Systems & Control Letters 156 (2021) 105025

Contents lists available at ScienceDirect

Systems & Control Letters

journal homepage: www.elsevier.com/locate/sysconle

Structural controllability and time-to-control of directed scale-free networks with minimum number of driver nodes[☆]

Takanobu Imae, Kai Cai [*]

Department of Electrical and Information Engineering, Osaka City University, Japan

ARTICLE INFO

Article history:
Received 7 April 2021
Received in revised form 26 July 2021
Accepted 13 August 2021
Available online xxxx

Keywords:
Complex networks
Structural controllability
Time-to-control

ABSTRACT

In this paper we study structural controllability and time-to-control of directed scale-free networks, and propose algorithms to guarantee these properties with the minimum number of driver nodes. Structural controllability is a qualitative (i.e. topological) measure of the ability to steer the network to a desired state from an arbitrary initial state; while time-to-control measures how fast the above steering can be done. First, we develop an algorithm that generates directed scale-free networks which are provably structurally controllable with only one driver node. Moreover, considering the tradeoff between control cost (number of driver nodes) and control performance (time-to-control), we propose another algorithm that constructs directed scale-free networks which can be steered to a desired state within a prescribed time bound and with the minimum number of driver nodes.

1. Introduction

Recently, the scale of many real networks has grown larger and their topologies become more complex. In response, many network models [1–5], including small-world networks [2] and *scale-free* networks [3] have been suggested to analyze the topological properties of real complex networks.

Control properties of complex networks have also attracted much attention [6–10]. In particular, much work has been devoted to revealing fundamental aspects of evaluating *controllability* of complex networks, including exact controllability [6], control energy [7,9], network centrality [8], and network aggregation [10]. While these approaches are effective in *quantitatively* characterizing the difficulty of control tasks and designing practical controllers, they require knowledge of concrete values of interconnection parameters of the whole network.

If only topological information of complex networks is available and generic properties are sought for, one may resort to graph-based methods from *structural control theory* [11–15] to analyze network controllability of almost all networks that share a common topology. In [16], the *structural controllability* of both real and theoretic complex networks has been investigated. In order to steer a network to a desired state, input signals need to be injected into certain nodes, which are called *driver nodes*. It is shown in [16] that complex networks in the real world often require many driver nodes to be able to fully control them. This indicates that real complex networks are costly to control.

Since [16], many extensions of structural controllability of complex networks have been investigated, including input/output node selection [17,18], node classification [19], structural target controllability [20], and strong structural controllability [21,22]. A particularly well-studied topic is how to reduce the number of driver nodes (thereby control cost) for structural controllability. Several approaches have been developed by changing the network topology, e.g. adding the minimum number of edges [23,24], rewiring redundant edges [25], and assigning the direction of edges [26,27].

Our first goal in this paper is along this line of reducing the number of driver nodes. In particular, we focus on *directed scale-free networks*, which are more general than the undirected (symmetric) counterparts and have been shown to typically require a large number of driver nodes [16]. Our inquiry is, is it possible to control a directed scale-free network by a *single* driver node?

Our special focus on directed scale-free networks, among other complex networks, is motivated by two main reasons. First, the scale-free property has been identified in a number of real networks, including the WWW, the Internet, the *E. coli* metabolic networks, social networks, and citation networks [5]. Thus scale-free networks represent a wide and important class of complex networks to study their fundamental properties (in this case structural controllability). Second, the observations made in [16] that scale-free networks appear to need large portions of their nodes to be driver nodes and that hubs[1] are typically not

☆ This work was supported in part by JSPS KAKENHI Grant no. 21H04875.

* Corresponding author.
E-mail addresses: imae@c.info.eng.osaka-cu.ac.jp (T. Imae), kai.cai@eng.osaka-cu.ac.jp (K. Cai).

1 Conceptually defined in the literature of complex networks (e.g. [5]), a *hub* is a node connected with a large number of other nodes.

https://doi.org/10.1016/j.sysconle.2021.105025
0167-6911/

driver nodes indicate the extraordinary difficulty to control such networks as compared to other types of complex networks. Hence designing an effective way to reduce the number of driver nodes for scale-free networks is of particular interest and significance.

Beyond mere structural controllability, from the control performance point of view it is important to steer a network to a desired state within an acceptable time bound. However, the studies in [23–27] do not consider reducing the *time-to-control* parameter of networks [28], which measures how long it takes to steer a network to a desired state. In [28,29] both real and theoretic complex networks are investigated, and it is demonstrated that networks which have a handful of driver nodes typically have long time-to-control, while networks have short time-to-control tend to require a large number of driver nodes. This reflects a fundamental tradeoff in classical optimal control (e.g. LQR).

For many real networks, it is desirable to steer them to a desired state with low control cost (measured by number of driver nodes) and within a reasonably short time (measured by time-to-control). Thus our second goal in this paper is to reduce the number of driver nodes for an imposed time-to-control. In particular, we again focus on directed scale-free networks and ask the question: given a time bound T, is it possible to find the *minimum* number of driver nodes by which the network can be steered to a desired state within T?

In this paper we address the above posed questions from the theoretic point of view, and our main contributions are stated as follows. First, we set aside time-to-control and consider only structural controllability. We propose an algorithm that generates a directed network, and prove that the network not only has scale-free property but also is structurally controllable with only one driver node. In particular, the driver node is a hub node; this is in contrast with the observation made in [16]. Second, we consider time-to-control and a specified time bound T. We develop another algorithm that generates a directed scale-free network, and prove that the network needs the minimum number of driver nodes to be steered to a desired state within time bound T. In this case, driver nodes consist of both hubs and non-hubs. The issue of reducing the number of driver nodes to meet the requirement of time-to-control is not addressed in [28]; in this sense our work extends [28]. Summarizing from the above, the contributions of this work are:

- a new algorithm that provably generates a directed scale-free network that is structurally controllable with a single driver node;
- another new algorithm that provably generates a directed scale-free network that is T-structurally controllable with the minimum number of driver nodes.

