Journal of Complex Networks (2022) **6**, Advance Access Publication on 28 November 2022
https://doi.org/10.1093/comnet/cnac054

Controllability of a class of swarm signalling networks

Peng Sun
Department of Quantum and Computer Engineering, Delft University of Technology, Delft, The Netherlands

Robert E. Kooij†
Department of Quantum and Computer Engineering, Delft University of Technology, Delft, The Netherlands
Unit Information and Communication Technology, TNO, The Hague, The Netherlands
†Corresponding author. Email: r.e.kooij@tudelft.nl

and

Roland Bouffanais
Applied Complexity Group, University of Ottawa, Ottawa, Canada

[Received on 20 March 2022; editorial decision on 7 November 2022; accepted on 10 November 2022]

In this article, we propose closed-form analytical expressions to determine the minimum number of driver nodes that is needed to control a specific class of networks. We consider swarm signalling networks with regular out-degree distribution where a fraction p of the links is unavailable. We further apply our method to networks with bi-modal out-degree distributions. Our approximations are validated through intensive simulations. Results show that our approximations have high accuracy when compared with simulation results for both types of out-degree distribution.

Keywords: network controllability; swarm signalling networks; driver nodes

1. Introduction

Network controllability is an essential property for the safe and reliable operation of real-world infrastructures, and as such this research area has attracted significant attention over the past decade [1–4]. For definiteness, a system is said to be controllable if it can be driven from any initial state to any desired final state by external inputs in finite time [5]. By blending classical control theory with concepts from network science, the notion of structural controllability has emerged [6]. Classically, let $\mathbf{A}$ be the $N \times N$ adjacency matrix of a given network with N nodes, while the connection of $\mathbf{M}$ input signals to the network is described by the $N \times M$ input matrix $\mathbf{B}$, where $M \leq N$. Then, the system characterized by $(\mathbf{A}, \mathbf{B})$ is structurally controllable if it is possible to find the non-zero parameters in $\mathbf{A}$ and $\mathbf{B}$ such that the obtained system $(\mathbf{A}, \mathbf{B})$ is controllable in the classical sense of satisfying the Kalman rank condition.

In their seminal article, Liu *et al.* [2] used maximum matching to get the minimum number N_D of driver nodes—that is, nodes driven by external inputs—that are needed to achieve structural controllability of a directed network. However, the results reported in Liu *et al.* [2] critically depend on the assumption that the network has no self-links, that is, a node's internal state can only be changed upon interaction with neighbouring nodes [7]. Yuan *et al.* [4] further proposed the concept of exact controllability based on the maximum multiplicity of all eigenvalues of the adjacency matrix A to find the driver nodes in networks. Ruths *et al.* [8] developed a theoretical framework for characterizing control profiles of networks.

Jia *et al.* [1] classified each node into one of three categories, based on its likelihood of being included in a minimum set of driver nodes and discovered bi-modal behaviour for the fraction of redundant nodes when the average degree of the networks is high. Yan *et al.* [9] investigated the relation between the maximum energy needed for controllability and the number of driver nodes. Nepusz *et al.* [3] indicated that most real-world networks are more controllable than their randomized counterparts. More recently, Zhang *et al.* [10] studied the change of network controllability in growing networks and found a lower bound for the maximum number of nodes that can be added to a network while keeping the number of driver nodes unchanged.

The robustness of network controllability under perturbation of the network topology has been investigated extensively. Lu *et al.* [11] discovered that a betweenness-based strategy is quite efficient to harm the controllability of real-world networks. Lou *et al.* [12] present a search for the network configuration with optimal robustness of controllability against random node-removal attacks. Wang *et al.* [13] proposed a dynamic cascading failure model and investigated the controllability robustness of real-world logistic networks. Nie *et al.* [14] found that the controllability of Erdős–Rényi random networks with a moderate average degree is less robust, whereas a scale-free network with moderate power-law exponent shows a stronger ability to maintain its controllability when these networks are under intentional link attack. Sun *et al.* [15] proposed closed-form analytic approximations for the minimum number of driver nodes needed to fully control networks, where links are removed according to both random and targeted attacks. Komareji *et al.* [16] discussed the resilience and controllability of dynamic collective behaviours for a class of swarm signalling networks (SSNs) [16]. The SSNs are modelled as directed (unweighted) graphs where the nodes have k-regular out-degree and Poisson-like in-degree distribution with average k. Following the paper by Liu *et al.* [2], an implicit equation is derived, whose solution leads to the minimum number of driver nodes to control the whole swarm [16]. However, upon validation of the formula given in [16] through simulation, we found significant differences between the analytical results and simulation results.

