; second and fourth columns in table shows the comparison metric $\Delta I$ of the proposed control than the random algorithm.

|            | 50 nodes | $\Delta I(\%)$ | 100 nodes | $\Delta I(\%)$ |
|------------|----------|----------------|-----------|----------------|
| $PS$       | 2.7368   | -2.93          | 2.7803    | -0.81          |
| $DE$       | 4.2619   | -6.31          | 5.8934    | -1.65          |
| $\lambda_2$ | 0.9652 | 29.13          | 1.2540    | 26.32          |
| $\lambda_2/\lambda_N$ | 0.0963 | 23.74          | 0.1382    | 47.09          |
| $DI$       | 4.9677   | -33.22         | 4.5484    | -52.58         |
| $NL$       | 106.55   | -6.31          | 294.67    | -1.65          |
In Figure 2 it is reported the dynamic evolution of the path sum $PS$ when the setpoint $\bar{N} = \overline{PS} = 2.8$. The algorithm adds links till the index value is decreased to the desired setpoint value (in about 500 time steps). At steady state, the algorithm is able to guarantee the setpoint tracking.

Now we will validate the proposed topology control algorithm when a different kind of index is controlled. Specifically we control the global efficiency $GE$ and clustering coefficient $CC$ index. To this aim we need to have a “rich get richer” behaviour of the controlled network by inverting the sign of the addends in (2).

The results in Table 5 show the effectiveness of the algorithm to track the desired value of $GE$.

Similarly, the clustering coefficient index is efficiently controlled to the desired value as shown by the results in Table 6.

Table 7 shows the simulation results when the network connectivity $\lambda_2$ is the controlled index.

For sake of comparison, we implement a simple centralised algorithm (later random algorithm) that randomly adds links in order to get the desired index value.

### Table 5. Control and construction of a network of 50 with a setpoint index $GE = 0.52$ and 100 nodes with a setpoint index $GE = 0.40$. First and third columns in table presents the indexes values obtained under the proposed algorithm $pro$; second and fourth columns in table shows the comparison metric $\Delta I$ of the proposed control than the random algorithm.

| $GE$ | $50$ nodes | $100$ nodes | $\Delta I$ (%) | $pro$ | $\Delta I$ (%) |
|------|-------------|-------------|----------------|------|----------------|
| $GE$ | 0.5125      | 0.4151      | -0.93          | 0.1525 | -0.019 |
| $PS$ | 2.2119      | 2.7020      | 0.25           | 2.2119 | -0.14 |
| $DE$ | 6.8387      | 6.2458      | 3.03           | 6.8387 | -2.15 |
| $\lambda_2$ | 5.81 | 0.9033 | 2.42 |
| $\lambda_2/\lambda_N$ | 0.1012  | 0.0591 | -4.86 |
| $DI$ | 4.1613      | 5.1613      | -32.91         | 4.1613 | -3.88 |
| $NL$ | 170.97      | 312.29      | -3.03          | 170.97 | -2.15 |

### Table 6. Control and construction of a network of 50 and 100 nodes respectively with a setpoint index $\lambda_2 = 1.5$. First and third columns in table presents the indexes values obtained under the proposed algorithm $pro$; second and fourth columns in table shows the comparison metric $\Delta I$ of the proposed control than the random algorithm.

| $\lambda_2/\lambda_N$ | $50$ nodes | $100$ nodes | $\Delta I$ (%) | $pro$ | $\Delta I$ (%) |
|-----------------------|-------------|-------------|----------------|------|----------------|
| $PS$                  | 2.1992      | 2.4124      | 7.32           | 2.1992 | -8.35 |
| $DE$                  | 6.7742      | 8.1465      | -1.80          | 6.7742 | -11.65 |
| $\lambda_2$          | 1.4711      | 1.5176      | -1.22          | 1.4711 | 0.08 |
| $\lambda_2/\lambda_N$ | 0.1311  | 0.0863 | 4.56 |
| $DI$                  | 4.9023      | 4.1935      | -7.20          | 4.9023 | -24.32 |
| $NL$                  | 169.35      | 407.32      | -1.80          | 169.35 | 11.65 |

We compare the metrics of the topology obtained under the proposed algorithm than ones of the random algorithm by evaluating the comparison metric $\Delta I$. From the results in Tables 4–7 it appears that for most of indexes the performance are comparable ($|\Delta I| \leq 10\%$). Moreover, the topology obtained by the proposed algorithm presents better network connectivity $\lambda_2$, eigenratio $\lambda_2/\lambda_N$, and lower diameter. Notice that in this case of larger network, the number of links used by both algorithms is almost the same.