We note that in the literature, there are many works that propose methods to analyze and/or ensure certain properties of *given* networks (e.g. [16,25,28]), whereas there are also many works that propose models to *generate* networks with certain properties (e.g. [4,30,31]). Our approach is the same as the latter. While results derived from this approach do not directly address given (real) networks, such results can often reveal insights of the network properties of interest and thereby provide indications of dealing with given networks. In our case, the algorithmic mechanisms of generating both scale-free and structurally-controllable properties shed light on plausible strategies to ensure given networks to have these properties (say) by edge addition or rewiring. We leave thorough investigation on this issue to future work, and believe that this paper is an essential step with theoretical results of interest in their own right.

The rest of the paper is organized as follows. In Section 2, we study pure structural controllability. We first introduce the structural controllability theory and scale-free networks, and then present an algorithm for constructing structurally controllable directed scale-free networks with a single driver node. In Section 3, we further consider time-to-control and develop an algorithm that generates directed scale-free networks which can be steered into a desired state within a prescribed time bound by using the minimum number of driver nodes. Finally, our conclusions are stated in Section 4.

2. Structurally controllable scale-free networks with a single driver node

2.1. Preliminaries

2.1.1. Structural controllability

Consider a linear discrete-time dynamic network given by

$$x[t+1] = \boldsymbol{A}x[t] + \boldsymbol{B}u[t] \tag{1}$$

where $x[t] \in \mathbb{R}^N$ represents the state vector of N nodes at time t, $\boldsymbol{A} \in \mathbb{R}^{N\times N}$ is the state matrix, $\boldsymbol{B} \in \mathbb{R}^{N\times M}$ is the input matrix, and $u[t] \in \mathbb{R}^M$ represents the input vector of M signals at time t. The network (1) is *controllable* if the controllability matrix

$$\mathcal{C} = [\boldsymbol{B}\ \boldsymbol{AB}\ \boldsymbol{A}^2\boldsymbol{B}\ \cdots\ \boldsymbol{A}^{N-1}\boldsymbol{B}]$$

satisfies rank $\mathcal{C} = N$. Controllability indicates that one can steer the network to a desired state from an arbitrary initial state by appropriate selecting M input signals.

However, for real complex networks we might have access only to the topological information. In other words, we are often unable to know the precise entries in $\boldsymbol{A}, \boldsymbol{B}$, but knowing only whether each element is nonzero or not. For this, we use the concept of *structural controllability* [11], which can be checked based on the network topology.

Consider a linear time-invariant network described by a pair of structural matrices $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$, where $\bar{\boldsymbol{A}} \in \{0, *\}^{N\times N}$ and $\bar{\boldsymbol{B}} \in \{0, *\}^{N\times M}$. Here, $\{0, *\}^{N\times M}$ is the set of matrices of the size $N \times M$ whose elements are either nonzero $*$ (unknown values) or 0. This network can be represented by a digraph $\mathcal{D}(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}) = (V, E)$, referred to as the *system digraph*, where $V = V_A \cup V_B$ is the node set which includes both the *state* nodes $V_A = \{x_1, \ldots, x_N\}$ and the *input* nodes $V_B = \{u_1, \ldots, u_M\}$, and $E = E_A \cup E_B$ is the edge set which includes both $E_A = \{(x_i, x_j) \mid \bar{\boldsymbol{A}}_{ji} \neq 0\}$ and $E_B = \{(u_m, x_i) \mid \bar{\boldsymbol{B}}_{im} \neq 0\}$. In addition, we define the *state digraph* $\mathcal{D}(\bar{\boldsymbol{A}}) = (V_A, E_A)$.

The network $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ is said to be *structurally controllable* if we can find a pair of real-valued matrices $(\boldsymbol{A}', \boldsymbol{B}')$ which is controllable with the same structural pattern as $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ (i.e. the same zero/non-zero locations). Especially, if such a pair $(\boldsymbol{A}', \boldsymbol{B}')$ exists then almost all possible pairs with the same structural pattern as $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ are controllable.

It is shown in [11] that one can determine whether the network is structurally controllable from the topology of system digraph. A *stem* $S = (V, E)$ (Fig. 1(a)) is an elementary directed path, i.e. $V = \{s_0, s_1, \ldots, s_l\}$ (l is the *length* of S) and $E = \{(s_i, s_{i+1}) \mid 1 \leq i \leq l-1\}$. The initial (resp. terminal) node of a stem is called the *root* (resp. *top*) of the stem. A *bud* $B = (V, E)$ (Fig. 1(b)) is an elementary directed cycle of size n with an additional edge e that begins at an external node v and ends at one node on the cycle, i.e. $V = \{v, c_1, \ldots, c_n\}$ and $E = \{(v, c_1), (c_1, c_2), \ldots, (c_{n-1}, c_n), (c_n, c_1)\}$. This additional edge e is called the *distinguished edge*. A *cactus* (Fig. 1(c)) is a subgraph defined as follows. Given a stem S_0 and buds $B_1, B_2, \ldots, B_l$, the union graph $S_0 \cup B_1 \cup B_2 \cup \cdots \cup B_l$ is a cactus if for every i ($1 \leq i \leq l$) the starting node of the distinguished edge of B_i is (i) the only node of B_i that also belongs to the node set of $S_0 \cup B_1 \cup B_2 \cup \cdots \cup B_{i-1}$; and (ii) also the starting node of a directed edge in the edge set of $S_0 \cup B_1 \cup B_2 \cup \cdots \cup B_{i-1}$. With the notion of

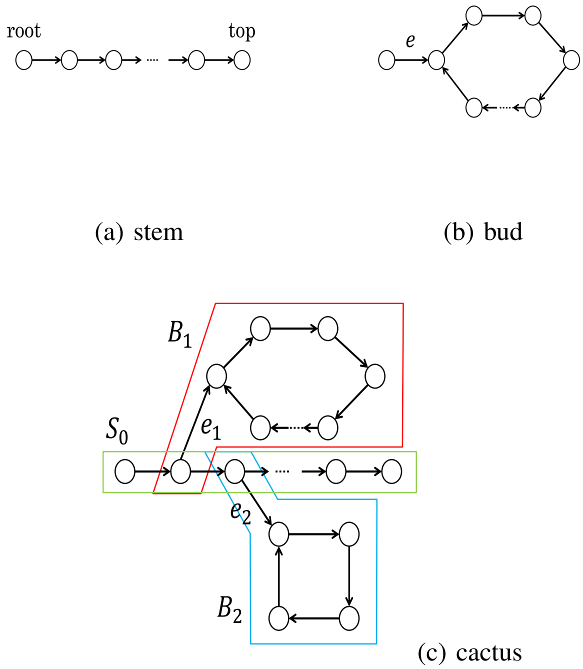

Fig. 1. Classification of specific subgraphs. Nodes and edges are denoted by circles and arrows, respectively.

cactus, the following lemma allows us to determine whether the network is structurally controllable from the topology of system digraph.

