Safeguarding Structural Controllability in Cyber-Physical Control Systems

Cristina Alcaraz(✉) and Javier Lopez

Department of Computer Science, University of Malaga,
Campus de Teatinos s/n, 29071 Malaga, Spain
{alcaraz,jlm}@lcc.uma.es

Abstract. Automatic restoration of control wireless networks based on dynamic cyber-physical systems has become a hot topic in recent years, since most of their elements tend to have serious vulnerabilities that may be exploited by attackers. In fact, any exploitation may rapidly extend to the entire control network due to its problem of non-locality, where control properties of a system and its *structural controllability* can disintegrate over time. Unfortunately, automated self-healing processes may become costly procedures in which the reliability of the strategies and the time-critical of any recovery of the control can become key factors to re-establish the control properties in due time. This operational need is precisely the aim of this paper, in which four *reachability-based recovery* strategies from a theoretical point of view are proposed so as to find the best option/s in terms of optimization, robustness and complexity. To do this, new definitions related to structural controllability in relation to the type of distribution of the network and its control load capacity are given in this paper, resulting in an interesting practical study.

Keywords: Structural controllability · Control systems · Cyber-physical systems · Restoration · Self-healing

1 Introduction

As control systems continue to grow both in size and complexity [1] by adapting the new cyber-physical systems (CPSs) for the automation of operations, the protection of such networks from external or unforeseen forces becomes an essential issue. Namely, operational efficiency has an important role to play in the monitoring and management of many of our critical infrastructures (CIs) such as industrial automation applications or power grids. Unfortunately such functionality today is highly susceptible to threats and/or changes. Many of these changes come from vulnerabilities or incompatibilities of the cyber-physical control elements, which tend to incorporate and connect computation elements with existing physical components [2,3] through multiple types of communication technologies like, for example, wireless [4]. However, the exploitation of these vulnerabilities is also intertwined with the nature of the threats, which may

I. Askoxylakis et al. (Eds.): ESORICS 2016, Part II, LNCS 9879, pp. 471–489, 2016.
DOI: 10.1007/978-3-319-45741-3_24

sometimes cause a minor or even, major impact on the performance, security and safety of the underlying infrastructures [5].

In these circumstances it is easy to understand that preventive measures related to resilience and fault-tolerance have to be properly addressed in critical environments [6], regardless of the fact that some measures can become quite difficult to implement [7,8]. For example, the mere act of helping restore large and complex control distributions to their natural state in time, might provoke serious complexities that may subsequently affect the overall performance of the system. So it becomes crucial to research how to design optimized recovery mechanisms that can 'automatically' establish connectivity of control from anywhere and at any time. However, the implementation of large control networks can also be quite costly from a research point of view. This means that the modeling and simulation of the challenge (taking into account the network topology and the nature of its distribution), have to be done through *graph theory*.

Within the literature some authors have already tried to address restoration topics through graph theory. For example, Nakayama *et al.* base their research on tie-set notions, associated with graphical-theoretical tree structures so as to implement a ring-based solution against link failures [9]. A variant of this solution is the rapid spanning tree protocol (RSTP), an evolution of the spanning tree protocol (STP), to manage traffic loops and broadcast congestion in mesh topologies [10]. Tree-like structures are also applied to group and activate, through a nice tree decomposition, backup instances of driver nodes in charge of delivering control signals to the rest of nodes of the network [11], or to build edge-redundant networks to activate backup links [11–13]. Médard *et al.* in [12] support their approach on two trees so that the removing of any resource leaves each destination connected to one of the directed trees; whereas Quattrociocchi *et al.* in [13] center their study on modeling a routing protocol based on the maximum spanning tree and on the online activation of fixed redundant links. Likewise, Wang *et al.* apply the redundancy concept in the controllability field by applying transitivity of control routes, taking into account a control robustness index with reliance on the number of driver nodes [14]. Wang *et al.* in [15] and Ding *et al.* in [16] also propose optimizing the robustness of controllability by adding a minimum number of strategic links within the network.

However, more research on dynamic preservation of control structural properties for critical environments is still required since most of these approaches are composed of static structures for the recovery, and/or are centered on the restoration of general-purpose networks. Indeed, the vast majority of the critical control systems follow particular topological structures of the type power-law $y \propto x^{\alpha}$, [17], whose structures tend to produce small sub-networks similar to current control substations. Moreover, this research shortfall also forces us to think that it is necessary to propose specific restoration strategies that help the underlying system (i) maintain its control properties at all times and (ii) survive in crisis situations. So, four restoration strategies for *structural controllability* are presented in this paper. They are based on the automatic activation of redundant edges so as to exhibit the optimal scenario, and on the dynamic reachability of nodes

through relink techniques together with a further set of parameters described throughout this paper. To complement this study, analyses on which of these approaches are the most suitable for critical contexts with heavy dependence on CPSs are also presented, thereby complying with optimization aspects.

In order to clarify some theoretical concepts introduced in the following sections and their relationships with respect to the main goals and contributions of this paper, topics related to structural controllability and power dominance are described here. The concept of structural controllability was introduced by Lin in 1974 [18] so as to model the controllability and its control capacity through graphical representations, where the control is generally associated with a subset of nodes with the maximum capacity of dominance. This subset of nodes, also known as driver nodes and denoted here as N_D, has to be selected according to a predefined method based on the type of context and the general structure of its networks; in our case, attending to power-law control networks. A suitable method is, for example, the Power Dominating Set (PDS) problem defined by Haynes *et al.* in [19] rather than the traditional maximum matching method. Through PDS it is possible to obtain the set of N_D in charge of managing the control of the entire or a supart of the network, whose concept was originally designed as a variant of the well-studied problem of domination and motivated in part by the structure of electric power networks and their monitoring networks [19]. Therefore these two concepts, structural controllability and PDS, constitute the theoretical basis of our research, and the goal now is to provide a redundancy-based restoration layer with the possibility of reaching linear complexities in optimal restoration scenarios.

The remainder of this paper is structured as follows: Sect. 2 outlines preliminary concepts concerning dynamic control networks, in addition to detailing the initial assumptions and the threat model. Section 3 presents the four recovery strategies together with their redundancy principles, which are theoretically developed and discussed in Sect. 4. Finally, Sect. 5 concludes the paper and presents future work.

2 Dynamic Control: Preliminary

Let a directed weighted $\mathscr{G}_w(V, E)$ graph represent the construction of a control system composed of V control nodes corresponding to cyber-physical elements, and E communication links. Through $\mathscr{G}_w(V, E)$, it is possible to characterize dynamic control networks capable of accepting the existence of *loops* and *weighted edges* to plot control loads related to controllability. In the real-world, many of these variables traverse specific links that help control devices (or driver nodes), such as remote terminal units or gateways in charge of managing sensor or actuator states, to be reached. This in turn recreates a decentralized system where the main control exclusively depends on a dominant subset of elemental nodes and links. Concretely, these links contain the maximum capacity to

conduct the main traffic[1] between two points, also defined here as the *control load capacity* (CLC).