Beyond the theoretical interest in analytical results related to the controllability of complex networks, it is worth stressing that our particular focus on SSNs stems from their practical importance and ubiquity in a number of key problems related to collective behaviours and space-dependent collective decision-making [17]. By construction, the nodes of SSNs are embedded in the physical space and the specific nature of inter-agent interactions governs the distribution of edges. Hence, the SSN topology—with its particular in- and out-degree distributions, and high clustering—is a powerful abstraction to study the dynamics of these collective behaviours. For instance, when considering natural swarms—for example, schools of fish or flocks of birds—the concept of controllability of the SSN is key to explain how a single agent detecting a predator is capable of triggering a collective evasive manoeuvre [16, 18]. The analysis of the controllability of SSNs is even more important when considering artificial swarming systems: for example, groups of robots collectively moving in space [19], or performing a decentralized mapping of an open space [20], or aiming at achieving a spatial consensus [21]. In all these multi-robot systems, the tuning of the topology of the SSNs plays a key role in achieving the desired collective actions. Even for problems of social contagion in collective decision-making, the Kirchhoff index and clustering coefficients of the SSN have been found to be responsible for a transition from a simple social contagion to a complex one [22]. In all these natural, artificial and social systems, the effectiveness in achieving an effective collective response rests on the amplified influence of a few agents (i.e. nodes of the SSN) over the entire network. Therefore, a detailed understanding of the controllability of various types of SSN would offer valuable insights into the complex dynamics of this broad class of collective behaviours.

The aim of this article is threefold. First, we correct the assumption when calculating the minimum fraction of driver nodes given in [16] and back this up with simulations. Second, we generalize the results by considering SSNs in which a fraction p of the links is removed at random. Also for this case, we are capable of deriving an implicit equation, whose solution leads to the minimum number of driver nodes. Finally, we relax the condition that the out-degree is regular. Specifically, we consider bi-modal out-degree distributions, where the out-degree is k_1 for a fraction α of the nodes and k_2 for the remaining fraction $(1-\alpha)$ of the nodes. Note that the impact of having unavailable links is also considered a more general scenario.

2. Controllability of networks and driver nodes

2.1 *Controllability of networks*

A system is controllable if it can be driven from any initial state to any desired final state, by proper variable inputs, in finite time [5]. Most real systems are driven by nonlinear processes, but the controllability of non-linear systems is in many aspects structurally similar to that of linear systems [2]. The linear and time-invariant (LTI) dynamics on a directed network can be described by:

$$\frac{d\mathbf{x}(t)}{dt} = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \tag{2.1}$$

where the $N \times 1$ vector $\mathbf{x}(t) = (x_1(t), x_2(t), ..., x_N(t))^\top$ denotes the state of the system with N nodes at time t. The weighted matrix $\mathbf{A}$ is an $N \times N$ matrix which describes the network topology and the interaction strength between the components. The $N \times M$ matrix $\mathbf{B}$ is the input matrix that identifies the $M \leq N$ driver nodes controlled by outside input signals. The $M \times 1$ vector $\mathbf{u}(t) = (u_1(t), u_2(t), ..., u_M(t))^\top$ is the input signal vector. A driver node $j \in \{1, \ldots, M\}$ has an input signal $u_j(t)$ which is externally fed into it.

The LTI system defined by Eq. (2.1) is controllable if and only if the $N \times NM$ controllability matrix:

$$\mathbf{C} = (\mathbf{B}, \mathbf{AB}, \mathbf{A}^2\mathbf{B}, ..., \mathbf{A}^{N-1}\mathbf{B}), \tag{2.2}$$

has full rank, that is, $\mathrm{rank}(\mathbf{C}) = N$. This criterion is the so-called Kalman controllability rank condition [23]. The rank of the matrix $\mathbf{C}$ provides the dimension of the controllable subspace of the system. One therefore needs to find the right input matrix $\mathbf{B}$ consisting of a minimum number of driver nodes to ensure that the controllability matrix $\mathbf{C}$ has full rank.

2.2 *Driver nodes*

Liu *et al.* [2] proved that the minimum number of driver nodes needed for structural controllability, where the input signals are injected to control the directed network, can be obtained through the 'maximum matching' of the network. The source node of a directed link is defined as the node from which the link originates, while the target node is the node where the link terminates. A maximum matching of a directed network is a maximum set of links that do not share source or target nodes [24], which is illustrated in Fig. 1(a). Such links are coined 'matching links'. Target nodes of matching links are matched nodes, and the other nodes are unmatched nodes. For a given maximum matching, connecting driver nodes with unmatched nodes gives a minimum number of driver nodes N_D needed for controlling the network.