Lemma 1 ([11]). *A network $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ is structurally controllable if and only if the system digraph $\mathcal{D}(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ is spanned by cacti, i.e. there exists a set of disjoint cacti rooted at input nodes (one cactus for each input node) that contains all state nodes.*

Remark 1. It follows from Lemma 1 that if a network is structurally controllable, the number of driver nodes (where control inputs are injected) is equal to the number of cacti that span the network. In the special case where the network is spanned by a single cactus, then exactly one driver node is sufficient to make the network structurally controllable. Moreover, this single driver node is the root of the cactus (equivalently the root of the stem S_0 of the cactus).

2.1.2. Scale-free networks

Next, we introduce the *scale-free* property, which is found to be a common feature in many real networks [3]. This property means roughly that many nodes are connected with only a handful of other nodes, while some (hub) nodes with a large number of nodes. Let k_{in} (resp. k_{out}) be the in-degree (resp. out-degree) of a node, namely the number of in-edges (resp. out-edges) of that node. Also let $P(k_{in})$, $P(k_{out})$ be the in-degree distribution and the out-degree distribution, respectively; these are the ratios of the number of nodes with in-degree k_{in} or out-degree k_{out} with respect to the total number of nodes in the network. The scale-free property of directed networks refers to that $P(k_{in})$, $P(k_{out})$ follow the power laws [32]:

$$P(k_{in}) \sim k_{in}{}^{-\gamma_{in}}, \qquad P(k_{out}) \sim k_{out}{}^{-\gamma_{out}}$$

where $\sim$ means "proportional to" and γ_{in}, γ_{out} are called the exponents of the in-degree distribution and the out-degree distribution, respectively. As an example, the in/out-degree distributions of WWW follow power laws with $\gamma_{in} \simeq 2.1$, $\gamma_{out} \simeq 2.7$ [32].

Barabasi and Albert introduced an algorithm to generate *undirected* scale-free networks [3]. This algorithm has two essential ingredients: "growth" and "preferential attachment". First, the network grows by adding one new node at each iteration. Second, the probability that the new node is connected to an existing node is proportional to the latter's degree. It was shown [3] that the degree distribution of undirected scale-free networks generated by their algorithm follows a power law. Extending this model, an algorithm for constructing *directed* scale-free networks has been proposed in [33]; it is shown that both the in-degree and out-degree distributions of the generated networks follow power laws.

Although the algorithm in [33] can generate directed scale-free networks, it is shown in [16] that for such networks a large number of driver nodes are typically needed for ensuring structural controllability. Thus networks generated in [33] may be too costly to be fully controlled. To reduce control cost, we will design an algorithm to generate directed scale-free networks that require just one driver node.

2.2. Algorithm and main result

First, we present the algorithm to generate structurally controllable directed scale-free networks of N (state) nodes, where N is typically a large positive number. The design is an extension of that in [33].

Algorithm 1 (*Constructing Structurally Controllable Directed Scale-free Networks with a Single Driver Node*)**.**

(1) Initially let $\mathcal{D}_0$ be a directed graph with $m_0(> 1)$ nodes that is spanned by a stem. Also number the nodes in $\mathcal{D}_0$ from 1 to m_0 from the root to the top of the stem.
(2) At each iteration h $(m_0 + 1 \le h \le N)$, add one new node (numbered h) and establish for node h one in-edge from the node $h - 1$. Moreover, establish for node h in-edges from $m_{in} \in [1, m_0]$ existing nodes and out-edges to $m_{out} \in [1, m_0]$ existing nodes. Here, the probability $\Pi_{i,in}$ (resp. $\Pi_{i,out}$) that an existing node $i \in [1, h-1]$ with in-degree $k_{i,in}$ (resp. out-degree $k_{i,out}$) obtains an in-edge from (resp. out-edge to) the new node is

$$\Pi_{i,in} = \frac{k_{i,in}}{\sum_{j=1}^{h-1} k_{j,in}}, \tag{2}$$

$$\text{resp.} \quad \Pi_{i,out} = \frac{k_{i,out}}{\sum_{j=1}^{h-1} k_{j,out}}. \tag{3}$$

No multiple edges are allowed.
(3) Advance h to $h + 1$. If $h \le N$ then go to Step 2). Otherwise stop.

In the above algorithm, we initialize the directed graph $\mathcal{D}_0$ such that it is spanned by a stem, which contains all the m_0 nodes. Moreover, at each iteration we always add an in-edge to the new node from the existing node which is added to the network in the previous iteration. In this way, the generated network contains a stem from the node 1, the root of stem in the initial directed graph $\mathcal{D}_0$, all the way through to the final node N; namely, spanned by a stem. Since a stem is a special case of a cactus, the resulting network is spanned by a cactus, and thus is structurally controllable if we inject a control input to the root node. That is, the root node is the only driver node. Moreover, it can be shown that the network generated from Algorithm 1 has scale-free properties of both in-degree and out-degree distributions. Summarizing, we present the following main result.

Theorem 1. *The network generated by Algorithm 1 is structurally controllable with a single driver node and has scale-free properties as follows:*

$$P(k_{in}) \sim k_{in}^{-\gamma_{in}}, \quad P(k_{out}) \sim k_{out}^{-\gamma_{out}}$$

where $\gamma_{in} = 2 + \frac{m_{in}+1}{m_{out}}$, $\gamma_{out} = 2 + \frac{m_{out}+1}{m_{in}}$. *Moreover, the single driver node is the root of the stem in the initial directed graph* $\mathcal{D}_0$.