To represent this capacity it is necessary to work with a weighted decentralized system containing information about the edge betweeness centrality (EBC) [5]. EBC is an indicator that corresponds to the sum of the fraction of the shortest paths that pass through a given edge, such that, edges with the highest centrality participate in a large number of shortest paths. The result is a weighted matrix related to $\mathscr{G}_w(V, E)$ whose weights are computed as follows:

$$E_{BC}(e) = \sum_{s,t \in V} \frac{\delta(s, t \mid e)}{\delta(s, t)} \tag{1}$$

where $\delta(s, t)$ denotes the number of shortest (s,t)-paths and $\delta(s, t \mid e)$ the number of paths passing through the edge e. Hence, CLC in control theory corresponds to the traditional weighted interaction strength matrix $\mathbf{A}$ [5] supported by the linear time-invariant (LTI) dynamical system introduced by Kalman in [20]:

$$\dot{x}(t) = \mathbf{A}x(t) + \mathbf{B}u(t), \qquad x(t_0) \;=\; x_0 \tag{2}$$

From this equation, $\dot{x}(t)$ comprises the vector $(x_1(t), \ldots, x_n(t))^T$ containing the current state of n nodes at time t; $\mathbf{A}$, the network topology with the interaction strength $(n \times n)$; and $\mathbf{B}$ an *input* matrix $(n \times m,\, m \leq n)$ holding the set of driver nodes, controlled by a time-dependent *input vector* $u(t) = (u_1(t), \ldots, u_m(t))$ responsible for forcing the system to reach a desired configuration state. The system in Eq. 2 is *controllable* if and only if $\text{rank}[\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, \ldots, \mathbf{A}^{n-1}\mathbf{B}] = n$ (Kalman's rank criterion). However, whilst the computation of this equation seems to be straightforward, for large and heterogeneous networks like CPSs embedded in control systems where the number of nodes grows exponentially (e.g., sensors, actuators, smart meters, remote units or hand-led interfaces), it becomes extremely expensive and problematic. So the problem associated with maintaining weights in $\mathbf{A}$ and the exponential growth of nodes leads to a new control theory known as structural controllability [18], which is described in more detail below.

2.1 Structural Controllability and its CLC

Structural controllability refers to a graphical-theoretical interpretation of the style $\mathscr{G}_w(\mathbf{A},\, \mathbf{B}) = (V, E)$ where $\mathbf{A}$ and $\mathbf{B}$ contain non-zero weights, such that $V = V_{\mathbf{A}} \cup V_{\mathbf{B}}$ comprises the set of vertices and $E = E_{\mathbf{A}} \cup E_{\mathbf{B}}$ the set of edges. In this representation, $V_{\mathbf{B}}$, analogous to $u(t)$ in Eq. 2, embraces all those nodes with the capacity to inject control signals throughout the entire network, which is composed of different control load capacities, $l_{i,j}$, for each edge $e_{i,j}$ in E (i.e., $l_{i,j}$ is part of the concept of CLC).

[1] Note that we do not consider in this study either the type of traffic or the content of messages, only those concepts that help define mechanisms of restoration.

As indicated above, there are two main approaches that obtain the minimum, but not the only set of driver nodes associated with $V_{\mathbf{B}}$: the maximum matching and the PDS. In graph theory, the former aims to obtain N_D (unmatched nodes) by identifying those nodes that do not share input vertices [21]. Although the concept has been proven multiple times [5,14,15], we primarily focus on the PDS problem by offering the necessary means to exemplify, through graph theory, structures similar to real power grids and their monitoring systems, and whose concept corresponds to an extension of the Dominating Set (DS). From the original formulation of the PDS, given by Haynes *et al.* in [19], the problem was later simplified into two fundamental observation rules by Kneis *et al.* in [22]. These two rules, substantiated on the 'dominance' concept, are as follows:

OR1 *A vertex in N_D observes itself and all its neighbors*, complying with DS.
OR2 *If an observed vertex v of degree $d^+ \geq 2$ is adjacent to $d-1$ observed vertices, the remaining un-observed vertex becomes observed as well.* This also implies that **OR1** $\subseteq$ **OR2** given that the subset of nodes that comply with **OR1** becomes part of the set of nodes that complies with **OR2**.

Both rules and their susceptibility to threats have also been analyzed in recent publications [1,23], and for different types of graphs under the restriction of degree and specific graph structures (circle, planar, split, and partial k-tree graphs as well as grid and blocks). However, and as previously mentioned, we are not interested in applying the concept of PDS in general distributions. Rather our interest lies in applying the PDS problem in power-law networks since most of the topologies of CIs follow similar structures to $y \propto x^{\alpha}$ [17].

The pursuit of all these methods and their application as a whole results in a complex control structure supported by E_{BC} to establish control loads. The handling of anomalous loads is done through the definition given by Nie *et al.* in [5], in which the capacity of a node, $l_{i,j}$, is always bounded to "*the maximum load that the edge, $e_{i,j}$, can operate*". In normal situations, $l_{i,j}$, has to be related to the initial capacity, denoted here as $L^0_{i,j}$ $(n \times n)$, and depending on the type of activity within the network and the overloading of the links, the initial state of the network may significantly vary over time. Therefore, the load capacity has to be managed each time by verifying that $l_{i,j}$ does not exceed *maximum CLC* [5]:

$$H_{i,j} = (1+\alpha) \times L^0_{i,j} \tag{3}$$

of size $n \times n$, where α comprises a tolerance indicator with value $\alpha > 0$ and $L^{t=0}_{i,j} \leq L^{t>0}_{i,j} \leq H_{i,j}$. Under these conditions, any topological impact may force the system not only to redistribute its control loads, but also its shortest paths, thereby affecting, sooner or later, the network diameter. This could also trigger a *cascading effect* when the permitted thresholds, retained in $H_{i,j}$, are clearly surpassed. Given this and its importance for control contexts, the following section provides a set of initial assumptions required for dynamic control restoration together with the threat model.

2.2 Initial Assumptions and the Adversarial Model

Apart from cyclicity between nodes, the existence of $l_{i,j}$ in each $e_{i,j}$ and $l_{i,j} \leq H_{i,j}$, the two observation rules (**OR1**, **OR2**) introduced in the previous section must not be violated at any moment. In relation to this, the number of driver nodes should not increase significantly during the life cycle of the network, maintaining, as much as possible, its spatial complexity. This also means that no protection approach should hamper the control processes and the responsiveness degree of the system, while still providing the necessary means to self-heal the control in time, with a reasonable computational cost.

For the analysis, the adversary model follows a weak model in which adversaries are able to access the general structure of the graph, its topology and the location of the current driver nodes, despite the random nature of N_D. We also assume that their mobility within the network and their performances remain reduced to a random subset of nodes, such that $\delta \leq \frac{|V|}{2}$, where their actions are focused on availability and integrity of assets, composed of random (launch random actions on an arbitrary set of nodes) or targeted attacks (specific actions on particular nodes).

Within the random category, four attacks are highlighted:

- [**R1**] isolate a selective set of nodes by removing all their edges (e.g., jamming);
- [**R2**] arbitrarily choose some nodes and remove a few, but not all, of their edges (e.g., obstacles, congestion);
- [**R3**] randomly insert a limited set of nodes whose links are causally created; and
- [**R4**] arbitrarily add new edges within the network.

In real scenarios, there also exists the possibility of finding mobile automation contexts in which nodes do not necessarily have to be compromised. They may, for example, (i) leave a network by themselves (henceforth denoted as [**Lv**]) by simply removing all their connections, or (ii) join the network, by themselves, by increasing the number of members and links. To tackle these two new situations, we consider the definition of [**R1**] but without applying preventive measures to avoid the re-connection, and [**R3**] to engage the new joining.