### 3.3 Properties preservation under node/link fault

Herein we point out how the proposed algorithm is dynamic and allows to regulate the index to a desired value despite the presence of link and node faults, differently than the random algorithm.

In other terms, the proposed algorithm is robust with respect to failure of links or nodes. In Figure 3 is depicted the dynamic evaluation of the path sum with a setpoint value of 2.8 when the fault of the 10% of the links occurs at 1000 time steps. The effect of the fault is to increase the path sum due to the increase of the length of shortest paths. The proposed algorithm copes with the fault and guarantees the index regulation to the setpoint. Differently the random algorithm is unable to guarantee the setpoint regulation as the path sum definitely increases (yellow line...
The path sum (yellow continuous line in Fig. 4) increases scenario and remove up to 30% of nodes at 1000 unit steps. nodes fault. We firstly consider the previous network scenarios: (1) fault of 30% of the nodes at time 1000 (yellow continuous line); (2) addition of 30% of nodes at time 1000 (blue continuous line). Dynamic evolution of the path sum for a network of 100 nodes and $PS = 2.8$ when at time 1000 the 10% of links fault occurs. Dynamic evolution of the path sum for a network of 100 nodes and $PS = 2.8$ (red dashed line in Fig. 3). Similar performance are obtained when a 30% of nodes has been added as shown in Figure 4 (blue continuous line). The path sum value firstly decreases due to the addition of new nodes and related links, and finally is controlled to the setpoint value.

3.4 Performance under limited range of action

The proposed algorithm is implemented assuming that each node may add a link to any other node. This assumption is realistic for small networks or for infrastructure (i.e. computer networks) for which is possible to have a link between any nodes at relatively low cost. In spacial-distributed complex networks, nodes have a specific location and links between them have a cost proportional to the distances between them. In other scenarios a node may be constrained to connect with neighbours within a specific range of action (i.e. wireless communications) [31–33].

In the latter cases, the applicability of the proposed algorithm is constrained to a limited range of action. In order to model this behaviour, let $p_i = [x_i, y_i]$ the coordinates of node $i$, $D_{ij}$ the euclidean distance between nodes $i$ and $j$, and $R$ the nodes transmission or connection radius, the elements of adjacency matrix are recast as:

$$a_{ij} = \begin{cases} 
1 & \text{if } j \in \mathcal{N}_i \text{ and } r_{ij} \leq R \\
0 & \text{otherwise.}
\end{cases}$$

(6)

Obviously when $R$ is set as the maximum network distance, we have the scenarios previously analysed. Interestingly there is a trade off between the value of $R$ (i.e. node visibility or cost to build link of maximum range $R$) and time to control the index to its setpoint value. The latter time is referred as convergence time $t_c$ and it is defined as the time at which the relative error of the controlled index than the setpoint value is below the 5%. We may tune the algorithm parameters $k_I$ and $k_D$ in equations (4) in order to affect the convergence time for a given value of $R$. Higher values of $k_I$ and $k_D$ improve algorithm responsiveness and so the convergence time is reduced.

In order to evaluate the trade-off between the convergence time $t_c$ and the transmission radius $R$, we compute $t_c$ for a network of 100 nodes on a square lattice with area $L^2$. $L = 30$ under variation of $R \in [0.2 \sqrt{2L}, 2L]$. In this way the network topology ranges from being disconnected to be full connected. In Figure 5 the convergence time $t_c$ versus the relative radius (i.e. $R/L$) is depicted when the controlled index is the path sum and $k_I = 10^{-2}$ and $k_D = 10^{-4}$ (red line). Notice that for $R/L < 0.38$, the network is disconnected and $t_c$ is not valuable. For $R/L \in [0.38, 1)$, $t_c$ rapidly decreases at a lower rate for $R/L = 0.6$. 

In Figure 5 are also shown the results of convergence time for different values of $k_I$ and $k_D$. As expected, higher (resp. lower) values of $k_I$ and $k_D$ improves (resp. deteriorate) the convergence time for a given $R$. The results in Figure 5 allow to tune $R$, $k_I$ and $k_D$ to have a desired convergence time under cost constraints (i.e. length of links).
In this paper we proposed a simple decentralized topology control algorithm that by local actions carried out at nodes allow to regulate complex network global properties. The main features of the proposed algorithm are to be dynamic and decentralized. Additionally it copes with both nodes and links fault. A performance comparison of the proposed topology control algorithm than a static network construction and random algorithms has been carried out. The simulation results show the effectiveness of the proposed algorithm to regulate the complex network indexes to given setpoint values.