We postpone the proof of Theorem 1 to the next subsection. Here we provide several remarks and an illustrating example.

Theorem 1 asserts that theoretically, one may effectively generate directed networks that are both scale-free and structurally controllable with a single driver node. This driver node is the first node and thus a hub (inasmuch as it is one of the oldest nodes in the network and has the longest time to establish new edges). These conclusions are in contrast with observations of real scale-free networks, as well as previously studied theoretic scale-free networks [16]. While networks generated by Algorithm 1 are theoretic ones and thus different from real networks, this algorithm design suggests a strategy if one wishes to fully control a real scale-free network by a single driver node: i.e. establish a stem throughout by adding new edges or rewiring existing edges.

The power exponents γ_{in} and γ_{out} in Theorem 1 are determined by network parameters m_{in} and m_{out}. However, as compared to [33], both the in-degree power exponent γ_{in} and the out-degree power exponent γ_{out} are different (in [33] they are respectively $2 + \frac{m_{in}}{m_{out}}$ and $2 + \frac{m_{out}}{m_{in}}$). Specifically, if the parameters m_{in}, m_{out} are such that $m_{in} = m_{out}$, then γ_{in} and γ_{out} still depend on the values of m_{in}, m_{out} in Theorem 1, while they are always equal to 3 in [33].

In Fig. 2, we provide an example to illustrate the process of Algorithm 1. Let $m_0 = 4$, $m_{in} = m_{out} = 2$. First, the initial directed graph $\mathcal{D}_0$ is a cycle; thus we can find a stem spanning $\mathcal{D}_0$. Then we number each node as shown in Fig. 2(a). Second, we add the new node 5 and establish an in-edge (colored in red) from the node 4. After that, we calculate the probabilities $\Pi_{i,in}$ (resp. $\Pi_{i,out}$) for each existing node by (2) (resp. (3)). As a result, we obtain $\Pi_{1,in} = \Pi_{2,in} = \Pi_{3,in} = \Pi_{4,in} = \frac{1}{4}$ and $\Pi_{1,out} = \Pi_{2,out} = \Pi_{3,out} = \frac{1}{5}$, $\Pi_{4,out} = \frac{2}{5}$. According to these probabilities, we select m_{in} $(= 2)$ nodes which give an out-edge to and m_{out} $(= 2)$ nodes which is given an in-edge from the node 5, respectively, as shown in Fig. 2(b). At the next iteration, a new node 6 is added and gets an in-edge from the node 5. Calculating each probability, we obtain $\Pi_{1,in} = \Pi_{2,in} = \frac{1}{4}$, $\Pi_{3,in} = \Pi_{4,in} = \frac{1}{8}$, $\Pi_{1,in} = \frac{1}{4}$ and $\Pi_{1,out} = \Pi_{2,out} = \frac{1}{9}$, $\Pi_{3,out} = \Pi_{4,out} = \frac{2}{9}$, $\Pi_{5,out} = \frac{1}{3}$, and add new edges as shown in Fig. 2(c). After adding new edges, it is clear that there is a stem with node set $\{1, 2, 3, 4, 5, 6\}$. We continue these iterations until the network size becomes N. In Fig. 2(d), we illustrate an example of the generated network of size $N = 100$. Note that there is a long stem which starts from node 1 (colored in red) and ends at node 100; the stem is emphasized by red edges. In other words, this network can be controlled by a single input signal injected into the root of the stem.

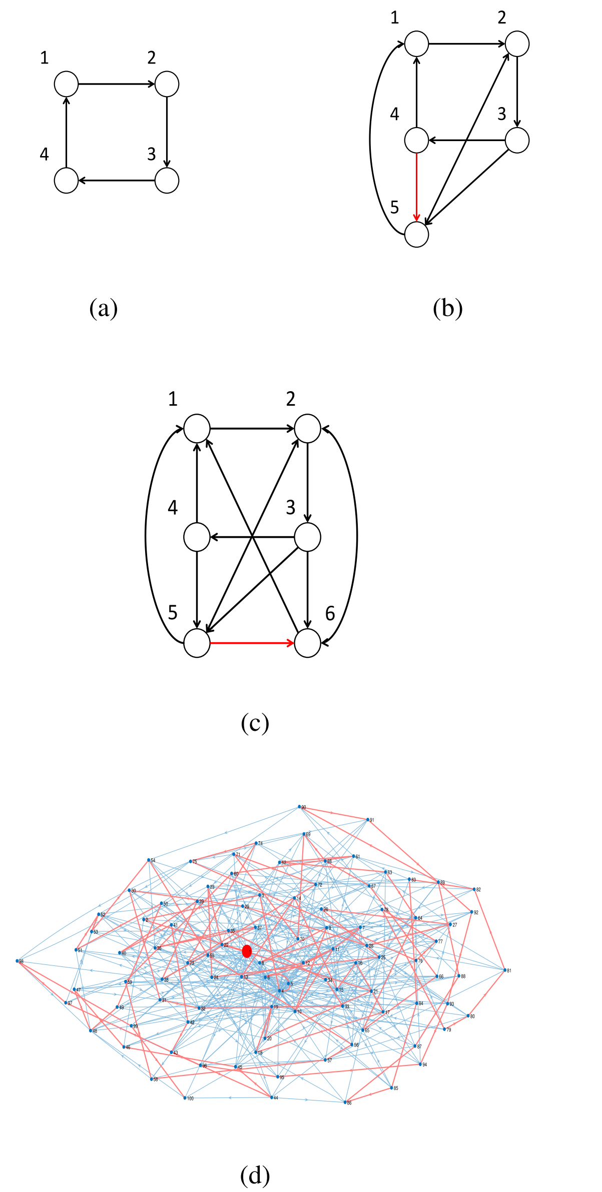

Fig. 2. Example illustrating the process of generating a directed scale-free network by the Algorithm 1. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2.3. Proof of Theorem 1