With respect to the targeted class, four kinds of attacks can be identified:

- [**T1**] isolate those nodes with the highest degree, i.e., the hubs;
- [**T2**] isolate the node with the highest strength within the network, equivalent to the node with the highest CLC $- max(\sum_{i \in E}(E_{EB}(v,i) + E_{EB}(i,v)))$; and
- [**T3**] remove an arbitrary set of δ links with the highest peaks of centrality.

3 Four Reachability-Based Strategies

Reachability of assets and their maintenance can be achieved through four types of reconnection approaches, the strategies of which aim to find redundant pathways, for each disconnected vertex $v_i \in V$. For the relink, the approaches force the system to first identify those most prominent $\{n_{d_1}, \ldots, n_{d_n}\} \in N_D$, such that:

STG1 Select one "brother" n_d located in the surrounding area, such that $(n_d, v_i) \notin E$, but there exists a common node $v_j \in E$ where $(v_j, n_d) \in E$ and $(v_j, v_i) \in E$, and it may serve as a possible candidate to establish a new redundant relationship $(n_d, v_i) \in E$. Note that the selection of prominent nodes is restricted to the redundancy principles described below.

STG2 Choose one "father" n_d with the capacity for reconnecting $(n_d, v_i) \in E$.

STG3 Take one "grandfather" n_d located to 2-hops with the ability to relink v_i.

STG4 Select one "remote" n_d situated at n-hops with the possibility of relinking v_i in crisis situation.

If we observe Fig. 1, it is possible to see that the first three scenarios (**STG**x ($x = \{1, 2, 3\}$)) establish a protection on a local level, whereas **STG4** addresses the protection for a remote level in which the selection of outstanding driver nodes relies on the minimum diameter, using for this the traditional *breadth-first search* (BFS) method. Each link represents the control load capacity between two points, $l_{i,j}$; and when a node has different paths (e.g., x, y, z) to transmit a critical message until reaching j, then it is necessary to choose the path with the highest load capacity: $max\{l_{i,x}, l_{i,y}, l_{i,z}\}$. For the mapping of secondary routes, it is also necessary to redesign the **OR1** and **OR2** algorithms specified in [1], not only to select the best driver candidates but also to introduce, from the initial stage (the commissioning phase), redundant pathways. This modification involves:

– expanding the DS selection scheme (**OR1** included in [1]) by adding redundant links; and

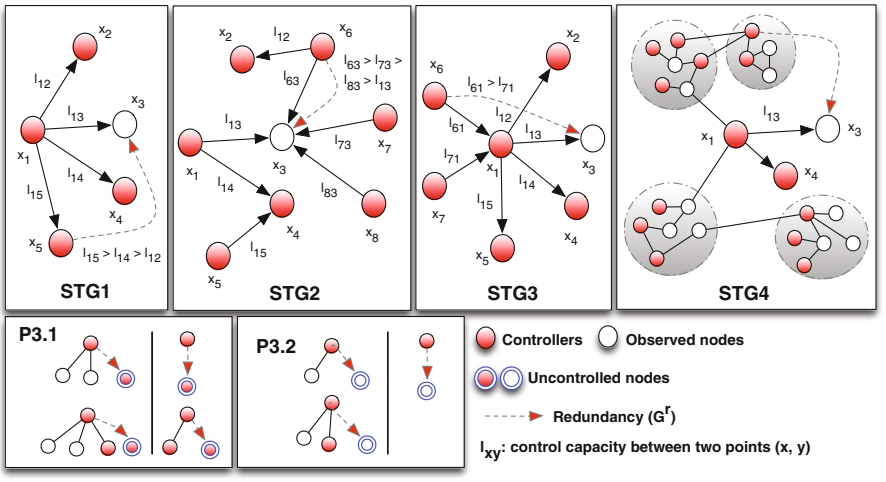

Fig. 1. Restoration scenarios **STG**x ($x = \{1, 2, 3, 4\}$) and redundancy principles **P3.1** and **P3.2**

– extending the approach **OR2** from [1] so as to avoid breaking the second observation rule due to the existence of new links.

Given this, the next section specifies the new approaches of **OR1** and **OR2**, since they constitute the foundation of the new restoration strategies.

Algorithm 3.1. REDUNDANCY PRINCIPLES ($\mathscr{G}_w(V,E), \mathscr{G}^r_w(V,E'), DS, N_D^{P1}, v_i$)

output (N_D^{P2})
local $candidate, O_nd, DS_{nd}, N_D^{P2} \leftarrow \oslash$;

while ($N_D^{P1} \neq \oslash$)
do
- $candidate \leftarrow$ *Randomly select one candidate* $\in N_D^{P1}$;
- $DS_{nd} \leftarrow$ (CHILDREN[a]$(candidate, \mathscr{G}_w(V,E)) \cap DS$;
- $O_{nd} \leftarrow$ (CHILDREN$(candidate, \mathscr{G}_w(V,E)) \setminus DS_{nd}$;
- **comment: P3.1** (see Section 3.1);
- **if** ($v_i \in DS$) **and** *restriction given in* **P3.1**
  - **then**
    - $DS_{nd} \leftarrow$ (CHILDREN$(\mathscr{G}^r_w(V,E), candidate)) \cap DS$;
    - $O_{nd} \leftarrow$ (CHILDREN$(\mathscr{G}^r_w(V,E), candidate)) \setminus DS_{nd}$;
    - **if** *restriction given in* **P3.1**
      - **then** $\{N_D^{P2} \leftarrow N_D^{P2} \cup candidate$;
  - **else**
    - **comment: P3.2** (see Section 3.1);
    - **if** ($v_i \notin DS$) **and** *restriction given in* **P3.2**
      - **then**
        - $DS_{nd} \leftarrow$ (CHILDREN$(\mathscr{G}^r_w(V,E), candidate)) \cap DS$;
        - $O_{nd} \leftarrow$ (CHILDREN$(\mathscr{G}^r_w(V,E), candidate)) \setminus DS_{nd}$;
        - **if** *restriction given in* **P3.1**
          - **then** $\{N_D^{P2} \leftarrow N_D^{P2} \cup candidate$;
- $N_D^{P1} \leftarrow N_D^{P1} \setminus candidate$;

return (N_D^{P2})

[a] CHILDREN: returns the children of a given node v_i, such that $\forall v_j \in V$, $(v_i, v_j) \in E$.

3.1 Redundancy Principles and Approaches

For the specification of the new **OR1** and **OR2** approaches, three basic redundancy principles have to be defined, which help remodel the control structures in relation to redundant pathways. These principles are described as follows and sketched out in Algorithm 3.1:

P1 The selection of new paths is conditioned by all those edges belonging to those driver nodes $\in DS$ (since **OR1** $\subseteq$ **OR2** – cf. Sect. 2.1) with the highest edge betweeness centrality $E_{BC}(v)$ – i.e., those nodes containing the highest control capacity $l_{i,j}$.

P2 Any relink should be done, taking into account the properties of the underlying network. As the control network is based on power law distributions, the redundancy should be subject to those nodes with the maximum degree in order to comply with the power notion. **P1** and **P2** result in a new set of driver nodes N_D^{P1} representing the set of suitable candidates for the relink, capable of ensuring the greatest control transference in perturbed scenarios.