**Author contribution statement**

Sabato Manfredi contributed by the formulation of the proposed control approach, the writing and organization of the paper. Edmondo Di Tucci contributed the numerical simulations and manuscript preparation by constructive discussions.

**Open Access** This is an open access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

**References**

1. L. da F. Costa, F.A. Rodrigues, G. Travieso, P.R.V. Boas, Adv. Phys. 56, 167 (2007)
2. S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.U. Hwang, Phys. Rep. 424, 175 (2006)
3. R. Hulsermann, A. Betker, M. Jager, S. Bodamer, M. Barry, J. Spalth, C. Gauger, M. Kohn, ITG FACHBERICHT 182, 65 (2004)
4. F. Constantinou, P. Mavrommatis, Identifying known and unknown peer-to-peer traffic, in Network Computing and Applications, 2006. NCA 2006. Fifth IEEE International Symposium on (IEEE, 2006), pp. 93–102
5. D.J. Watts, S.H. Strogatz, Nature 393, 440 (1998)
6. A.L. Barabasi, Z.N. Oltvai, Nat. Rev. Genetics 5, 101 (2004)
7. A. Fronczak, P. Fronczak, J.A. Holyst, Phys. Rev. E 70, 056110 (2004)
8. A. Krause, S. Giansante, J. Economic Behavior & Organization 83, 583 (2012)
9. A. Ganesh, L. Massoulié, D. Towsley, The effect of network topology on the spread of epidemics, in INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE (IEEE, 2005), Vol. 2, pp. 1455–1466
10. S. Manfredi, D. Angeli, IEEE Transactions on Automatic Control, DOI: 10.1109/TAC.2016.2598850
11. L. Donetti, P.I. Hurtado, M.A. Muñoz, Phys. Rev. Lett. 95, 188701 (2005)
12. S. Markose, S. Giansante, A.R. Shagaghhi, Journal of Economic Behavior & Organization 83, 627 (2012)
13. G. Fagiolo, M. Mastrorillo, Phys. Rev. E 88, 012812 (2013)
14. D.S. Bassett, E.T. Owens, K.E. Daniels, M.A. Porter, Phys. Rev. E 86, 041306 (2012)
15. D. Shi, G. Chen, W. Thong, X. Yan, Circuits and Systems Magazine, IEEE 13, 66 (2013)
16. R. Albert, A. Barabási, Rev. Mod. Phys. 74, 47 (2002)
17. P. Erdös, A. Rényi, Publ. Math. Instrum. Hungar. Acad. Sci. 5, 17 (1960)
18. M. Dadashi, I. Barjasteh, M. Jalili, Chaos 20, 043119 (2010)
19. D. Xu, Y. Li, T.J. Wu, Physica A 382, 722 (2007)
20. M. Jalili, Neural Networks and Learning Systems IEEE Trans. 24, 1009 (2013)
21. S. Manfredi, Control Engineering Practice 21, 381 (2013)
22. R. Olfati-Saber, R.M. Murray, IEEE Trans. Automatic Control 49, 1520 (2004)
23. S. Manfredi, D. Angeli, Automatica 77, 51 (2017)
24. S. Manfredi, D. Angeli, Automatica 64, 182 (2016)
25. F. Knorn, R. Stanojevic, M. Corless, R. Shorten, Int. J. Control 82, 2095 (2009)
26. P. Yang, R. Freeman, G. Gordon, K. Lynch, S. Srinivasa, R. Sukthankar, Automatica 46, 390 (2010)
27. L. Sabattini, C. Secchi, N. Chopra, Decentralized control for maintenance of strong connectivity for directed graphs, in Control Automation (MED), 2013 21st Mediterranean Conference on (IEEE, 2013), pp. 978–986
28. S. Manfredi, E.D. Tucci, Int. J. Control, DOI: 10.1080/00207179.2016.1201218
29. T. Nishikawa, A.E. Motter, Y. Lai, F.C. Hoppensteadt, Phys. Rev. Lett. 91, 014101 (2003)
30. J. Gómez-Gardeñes, V. Latora, Phys. Rev. E 78, 065102 (2008)
31. S. Manfredi, Ad Hoc Networks 13, 234 (2014)
32. S. Manfredi, IEEE Wireless Commun. 21, 81 (2014)
33. S. Manfredi, Ad Hoc Networks 11, 1942 (2012)