We provide the proof of Theorem 1. First, we show that Algorithm 1 generates a directed scale-free network. In Step (1), there are initially m_0 nodes in the network, and in Step (2) a new node is added to the network at each iteration. Let node i be the node that is newly added at iteration h_i, and we focus on how its in-degree and out-degree change with respect to $h (\geq h_i)$. Let $k_{i,in}(h)$ and $k_{i,out}(h)$ be in-degree and out-degree of node i at iteration $h (\geq h_i)$, respectively. In Step (2) of Algorithm 1 and iteration h_i, for the newly added node i, in addition to establishing in-edges from m_{in} distinct nodes, we also add an in-edge from the node $h_i - 1$. Thus node i has $m_{in} + 1$ in-edges and m_{out} out-edges, i.e.

$$k_{i,in}(h_i) = m_{in} + 1, \qquad (4)$$
$$k_{i,out}(h_i) = m_{out}.$$

At each iteration $h > h_i$, a new node is added to the network; in addition to the in-edge from the node added at the previous iteration, the new node establishes an in-edge from m_{in} distinct nodes and an out-edge to m_{out} distinct nodes. The probability $\Pi_{i,in}$ that each new node establishes an in-edge from node i and the probability $\Pi_{i,out}$ that each new node establishes an out-edge to node i are expressed in (2) and (3), respectively. Hence, at each step the expectation of the increase of the in-degree (resp. out-degree) of node i is $m_{out}\Pi_{i,in}$ (resp. $m_{in}\Pi_{i,out}$). When h is large, i.e. $h - h_i \gg 0$, one may regard h as a continuous variable. With this approximation, the temporal variations of $k_{i,in}$ and $k_{i,out}$ are represented as

$$\frac{dk_{i,in}}{dh} = m_{out}\Pi_{i,in} = \frac{m_{out}k_{i,in}}{\sum_j k_{j,in}}, \quad (5)$$

$$\frac{dk_{i,out}}{dh} = m_{in}\Pi_{i,out} = \frac{m_{in}k_{i,out}}{\sum_j k_{j,out}}. \quad (6)$$

In the following we focus on the derivation of the in-degree distribution based on (5); the out-degree distribution based on (6) is analogous.

The denominator of the right side of (5) stands for the summation of the in-degrees of all nodes in the network at iteration h. Since at each iteration, the network is added with $m_{in}+m_{out}+1$ directed edges, we have $\sum_j k_{j,in} = K_0 + (m_{in} + m_{out} + 1)h$, where K_0 is the number of edges in the initial directed graph $\mathcal{D}_0$. For $h \gg h_i$, the constant K_0 can be ignored and we obtain $\sum_j k_{j,in} = (m_{in} + m_{out} + 1)h$. Hence the probability $\Pi_{i,in} = \frac{k_{i,in}}{(m_{in}+m_{out}+1)h}$ and (5) becomes

$$\frac{dk_{i,in}}{dh} = \frac{m_{out}k_{i,in}}{(m_{in} + m_{out} + 1)h}. \quad (7)$$

By solving this differential equation, we have $k_{i,in}(h) = Ah^{\frac{m_{out}}{m_{in}+m_{out}+1}}$, where A denotes an integration constant. Using the initial condition (4), we obtain

$$A = \frac{m_{in} + 1}{h_i^{\frac{m_{out}}{m_{in}+m_{out}+1}}}.$$

Hence the solution of (7) is

$$k_{i,in}(h) = (m_{in} + 1)\left(\frac{h}{h_i}\right)^{\frac{m_{out}}{m_{in}+m_{out}+1}}.$$

By fixing $k_{i,in}(h) := k_{in}$ and replacing h_i by $h_{k_{in}}$, we have

$$h_{k_{in}} = \left(\frac{m_{in} + 1}{k_{in}}\right)^{1+\frac{m_{in}+1}{m_{out}}} h.$$

This equation represents the iteration when the node with in-degree k_{in} at h is added to the network. Let $N_{<k_{in}}$ be the number of nodes whose in-degrees are lower than k_{in} at iteration h; then this number is equal to the number of nodes which are added after the iteration $h_{k_{in}}$ and is represented as

$$N_{<k_{in}} = h - \left(\frac{m_{in} + 1}{k_{in}}\right)^{1+\frac{m_{in}+1}{m_{out}}} h.$$

On the other hand, let $P(k'_{in})$ be the in-degree distribution; then $P(k'_{in})$ represents the ratio of nodes with in-degree k_{in}. Thus $N_{<k_{in}}$ also has the form $N_{<k_{in}} = N(h)\int_{m_{in}+1}^{k_{in}} P(k'_{in})dk'_{in}$, where $N(h)$ denotes the number of nodes. If $h \gg m_0$, we can ignore m_0, so we have $N(h) = m_0 + h \approx h$. Thus we obtain

$$h\int_{m_{in}+1}^{k_{in}} P(k'_{in})dk'_{in} = h - \left(\frac{m_{in} + 1}{k_{in}}\right)^{1+\frac{m_{in}+1}{m_{out}}} h.$$

Dividing both sides by h, we have

$$\int_{m_{in}+1}^{k_{in}} P(k'_{in})dk'_{in} = 1 - \left(\frac{m_{in} + 1}{k_{in}}\right)^{1+\frac{m_{in}+1}{m_{out}}}. \quad (8)$$

Thus we can obtain the power distribution by differentiating (8) with respect to k_{in}:

$$P(k_{in}) \sim k_{in}^{-\left(2+\frac{m_{in}+1}{m_{out}}\right)} \sim k_{in}^{-\gamma_{in}}$$

where the exponent $\gamma_{in} = 2 + \frac{m_{in}+1}{m_{out}}$. By a similar derivation, the out-degree distribution follows the power law:

$$P(k_{out}) \sim k_{out}^{-\left(2+\frac{m_{out}+1}{m_{in}}\right)} \sim k_{out}^{-\gamma_{out}}$$

where the exponent $\gamma_{out} = 2 + \frac{m_{out}+1}{m_{in}}$. Therefore, it follows that the generated networks have the scale-free property.