P3 The selection of driver nodes has to be limited to **OR2** (cf. Sect. 2.1), in which the type of node to be relinked has to be considered (see Fig. 1):

P3.1 If the unobserved node is part of DS, then it is necessary to find a driver node $n_d \in N_D^{P1}$ that does not infringe **OR2**, such that: ($|O_{nd}| \geq 2$ and $|DS_{nd}| \geq 0$) or ($|O_{nd}| = 0$ and $|DS_{nd}| \geq 0$), where O_{nd} denotes the set of observed nodes controlled by an n_d, and DS_{nd} represents the set of driver nodes controlled by an $n_d \in DS$.

P3.2 If the unobserved node is not part of DS, then it is necessary to find a driver node $n_d \in N_D^{P1}$ such that ($|O_{nd}| \geq 1$ and $|DS_{nd}| \geq 0$) or ($|O_{nd}| = 0$ and $|DS_{nd}| = 0$).

The result of **P3** is a new set of driver nodes N_D^{P2}, such that $N_D^{P1} \subseteq N_D^{P2}$. To satisfy these principles and to obtain the maximum CLC (i.e., $H_{i,j}$ in Eq. 3) that $\mathscr{G}_w$ can support at any given moment, a second graph $\mathscr{G}_w^r(V, E')$ of the same size as $\mathscr{G}_w$ is required. $\mathscr{G}_w^r$ comprises all the redundant links from the commissioning phase such that $|E'| \geq |E|$, and through this graph it is possible to map the entire system and compute $H_{i,j}$, whereas $L_{i,j}^{t\geq 0}$ provides information of $\mathscr{G}_w$ at each state $t \geq 0$. The update of $\mathscr{G}_w^r$ will depend on the optimization of the restoration mechanisms, which are described in detail below.

3.2 OR1 and OR2 Based on Redundant Pathways

The reconstruction of **OR1** and **OR2** presupposes considering the four restoration strategies laid out in Sect. 3 and the redundancy principles specified in Sect. 3.1, leading to Algorithms 3.2 and 3.4. Both extend the rudimentary versions defined in [1] so as to include redundant links in E' from the commissioning phase, and protect the most critical control pathways over time. The identification of these routes is done through Algorithm 3.3, which is responsible for extracting the most prominent driver nodes from N_D^{P1} and N_D^{P2}.

Algorithm 3.2. OR1$_{v2}$[a] $(\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E'), STG, Lv$[b]$)$

local $DS, relink \leftarrow \emptyset, N \leftarrow V$;
output $(\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E), DS)$

$DS \leftarrow$ **OR1**$(\mathscr{G}_w(V,E))$; **comment:** Procedure **OR1** included in [1];
while $(N \neq \emptyset)$
do
- *Randomly choose one* $v_i \in N$;
- $\{\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E')\} \leftarrow$ STGs $(STG, \mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E'), DS, v_i, Lv)$;
- **if** $v_i \in DS$
  - **then** $N \leftarrow N \setminus \{v_i\}$;
  - **else**
    - **if** $relink \neq \emptyset$
      - **then** $DS \leftarrow DS \cup v_i$;
      - **else** $N \leftarrow N \setminus \{v_i\}$;

return $(\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E), DS)$

[a] OR1$_{v2}$, a redesigned version from the original OR1 specified in [1].

[b] Lv represents the set of those nodes that leave (by themselves) a determined network.

Algorithm 3.3. STGs $(STG, \mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E'), DS, v_i, relink, Lv)$

output $(\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E'))$
local $fathers, brothers, grandfathers, O_{nd}, DS_{nd}, N_D^{P1}, N_D^{P2}, candidate$;

comment: **P1** and **P2** (see Section 3.1);
if $STG \neq$ **STG4**
then
$fathers \leftarrow ((\text{FATHERS}(\mathscr{G}_w(V,E), v_i) \setminus Lv) \cap DS$
while $(fathers \neq \oslash)$
do
if $STG \neq$ **STG2**
then
if $STG =$ **STG1**
then
$brothers \leftarrow ((\text{CHILDREN}(\mathscr{G}_w(V,E), fathers(i)) \setminus Lv) \cap DS$
$N_D^{P1} \leftarrow N_D^{P1} \cup \text{MAXI EBC}^*$ [a] $(\mathscr{G}_w(V,E), fathers(i), brothers)$;
else
$grandfathers \leftarrow ((\text{FATHER}$ [b] $(\mathscr{G}_w(V,E), fathers(i)) \setminus Lv) \cap DS$
$N_D^{P1} \leftarrow N_D^{P1} \cup \text{MAX EBC}^*(\mathscr{G}_w(V,E), fathers(i), grandfathers)$;
$fathers \leftarrow fathers \setminus fathers(i)$;
if $STG =$ **STG2**
then $N_D^{P1} \leftarrow \text{MAX EBC}^*(\mathscr{G}_w(V,E), fathers, v_i)$;
else $\{N_D^{P1} \leftarrow$ MINIMUM DIAMETER WITH EBC* [c] $(\mathscr{G}_w(V,E), DS)$;
comment: **P3** (see Section 3.1);
$N_D^{P2} \leftarrow$ REDUNDANCY PRINCIPLES$(\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E'), DS, N_D^{P1}, v_i)$;
if $N_D^{P2} \neq \oslash$
then
$candidate \leftarrow$ *Randomly select one candidate* $\in N_D^{P2}$;
$\mathscr{G}_w^r(V,E) \leftarrow$ UPDATE NETW [d] $(\mathscr{G}_w^r(V,E), candidate, v_i)$;
return $(\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E'))$

[a] MAX EBC*: returns N_D with the maximum E_{BC} included in $\mathscr{G}_w(V,E)$ (**P1**) and the maximum dominance (**P2**).

[b] FATHERS: set of fathers nodes f_j that comprises a determined node v_i / $\forall\ f_j\ (f_j, v_i) \in E$.

[c] MINIMUM DIAMETER WITH EBC*: returns N_D with the min. diameter and the max. EBC*.

[d] UPDATE NETW: relinks the candidate to node v_i / (candidate, node) $\in E$.

The second observation rule **OR2** in Algorithm 3.4 has to verify until twice the fulfillment of the dominance. The first round is applied in $\mathscr{G}_w$ and the second one in its extended version $\mathscr{G}_w^r$. In this way, any activation of redundant pathways in $\mathscr{G}_w$ at a state t, will prevent the appearance of one or several n_d of degree $d^+ \geq 2$ adjacent to $d-1$ observed vertices, which could infringe **OR2** (cf. Sect. 2.1). This double exploration is crucial to providing a complete enough control structure at each life state t of the system.

As part of this analysis, we provide a brief study of computational complexity, evaluating the upper bound for the new versions of **OR1** and **OR2** together with their restoration scenarios **STG**x ($x = \{1,2,3,4\}$). For simplicity, we denote $\mid V \mid = n$, $\mid E \mid = e$, $\mid \mathbf{N}_D \mid = nd$, where we assume that $nd \approx n$ in the worst case. Concretely, Algorithms 3.2 and 3.4 are quite dependent on the complexity of the traditional algorithms **OR1** and **OR2**, also analyzed in [11] with an overhead of $O(n^2)$, and on the complexity of Algorithm 3.3 and the type of restoration scenario. For **STG**x ($x = \{1,2,3\}$), Algorithm 3.3 has to explore, for each node $\in V$, the existence of a father, brother or grandfather driver with the highest CLC in $\mathscr{G}_w$ (**P1**) and the highest degree (**P2**); both entailing a cost

of $O(n+e+n+e) = O(e+n) = O(n)$ – the process of verifying **P1** and**P2** is encompassed in a unique function denoted here as EBC*. **STG4** becomes analogous to **STG**x ($x = \{1,2,3\}$) but with the difference that it needs to explore those $n_d \in N_D$ with the minimum diameter. As we apply the BFS method (well-known to be $O(n+e)$) to obtain the minimal N_D with the minimum diameter in $\mathscr{G}_w$, the cost of obtaining N_D^{P1}, considering EBC* in this first stage, is $O(n+e+e+n) = O(n)$.

Algorithm 3.4. OR2$_{v2}$[a] $(\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E'), STG, Lv)$

local DS, N_D
output $(\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E), N_D)$

$\{\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E'), DS\} \leftarrow$ **OR1**$(\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E'), STG, Lv)$;
comment: Procedure **OR2** included in [1] with an overhead of $O(n^2)$ [11];
$N_D \leftarrow$ **OR2**$(\mathscr{G}_w(V,E), DS)$;
comment: In the following, the algorithm considers $\mathscr{G}_w^r(V,E')$ and **OR2**;
$i \leftarrow 1$;
while $i \leq \mid N_D \mid$
do {
  Choose vertex $w \in N_D$ *with degree* $d \geq 2$;
  $N \leftarrow$ CHILDREN$(\mathscr{G}_w^r(V,E), w)$;
  if ($d-1$ *vertices* $\in N$ **and** ($\exists$ *a vertex* $w_1 \in U$ *where* $w_1 \in N$)
  then { $N_D \leftarrow N_D \cup \{w_1\}; U \leftarrow U \setminus \{w_1\}; i \leftarrow 1$;
  else $i \leftarrow i+1$;
return $(\mathscr{G}_w(V,E), \mathscr{G}_w^r(V,E'), N_D)$

[a] OR1$_{v2}$, a redesigned version from the original OR2 specified in [1].

Once N_D^{P1} has been computed, Algorithm 3.1 has to be executed to extract N_D^{P2}. Assuming that $\mid N_D^{P1} \mid \approx nd$ in the worst case, the verification of **OR2** in $\mathscr{G}_w$ and $\mathscr{G}_w^r$ for each descendant driver node in N_D^{P1} becomes $O(n^2)$. Note that the costs implicit in *assignment* and *if* instructions tend to $O(1)$, and the same occurs with the updating of $\mathscr{G}_w$ and $\mathscr{G}_w^r$ since the insertion of new links does not involve an additional cost to Algorithm 3.3. As a result, the cost of computing Algorithm 3.3 becomes $O(n+n^2) = O(n^2)$. With all this information in hand, the cost of computing the new version of **OR1** is of at least $O(n \times n^2) = O(n^3)$ in the commissioning phase; whereas the new version **OR2** implies $O(n^3)$ by computing Algorithm 3.2, $O(kn^2)$ (**OR2** of [1]) and $O(kn^2)$ by processing the second rule in $\mathscr{G}_w^r$) (also stated in [11]), resulting in an overhead of $O(n^3+kn^2+kn^2) = O(n^3)$. Unfortunately, the computational cost of the new dominance versions (**OR1**, **OR2**) is higher than the traditional versions, but this increase is only applicable in the initial phase, when the redundant control is being configured. With respect to spatial complexities, it is worth noting that the spatial cost is heavily dependent on each **STG**x ($x = \{1,2,3,4\}$). In the case of **STG2**, the cost may be similar to the cost required by the traditional **OR1** and **OR2** since the redundancy is exclusively concentrated on the father drivers. In contrast, the spatial cost in **STG**x ($x = \{1,3,4\}$) may significantly rise depending on the selection of external driver nodes (brothers, grandfathers or remote nodes) and its penalty in **OR2** (see Algorithm 3.4).

4 Analysis and Discussion

Let Lv be the set of leaving nodes belonging to **[Lv]** (cf. Sect. 2.1); A_e the set of active links in $\mathscr{G}_w(V,E)$ such that $A_e \subseteq E'$; and F_{nd} the set of father drivers that observe a determined vertex in V. Algorithm 4.1 combines the functional features of the four restoration strategies described in the previous section.

Algorithm 4.1. DYNAMIC RECOVERY $(\mathscr{G}_w(V,E), \mathscr{G}^r_w(V,E'), N_D, Lv, A_e, STG)$

```
local v_i, F_nd, found, candidates, fathers
output (G_w(V,E), G^r_w(V,E'), N_D)

for v_i ← 1 to |V|
  do
  F_nd ← FATHERS(G_w(V,E), v_i) ∩ N_D;
  if (F_nd = ∅) and (v_i ∉ Lv)
     then
        comment: Optimal solution;
        found ← false ;
        fathers ← FATHERS(G^r_w(V,E), v_i) ∩ N_D;
        while fathers ≠ ∅) and not found
           do
              Randomly choose a vertex candidate ∈ fathers;
              if (candidate ∉ A_e)
                 then found ← true ;
        if found
           then
              G^r_w(V,E) ← UPDATE NETW (G^r_w(V,E), candidate, v_i);
              A_e ← A_e ∪ {candidate};
           else
              comment: Sub-optimal sol. - STG4 in Algorithm 3.3;
              N_D^P1 ← MIN. DIAMETER WITH EBC*(G_w(V,E), N_D \ Lv);
              N_D^P2 ← RED. PRINCIPLES(G_w(V,E), G^r_w(V,E'), DS, N_D^P1, v_i);
              if N_D^P2 ≠ ∅
                 then
                    candidate ← Randomly select one n_d ∈ N_D^P2;
                    G^r_w(V,E) ← UPDATE NETW (G^r_w(V,E),
                    candidate, v_i);
                    G_w(V,E) ← NEW EBC[a](G_w(V,E));
                    A_e ← A_e ∪ {candidate}; found ← true ;
              if not found
                 then
                    comment: Non-optimal solution;
                    N_D ← N_D ∪ {v_i};
                 else {G_w(V,E), G^r_w(V,E')} ← STGs (STG, G_w(V,E),
                    G^r_w(V,E'), N_D, v_i, Lv);
return (G_w(V,E), G^r_w(V,E'), N_D)
```

[a] NEW EBC: re-compute Eq. 1 to update the control load capacities retained in $\mathscr{G}_w(V,E)$.