It is left to show that the generated network is structurally controllable with a single driver node, which is the root of the stem in the initial directed graph $\mathcal{D}_0$. By the setup of Step (1) in Algorithm 1, the initial network $\mathcal{D}_0$ is spanned by a stem. We assume that the network $\mathcal{D}_h$ is spanned by a stem at iteration h ($\geq$ 1). At iteration $h + 1$, a new node establishes an in-edge from the existing node added at iteration h. Thus $\mathcal{D}_{h+1}$ is also spanned by a stem. By induction we conclude that the generated network is spanned by a single long stem, whose root is the node 1. Since a stem is a special cactus, it follows from Lemma 1 that the network is structurally controllable. Moreover, by Remark 1 we conclude that a single driver node is sufficient to make the network structurally controllable, and this driver node is always the root of a stem in the initial directed graph $\mathcal{D}_0$. □

3. *T*-structurally controllable scale-free networks with minimum driver nodes

So far we have considered pure structural controllability. In this section, we further consider control performance in the sense of steering networks to a desired state within a specified time bound. For this, we introduce the concept of time-to-control [28], and then extend Algorithm 1 to generate structurally controllable directed scale-free networks which can be steered to a desired state with the *minimum* number of driver nodes and within a prescribed time bound.

3.1. Time-to-control

Consider again the linear discrete-time dynamic network (1). Given $T \in [1, N]$, the *partial controllability matrix* is:

$$\mathcal{C}(\boldsymbol{A}, \boldsymbol{B}; T) = [\boldsymbol{B}, \boldsymbol{AB}, \boldsymbol{A}^2\boldsymbol{B}, \ldots, \boldsymbol{A}^{T-1}\boldsymbol{B}].$$

Define *controllability index* as the minimum value of T such that $\mathcal{C}(\boldsymbol{A}, \boldsymbol{B}; T)$ is of full rank, i.e.

$$\tau(\boldsymbol{A}, \boldsymbol{B}) = \min\{T \in [1, N] : \operatorname{rank}(\mathcal{C}(\boldsymbol{A}, \boldsymbol{B}; T)) = N\}.$$

As with structural controllability, however, one may only have access to the topological information of real networks. For this reason, the *structural controllability index* is introduced to investigate the time-to-control of the network without knowing the precise entries of $\boldsymbol{A}, \boldsymbol{B}$ [28]. Given a pair of structural matrices $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$, the network $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ is *structurally controllable with index* T if there exists a pair of real matrices $(\boldsymbol{A}, \boldsymbol{B})$, with the same structural pattern as $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$, such that the controllability index of $(\boldsymbol{A}, \boldsymbol{B})$ is equal to T. This means that we can find a network with a system digraph associated with $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ such that the network can be steered to a desired state within T time steps. Then the minimum value of T over all possible pairs of real matrices $(\boldsymbol{A}, \boldsymbol{B})$ is called the *structural controllability index*, which is denoted by $\bar{\tau}(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}}) := \min_{(\boldsymbol{A},\boldsymbol{B})\in\{0,*\}} \tau(\boldsymbol{A}, \boldsymbol{B})$. We say that the network $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ is *T-structurally controllable* if the structural controllability index of $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ is equal to T. Similar to the structural controllability, it is shown in [28] that we may determine the T-structural controllability by a graph-theoretical approach.

Lemma 2 (*[28]*)**.** *Consider a network $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ with N state nodes and let $T \in [1, N]$. The network is T-structurally controllable if and only if the system digraph $\mathcal{D}(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ associated with $(\bar{\boldsymbol{A}}, \bar{\boldsymbol{B}})$ is spanned by cacti, i.e. there exists a set of disjoint cacti that contains all N nodes, where every cactus contains at most T nodes.*

Although directed scale-free networks generated by Algorithm 1 needs only one driver node, from the viewpoint of time-to-control, it takes a long time to steer the generated networks to a desired state. Indeed, since the root of the stem, which starts from node 1 and ends at node N, is the only driver node, it follows from Lemma 2 that the networks generated by Algorithm 1 are N-structurally controllable. In many real situations, it is desirable not only to fully control networks with as few driver nodes as possible, but also to steer networks to a desired state within as short time as possible. In the next subsection, we will extend Algorithm 1 to generate directed scale-free networks which can be steered into a desired state within prescribed time bounds.

3.2. Algorithm and main result

First, we present the algorithm. Let T be the time bound specified *a priori*.

Algorithm 2 (*Constructing T-structurally Controllable Directed Scale-free Networks with Minimum Driver Nodes*).

(1) Initially let $\mathcal{D}_0$ be a directed graph with $m_0 (1 < m_0 < T)$ nodes that is spanned by a stem. Also number the nodes in $\mathcal{D}_0$ from 1 to m_0 from the root to the top of the stem.
(2) At each iteration h ($m_0 + 1 \le h \le N$), add one new node (numbered h). If $h \ne jT + 1$ (where j is an integer such that $1 \le j \le \lceil \frac{N}{T} \rceil - 1$), then establish for node h one in-edge from node $h - 1$. Moreover, establish for node h in-edges from $m_{in} \in [1, m_0]$ existing nodes and out-edges to $m_{out} \in [1, m_0]$ existing nodes. Here, the probability $\Pi_{i,in}$ (resp. $\Pi_{i,out}$) that an existing node $i \in [1, h-1]$ with in-degree $k_{i,in}$ (resp. out-degree $k_{i,out}$) obtains an in-edge from (resp. out-edge to) the new node h is (the same as (2), (3))
$$\Pi_{i,in} = \frac{k_{i,in}}{\sum_{j=1}^{h-1} k_{j,in}}, \tag{9}$$
$$\text{resp.} \quad \Pi_{i,out} = \frac{k_{i,out}}{\sum_{j=1}^{h-1} k_{j,out}}. \tag{10}$$
No multiple edges are allowed.
(3) Advance h to $h + 1$. If $h \le N$ then go to Step 2). Otherwise stop.