The heuristic (i.e., Algorithm 4.1) is based on three main restoration blocks, categorized according to:

- *Optimal* solution, capable of reestablishing the control by automatically activating an $e_{i,j} \in E'$. As the link activation is practically straightforward, the computational cost in performing this part of the algorithm is $O(n)$.
- *Sub-optimal* solution, with the ability to: (i) dynamically find an $n_d \in N_D$ with the minimum diameter in $\mathscr{G}_w$ and the maximum EBC* that ensures coverage of the unobserved node; and (ii) search a redundant pathway (dependent

on **STG**x ($x = \{1, 2, 3, 4\}$)) that guarantees a secondary way to the unobserved node in the near future. This dynamic search of prominent driver nodes follows the principles **P1**, **P2** and **P3**. If none of these principles are achieved, then Algorithm 4.1 looks at the possibility of offering at least a non-optimal solution. The computational overhead, at this point, becomes important since it not only contemplates the charge required in EBC* ($O(n)$) but also the charge necessary to verify **P3.1** or **P3.2** (Algorithm 3.1, $O(n^2)$), the upgrading of loads in $\mathscr{G}_w(V, E)$ after reparation with a further cost of $O(n^2 log(n))$ [24], and the updating of $\mathscr{G}_w(V, E)$ and A_e. That is, $O(n + n^2 + n^2 log(n)) = O(n^2 log(n))$.

– *Non-optimal* solution, to the contrary, deals with transforming any unobserved node to an observed node by including it as part of the N_D. In this way, the node is able to observe itself and comply with at least the first observation rule, **OR1**. Note that this option is also closely related to **[R3]**, when new nodes need to be joined to the network, or the previous options are not reached properly. In either of these two circumstances, the spatial complexity proportionally grows according to the number of unobserved nodes, tearing up the desirable conditions described in Sect. 2.2.

The correctness proof of the restoration problem is solved when the following requirements are satisfied: (1) the algorithm that restores, ensures controllability without violating the control structural properties (*restoration*); (2) the algorithm is able to properly finish in a finite time (*termination*); and (3) the algorithm is able to terminate and provide control at any moment (*validity*).

For the former requirement, if a node v_i is not observed by an $n_d \in N_D$ in a state t, then the control at that moment is not guaranteed. But if there exists (either at local or at remote) a redundant link in $E' \in \mathscr{G}_w^r(V, E')$ created from the commissioning phase, such that $(n_{d_2}, v_i) \in E'$ and $n_{d_2} \in N_D$, then this link is activated complying with **OR1** and **OR2** via Algorithms 3.2 and 3.4. Otherwise, Algorithm 4.1 finds an n_{d_2} with the minimum diameter and EBC* (i.e., **P1** and **P2**) such that $(n_{d_2}, v_i) \in E'$ and it ensures **OR2** (**P3**) by Algorithm 3.1, further complying with **OR1** by having found a suitable driver node n_{d_2} capable of observing itself and all its neighbors. In the case that it is unable to find an appropriate candidate, Algorithm 4.1 is forced to convert the unobserved v_i to a driver node to obey at least **OR1** such that **OR1** $\subseteq$ **OR2**. This way of modeling the network repair means that the structural controllability is maintained at all times where all the nodes are observed by one or several driver nodes $\in N_D$ or by itself if it is an n_d.

Through induction we show the termination of the algorithm, where we first define the initial and final conditions, and the base cases. The precondition adds that $\mathscr{G}_w$ is threatened by one or several (targeted or random) attacks (cf. Sect. 2.2), probably leaving some nodes in $\mathscr{G}_w$ without observation ($F_{nd} = \oslash$); whereas the post-condition certifies that the network is fully observed ($F_{nd} \neq \oslash$) where **OR1** and **OR2** are fulfilled. As for the base cases:

Case 1: $\forall$ nodes in V, $F_{nd} \neq \oslash$ after perturbation. In this case the loop of Algorithm 4.1 is completely processed where all the nodes are covered by a driver node in N_D.

Case 2: $\forall$ nodes processed in V, $\exists$ one $v_i \in V$ such that $F_{nd} = \oslash$ after perturbation. In these circumstances, three scenarios must be distinguished for v_i:

- Optimal solution: $\exists$ a father $n_d \in fathers$ such that $(n_d, v_i) \in E'$. In this case, the conditions, **P1**, **P2** and **P3** are met from the commissioning phase onward.
- Sub-optimal solution: $\mathscr{G}_w^r(V, E')$ does not cover v_i through an edge in E', so it is necessary to explore the existence of one or several candidates $\{n_{d_1}, n_{d_2}, \ldots, n_{d_n}\}$ with: (i) the minimum diameter and EBC*, and (ii) with the capability to relink v_i complying with **P3**.
  If these candidates exist, then $N_D^{P1} \neq 0$ and Algorithm 3.1 verifies the existence of an $n_d \in N_D^{P1}$ that suffices **P3.1** or **P3.2** depending on v_i. If in addition this n_d exists, then $N_D^{P2} \neq \oslash$ guaranteeing the relink. Otherwise, the algorithm enters the non-optimal solution.
- Non-optimal solution: if there is no suitable redundant link in E' or $N_D^{P2} = \oslash$, then N_D is updated by adding v_i as driver node; i.e.: $N_D \longleftarrow N_D \cup \{v_i\}$.

In the first two cases, the network is updated through a new link and in such a way that $\forall$ nodes in V, $F_{nd} \neq \oslash$, satisfying the post-condition. For the second case, N_D is actualized and **OR1** is finally met where **OR1** $\subseteq$ **OR2**.

Induction: if we assume that we are in step k ($k \geq 1$) of the loop where $\exists$ several nodes $\{v_1, v_2, \ldots, v_n\}$ in V with $F_{nd} = \oslash$, we can observe that for these nodes, three possible cases can arise as stated in Case 2. At the end of Algorithm 3.1 with $k = |V|$, the set $F_{nd} \neq \oslash$ for all the nodes in V, once again satisfies the post-condition. This also states that the latter requirement (the validity) is also satisfied since Algorithm 4.1 finishes and ensures that the two observation rules are provided at all times.

4.1 Experimental Results and Discussion

In order to show the practical validity of Algorithm 4.1 for small (∼100–500 nodes), medium (∼500–1000 nodes) and large (∼1000–1500 nodes) networks, a case study written in Matlab is presented in this section. The experiments have been planned to perturb a random number of nodes ($\delta \leq \frac{|V|}{2}$) belonging to pure power-law distributions. Specifically, our research focuses on the Power-Law Out-Degree (PLOD) [25] with a low connectivity probability of $\alpha = 0.1$ for illustrating realistic scenarios, where we evaluate: (1) the spatial overhead invested in N_D, and (2) the effects caused after δ disturbances such as the cascading effect and the optimization of **STG**x ($x = \{1, 2, 3, 4\}$).

Figure 2 shows the spatial cost invested by the new versions **OR1** (DS) and **OR2**. To understand this, it is necessary to observe the value associated with N_D^{bef} (the state of N_D before repair) with respect to the N_D^{orig} given in [1],

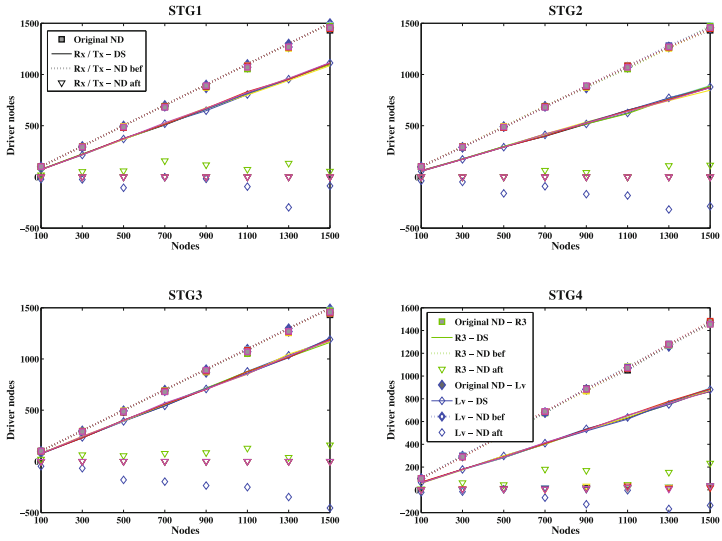

Fig. 2. Spatial complexity before and after perturbation and restoration

as well as the increase of N_D^{aft} after repair. The results indicate that the cardinality of the new N_D^{bef} regarding $|N_D^{orig}|$ is insignificant, regardless of the increase of DS for **STG**x ($x = \{1, 3\}$). Namely, the difference between $|N_D^{aft}|$ and $|N_D^{bef}|$ after repair becomes relevant when the threat is related to [**Lv**] or [**R3**], since the controllability properties are infringed and the network in general needs a new assignation of driver nodes (a concept also supported by the analysis in Sect. 4).

In relation to this research, Fig. 3 illustrates the effect of the threats carried out in the respective recovery scenarios, where we observe that the joining of δ members ([**R3**]), the insertion of δ edges ([**R4**]), and the isolation of the node with the highest degree (the hubs, [**T1**]) and the highest strength ([**T2**]) are the most devastating threats. The effect becomes more notable in those scenarios in which the redundant control is located in the surrounds (**STG**x ($x = \{1, 2, 3\}$)), reaching a fall of 60–80% of the entire network for [**R3**] and [**R4**]. This also means that **STG4** can become more resilient to topological changes. Moreover, these results certainly ratify the findings in [13,21], where it is concluded that power-law networks are in general quite sensitive to threats related to degree sequence.

Figures 4 and 5, in contrast, simplify the simulation results with respect to the optimization of strategies **STG**x ($x = \{1, 2, 3, 4\}$). From these two figures it is possible to appreciate how the system, depending on the degradation of the structural controllability properties after a threat, is able to drive one (non-optimal, suboptimal or optimal) strategy or another. In addition, as the number of attacks can be high in a round ($\delta \leq \frac{|V|}{2}$), the degradation of the structural controllability can drastically change. If the majority of surrounding links are

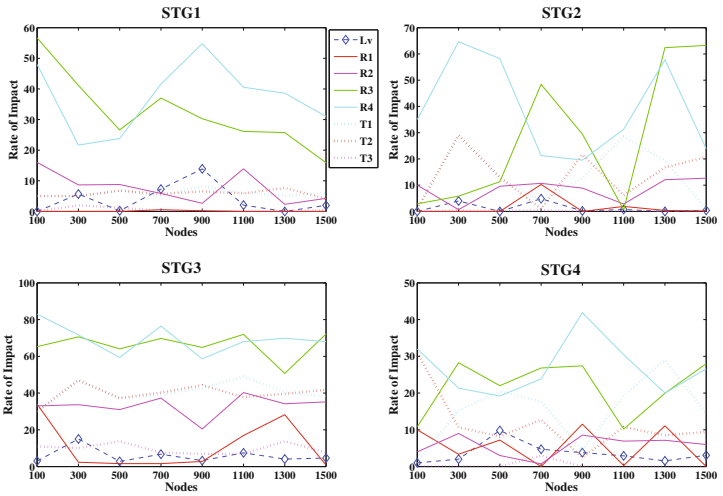

Fig. 3. Cascading effect after perturbation

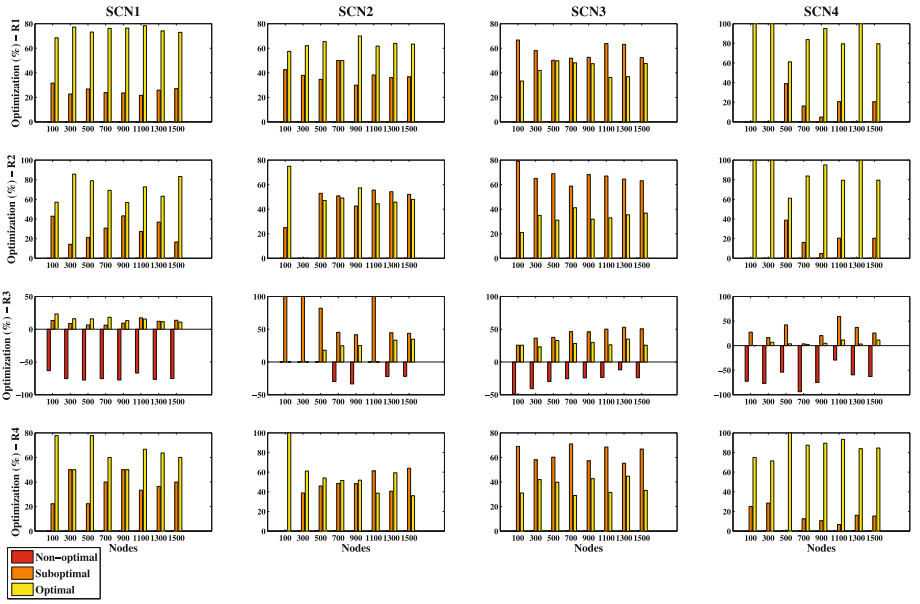

Fig. 4. Optimization of **STG**x ($x = \{1, 2, 3, 4\}$) considering random attacks

lost, the recovery should then depend on the less optimal strategies. But even so, it is also possible to note from the figures that **STG4** followed by **STG1** are the best strategies for self-healing with reduced restoration costs ($O(n)$) for the majority of simulated cases, whereas the worst scenario is **STG3** in

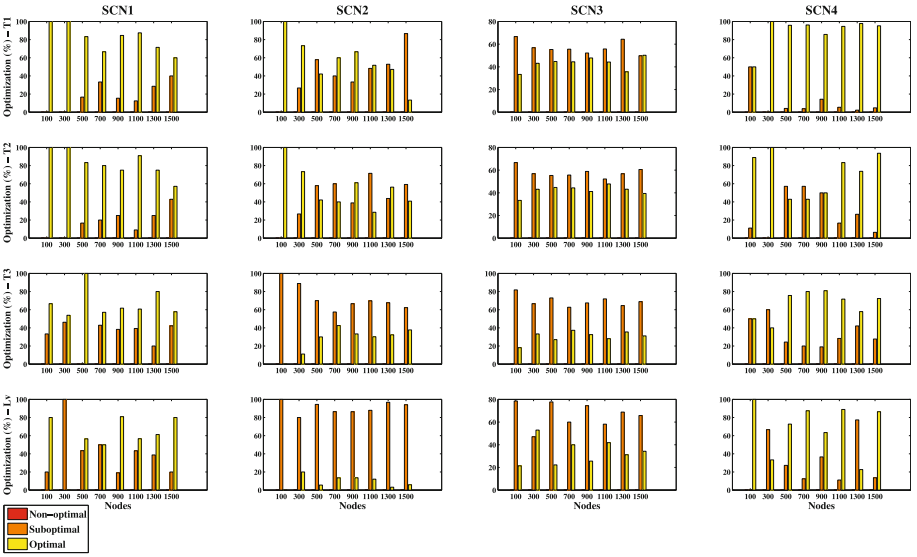

Fig. 5. Optimization of **STG**x ($x = \{1, 2, 3, 4\}$) considering targeted attacks and [**Lv**]

which the rate of optimization is mainly bounded to the sub-optimal solution. However, **STG**x ($x = \{1, 4\}$) is quite susceptible to new integrations where the non-optimal rate reaches more than 50 %, as opposed to the outcome of **STG2**. In these conditions, we determine that critical wireless environments should primarily be subject to relink procedures based on **STG4**. But even so, we also believe that the combined option **STG**x ($x = \{1, 4\}$) would be the best option to guarantee protection at both local and remote level, without discarding the possibility of adapting **STG2** to facilitate the integration of new members within a given network. However, this hypothesis requires evaluating the trade-off between safety and maintenance costs [26,27] when one or several redundancy strategies are established for each node within the network. So this study will be part of our future work.