In Fig. 3, we provide an example to illustrate Algorithm 2. Let $N = 100, m_0 = 4, m_{in} = m_{out} = 2$, and set the time bound to be $T = 20$. As shown in Fig. 3(a), each newly added node h ($5 \le h \le 20$) acquires an in-edge from the existing node $h - 1$. Thus there is a stem (red path) from node 1 to node 20, and node 1 colored in red is the driver node. When node 21 is added, which will become the next driver node (Fig. 3(b)), node 20 need not provide an edge to node 21 (because $21 = T + 1$). Finally in Fig. 3(c), the resulting network is illustrated. There are five (20-node) stems whose roots are nodes 1, 21, 41, 61, and 81 (colored in red). Therefore, these five nodes are the driver nodes to ensure that the network is T-structurally controllable.

In the network generated by Algorithm 2, there are $\lceil \frac{N}{T} \rceil$ stems; the jth stem starts from node $jT + 1$ and ends at node $(j + 1)T$ $(0 \le j \le \lceil \frac{N}{T} \rceil - 1)$. Thus if we select nodes $jT + 1 (0 \le j \le \lceil \frac{N}{T} \rceil - 1)$ as driver nodes, the input signals can reach all the nodes in each stem within the given time bound T. This means that the network is spanned by stems of length T. Therefore, by Lemma 2 the network is T-structurally controllable.

Note that the algorithm proposed in [33] and Algorithm 1 in Section 2 are in fact two special cases of Algorithm 2. When $T = 1$, the edge from the node $h - 1$ to node h need not be added (for all h). This reduces to the algorithm in [33]. On the other hand, if $T = N$, every newly added node is given an in-edge from the node added at the previous iteration. This means that in the generated networks there is a long stem which starts from node 1 and ends at node N. Therefore, this is the same as Algorithm 1.

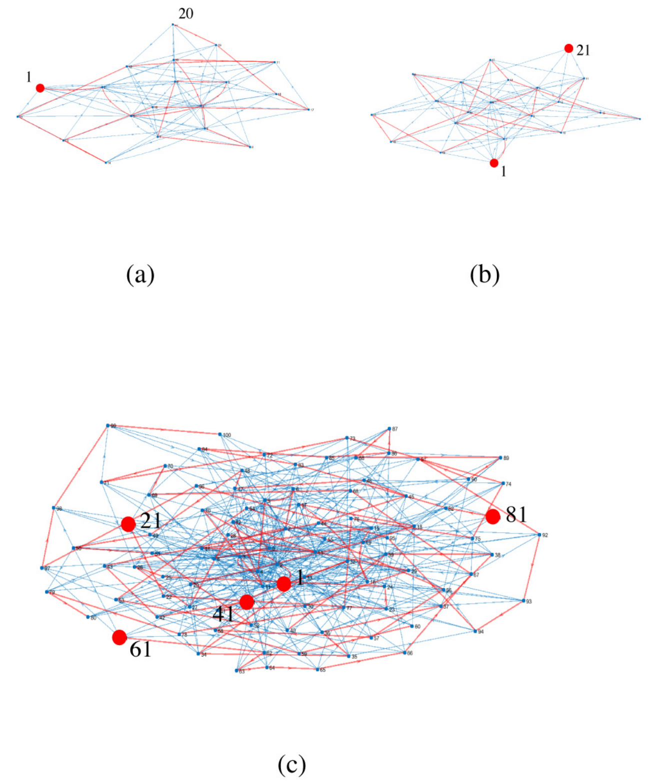

Fig. 3. Example illustrating the process of generating a directed scale-free network by the Algorithm 2. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The following is the main result of this section.

Theorem 2. *Given a time bound T $(1 \le T \le N)$, the network generated by Algorithm 2 has scale-free properties as follows:*

$P(k_{in}) \sim k_{in}^{-\gamma_{in}}, \quad P(k_{out}) \sim k_{out}^{-\gamma_{out}}$

where $\gamma_{in} = 2 + \frac{m_{in}+1-\frac{1}{T}}{m_{out}}$, $\gamma_{out} = 2 + \frac{m_{out}+1-\frac{1}{T}}{m_{in}}$. Moreover, the generated network is T-structurally controllable with the minimum $\lceil \frac{N}{T} \rceil$ driver nodes.

The proof of Theorem 2 follows similar derivations to those in the proof of Theorem 1. In particular, from the probability point of view, for a time bound T $(1 \le T \le N)$ a new edge is established (resp. is not established) from node $h - 1$ to node h $(h \ge m_0 + 1)$ with probability $1 - \frac{1}{T}$ (resp. $\frac{1}{T}$). Thus the expectation of the in-degree of a newly added node, as in (4) in the proof of Theorem 1, is $(m_{in}+1)\cdot(1-\frac{1}{T})+m_{in}\cdot\frac{1}{T} = m_{in}+1-\frac{1}{T}$. Therefore, carrying out the same calculations as those on p. 5, the power exponents γ_{in} and γ_{out} of the network generated by Algorithm 2 are obtained.

Theorem 2 asserts that for an arbitrary time bound T $(1 \le T \le N)$, Algorithm 2 generates directed scale-free networks which are T-structurally controllable with the minimum $\lceil \frac{N}{T} \rceil$ driver nodes. These drive nodes are numbered $1, T + 1, \ldots, (\lceil \frac{N}{T} \rceil - 1)T + 1$. The first few nodes are hubs (older ones in the network) while the last few are non-hubs (newer ones); thus the set of driver nodes consists of both hubs and non-hubs.

In addition, note that while the power exponents $\gamma_{in}, \gamma_{out}$ of networks generated from Algorithm 1 are determined only by parameters m_{in}, m_{out}, Algorithm 2 generates networks in which the time bound T also influences the power exponents. Finally, while networks generated by Algorithm 2 are theoretic ones and thus different from real networks, this algorithm design suggests a strategy if one wishes to control a real scale-free network to meet a given time bound using the minimum number of driver nodes: i.e. establish $\lceil \frac{N}{T} \rceil$ stems by adding new edges or rewiring existing edges.

4. Conclusions and future work

In this paper, we have proposed an algorithm to generate directed scale-free networks which need only one driver node to ensure structural controllability. Moreover, we have developed another algorithm that constructs directed scale-free networks which can be steered to a desired state within a required time bound as well as with the minimum number of deriver nodes.

The designs of these algorithms have suggested plausible strategies of ensuring scale-free, structurally-controllable, and time-to-control properties. In future work, we aim to investigate the problem of, for given real networks (whether scale-free or not), how to reduce the number of driver nodes by suitable topological changes (as in [23–27]) while meeting desired time constraints. Moreover, we also aim to extend these ideas to the case where the quantitative network information is available (as in [6–10]), and therein study the relations/tradeoffs between controllability with time-to-control constraint and network centrality and/or control energy.

CRediT authorship contribution statement

Takanobu Imae: Method development, Proof, Initial Writing, Programming. **Kai Cai:** Idea development, Literature review, Writing quality control.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] P. Erdos, On random graphs, Publ. Math. 6 (1959) 290–297.
[2] D. Watts, S. Strogatz, Collective dynamics of small-world networks, Nature 393 (1998) 440–442.
[3] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[4] B. Bollobás, C. Borgs, J. Chayes, O. Riordan, Directed scale-free graphs, in: Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, 2003, pp. 132–139.
[5] A.-L. Barabási, Network Science, Cambridge University Press, 2016.
[6] Z. Yuan, C. Zhao, Z. Di, W.X. Wang, Y.C. Lai, Exact controllability of complex networks, Nature Commun. 4 (2013) 2447.
[7] F. Pasqualetti, S. Zampieri, F. Bullo, Controllability metrics, limitations and algorithms for complex networks, IEEE Trans. Control Netw. Syst. 1 (2014) 40–52.
[8] N. Bof, G. Baggio, S. Zampieri, On the role of network centrality in the controllability of complex networks, IEEE Trans. Control Netw. Syst. 4 (2017) 643–653.
[9] G. Lindmark, C. Altafini, A driver node selection strategy for minimizing the control energy in complex networks, IFAC-PapersOnLine 50 (2017) 8309—8314.
[10] G. Casadei, C.C. de Wit, S. Zampieri, Scale free controllability of large-scale networks: an output controllability approach, in: Proceedings of the 47th IEEE Conference on Decision and Control, 2019, pp. 1–8.
[11] C.-T. Lin, Structural controllability, IEEE Trans. Automat. Control 19 (3) (1974) 201–208.
[12] K.J. Reinschke, Multivariable Control: A Graph Theoretic Approach, Springer, 1988.
[13] J.M. Dion, C. Commault, J.V. der Woude, Generic properties and control of linear structured systems: a survey, Automatica 39 (2003) 1125—1144.
[14] M.K.S. Faradonbeh, A. Tewari, G. Michailidis, Optimality of fast-matching algorithms for random networks with applications to structural controllability, IEEE Trans. Control Netw. Syst. 4 (2017) 770–780.
[15] T. Menara, D. Bassett, F. Pasqualetti, Structural controllability of symmetric networks, IEEE Trans. Automat. Control 64 (2019) 3740–3747.
[16] Y.-Y. Liu, J.-J. Slotine, A.-L. Barabási, Controllability of complex networks, Nature 473 (7346) (2011) 167–173.
[17] V. Tzoumas, M. Rahimian, G. Pappas, A. Jadbabaie, Minimal actuator placement with bounds on control effort, IEEE Trans. Control Netw. Syst. 3 (2016) 67–78.
[18] S. Pequito, S. Kar, A.P. Aguiar, A framework for structural input/output and control configuration selection of large-scale systems, IEEE Trans. Automat. Control 61 (2016) 303–318.
[19] C. Commault, J. van der Woude, A classification of nodes for structural controllability, IEEE Trans. Automat. Control 64 (2019) 3877–3882.
[20] J. Li, X. Chen, S. Pequito, G.J. Pappas, V.M. Preciado, Structural target controllability of undirected networks, in: Proceedings of the IEEE Conference on Decision and Control, 2018, pp. 6656–6661.
[21] G. Reissig, C. Hartung, F. Svaricek, Strong structural controllability and observability of linear time-varying systems, IEEE Trans. Automat. Control 59 (2014) 3087–3092.
[22] J. Jia, H. van Waarde, H. Trentelman, K. Camlibel, A unifying framework for strong structural controllability, IEEE Trans. Automat. Control 66 (2021) 391–398.
[23] W.-X. Wang, X. Ni, Y.-C. Lai, C. Grebogi, Optimizing controllability of complex networks by minimum structural perturbations, Phys. Rev. E 85 (2012) 026115.
[24] X. Chen, S. Pequito, G.J. Pappas, V.M. Preciado, Minimal edge addition for network controllability, IEEE Trans. Control Netw. Syst. 6 (1) (2019) 312–323.
[25] J. Xu, J. Wang, H. Zhao, S. Jia, Improving controllability of complex networks by rewiring links regularly, in: Proceedings of the 26th Chinese Control and Decision Conference, 2014, pp. 642–645.
[26] Y.-D. Xiao, S.-Y. Lao, L.-L. Hou, L. Bai, Edge orientation for optimizing controllability of complex networks, Phys. Rev. E 90 (4) (2014) 042804.
[27] M. Liang, S. Jin, D. Wang, X. Zou, Optimization of controllability and robustness of complex networks by edge directionality, Eur. Phys. J. B 89 (9) (2016) 186.
[28] S. Pequito, V.M. Preciado, A.-L. Barabási, G.J. Pappas, Trade-offs between driving nodes and time-to-control in complex networks, Sci. Rep. 7 (2017) 39978.
[29] G. Ramos, S. Pequito, Generating complex networks with time-to-control communities, PLOS ONE 15 (2020) e0236753.
[30] A.-L. Barabási, E. Ravasz, T. Vices, Deterministic scale-free networks, Physica A 299 (2001) 559–564.
[31] S. Laird, H. Jensen, A steady state network model with a $1/k$ scale-free degree distribution, 2006.
[32] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener, Graph structure in the web, Comput. Netw. 33 (1) (2000) 309–320.
[33] T. Imae, K. Cai, On algebraic connectivity of directed scale-free networks, J. Franklin Inst. B 355 (16) (2018) 8065–8078.