5 Conclusions and Future Work

Modernized control systems based on CPSs for dynamic automation of operations tend to suffer from (slight or grave) perturbations or frequent changes due to the mobile and sensitive nature of the wireless communications. In this context, the inherent non-locality problem of the control networks is a matter of utmost importance. Automated and reliable self-healing solutions have to be considered as an integral part of network designs. However, most current solutions lack efficient strategies that ensure an acceptable repair cost and responsiveness in time [11], complicating the provision of effective solutions for critical environments. For this reason, four reachability-based restoration strategies have

been presented in this paper, so as to find optimal solutions that guarantee control at all times and without damaging the structural controllability properties. Specifically, this research has entailed the restructuring of the two fundamental dominance rules given in [22] to allow redundancy of control links, either at local or remote level. From these four strategies, we have discovered that the best options are mainly to be found in those distant locations with the highest control load capacity and highest degree, followed by those brother drivers located in the nearest surrounding area. Both strategies offer optimal solutions for the great majority of simulated studies, reaching the expected restoration costs ($O(n)$).

Now, our intention is to broaden the study to find the most suitable redundancy combinations considering the lessons learned here, trying not to lose a suitable balance between installation and maintenance costs, and safety [26,27].

Acknowledgment. The first author receives funding from the *Ramón y Cajal* research programme financed by the Ministerio de Economía y Competitividad. In addition, this work also has been partially supported by PERSIST (TIN2013-41739-R) financed by the same Ministerio.

References

1. Alcaraz, C., Miciolino, E.E., Wolthusen, S.: Structural controllability of networks for non-interactive adversarial vertex removal. In: Luiijf, E., Hartel, P. (eds.) CRITIS 2013. LNCS, vol. 8328, pp. 120–132. Springer, Heidelberg (2013)
2. Alcaraz, C., Zeadally, S.: Critical control system protection in the 21st century: threats and solutions. IEEE Comput. **46**, 74–83 (2013)
3. Pasqualetti, F., Dorfler, F., Bullo, F.: Attack detection and identification in cyber-physical systems. IEEE Trans. Autom. Control **58**(11), 2715–2729 (2013)
4. Sridhar, S., Hahn, A., Govindarasu, M.: Cyber-physical system security for the electric power grid. Proc. IEEE **100**(1), 210–224 (2012)
5. Nie, S., Wang, X., Zhang, H., Li, Q., Wang, B.: Robustness of controllability for networks based on edge-attack. PLoS ONE **9**(2), 1–8 (2014)
6. Alcaraz, C., Lopez, J.: Wide-area situational awareness for critical infrastructure protection. IEEE Comput. **46**(4), 30–37 (2013)
7. Sanjay, B., Sanjeev, S., Ishita, T.: A detailed review of fault-tolerance techniques in distributed system. Int. J. Internet Distrib. Comput. Syst. **1**(1), 33–39 (2012)
8. Treaster, M.: A survey of fault-tolerance and fault-recovery techniques in parallel systems. ACM Computing Research Repository, CoRR 501002, pp. 1–11 (2005)
9. Nakayama, K., Shinomiya, N., Watanabe, H.: An autonomous distributed control method for link failure based on tie-set graph theory. IEEE Trans. Circuits Syst. I Regul. Pap. **59**(11), 2727–2737 (2012)
10. Marchese, M., Mongelli, M.: Simple protocol enhancements of rapid spanning tree protocol over ring topologies. Comput. Netw. **56**(4), 1131–1151 (2012)
11. Alcaraz, C., Wolthusen, S.: Recovery of structural controllability for control systems. In: Butts, J., Shenoi, S. (eds.) Critical Infrastructure Protection. IFIP AICT, vol. 441, pp. 47–63. Springer, Heidelberg (2014)
12. Médard, M., Finn, S.G., Barry, R.A.: Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs. IEEE/ACM Trans. Netw. **7**(5), 641–652 (1999)

13. Quattrociocchi, W., Caldarelli, G., Scala, A.: Self-healing networks: redundancy and structure. PLoS ONE **9**(2), e87986 (2014)
14. Wang, B., Gao, L., Gao, Y., Deng, Y.: Maintain the structural controllability under malicious attacks on directed networks. EPL (Europhys. Lett.) **101**(5), 58003 (2013)
15. Wang, W.-X., Ni, X., Lai, Y.-C., Celso, G.: Optimizing controllability of complex networks by minimum structural perturbations. Phys. Rev. E **85**, 026115 (2012)
16. Ding, J., Lu, Y.-Z., Chu, J.: Recovering the controllability of complex networks. In: 9th World Congress The International Federation of Automatic Control (IFAC), pp. 10894–10901 (2014)
17. Pagani, G.A., Aiello, M.: The power grid as a complex network: a survey. Physica A **392**(11), 2688–2700 (2013)
18. Lin, C.-T.: Structural controllability. IEEE Trans. Autom. Control **19**(3), 201–208 (1974)
19. Haynes, T., Hedetniemi, S.M., Hedetniemi, S.T., Henning, M.A.: Domination in graphs applied to electric power networks. SIAM J. Discrete Math. **15**(4), 519–529 (2002)
20. Kalman, R.E.: Mathematical description of linear dynamical systems. J. Soc. Ind. Appl. Math. Control Ser. A **1**, 152–192 (1963)
21. Liu, Y., Slotine, J.-J., Barabási, A.-L.: Controllability of complex networks. Nature **473**, 167–173 (2011)
22. Kneis, J., Mölle, D., Richter, S.: Parameterized power domination complexity. Inf. Process. Lett. **98**(4), 145–149 (2006)
23. Guo, J., Niedermeier, R., Raible, D.: Improved algorithms and complexity results for power domination in graphs. Algorithmica **52**(2), 177–202 (2008)
24. Robinson, E.: Complex graph algorithms. In: Graph Algorithm in the Language of Linear Algebra, Chap. 6, pp. 59–85. SIAM (2011)
25. Palmer, C., Steffan, J.: Generating network topologies that obey power laws. In: Global Telecommunications Conference, GLOBECOM 2000, vol. 1, pp. 434–438 (2000)
26. Alcaraz, C., Zeadally, S.: Critical infrastructure protection: requirements and challenges for the 21st century. Int. J. Crit. Infrastruct. Protection (IJCIP) **8**, 53–66 (2015)
27. Alcaraz, C., Lopez, J.: Analysis of requirements for critical control systems. Int. J. Crit. Infrastruct. Protection (IJCIP) **5**(137–145), 2012 (2012)