A directed network G with N nodes and L links can be converted into a bipartite graph $B_{N,N}$ with $2N$ nodes and L links in order to find the maximum number of matching links, so as to determine the

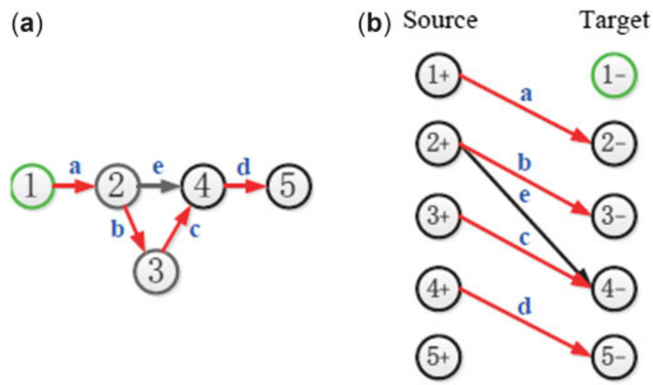

FIG. 1. Driver nodes and matching links (links a, b, c and d) in a directed network G. (a) An example network G with $N = 5$ nodes and $L = 5$ directed links. Node one is the only unmatched node. (b) The corresponding bipartite graph with $2N$ nodes and L links. By using the Hopcroft–Karp algorithm, a maximum set of matching links can be found in the bipartite graph. The target nodes of matching links are matched nodes. Other target nodes are unmatched nodes, which are also driver nodes.

minimum number of driver nodes N_D (see Fig. 1(b)). A maximum matching in a bipartite graph can be obtained efficiently by the Hopcroft–Karp algorithm [25] when the original directed network is small. The unmatched nodes in a maximum matching constitute a minimum set of driver nodes. It is worth noting that a minimum set of driver nodes is not necessarily unique. The Hopcroft–Karp algorithm guarantees to return the minimum number of driver nodes to completely control the network. In addition, the computational complexity of the Hopcroft–Karp algorithm to find all driver nodes is $O(\sqrt{N}L)$.

As discussed above, the Hopcroft–Karp algorithm works efficiently when the network is small and sparse. However, in real life, this is seldom the case. When the network is large and dense, the Hopcroft–Karp algorithm may no longer be a viable efficient option in finding the number of driver nodes. Even with the rapid increase in computational power, the use of Hopcroft–Karp algorithm can be rendered ineffective if one tries to identify the sensitivity of the number of driver nodes with respect to several parameters characterizing the SSN topology. For instance, as we will see with Theorem 5.3, the number n_D of driver nodes can have a very non-linear, implicit and intricate relationship with the parameters defining the degree distribution. In such cases, performing a systematic sensitivity analysis of the dependence of n_D with respect to these parameters using the Hopcroft–Karp algorithm would prove prohibitive. As an alternative, there exists a general expression for the minimum number N_D of driver nodes obtained by using generating functions [26], which is also provided in [2]. However, this approach requires the knowledge of the closed-form degree distribution of the network. In the rest of this article, we use this general expression to estimate the minimum number N_D of driver nodes in the SSNs with regular out-degree distribution and then deduce the general formula by considering the scenario when a fraction p of the links is unavailable. We subsequently relax the condition that the out-degree is regular and look into networks with bi-modal out-degree distributions.

3. Generating functions

In a network, let x denote the probability that a link is in state X. For example, X can denote the weight of each link in a weighted network: X can denote the existence of a link in an unweighted network. We assume that the states of links are independent from each other. Then, the probability that all the links of a node with degree k are in state X is x^k. Averaging this probability by the degree distribution of the network, we then obtain the probability that all the links of a randomly chosen node are in state X. According to the definition of the generating function [27], this probability can be written as

$$G(x) = \sum_{k=0}^{\infty} p_k x^k, \tag{3.1}$$

where p_k is the probability that a randomly chosen node in the network has degree k. Let $x = 1$, then we obtain $G(1) = \sum_{k=0}^{\infty} p_k = 1$. Besides, the average degree $\langle k \rangle$ of the network can be expressed as:

$$\langle k \rangle = G^{'}(1) = \sum_{k=0}^{\infty} k p_k. \tag{3.2}$$

Considering the degree of the node reached by following a randomly chosen link is k, the probability that all the other links of this node are in state X is x^{k-1}. The distribution of the degrees of the nodes reached by following a randomly chosen link is called the excess degree distribution q_k, which depends on the degree distribution p_k. Note that the larger p_k is, the larger q_k is. Furthermore, following a link, it is easier to reach a node with larger k. Hence, we have

$$q_k \propto k p_k. \tag{3.3}$$

The normalized distribution q_k is

$$q_k = \frac{k p_k}{\sum_{k=0}^{\infty} k p_k} = \frac{k p_k}{\langle k \rangle}. \tag{3.4}$$

Thus, the probability that all the other links of a node reached by following a randomly chosen link are in state X is given by

$$H(x) = \sum_{k=1}^{\infty} q_k x^{k-1} = \sum_{k=1}^{\infty} \frac{k p_k}{\langle k \rangle} x^{k-1} = \frac{G^{'}(x)}{G^{'}(1)}. \tag{3.5}$$

It must be highlighted that all these functions are based on the assumption that the states of links are independent from each other [26].

4. SSNs with k-regular out-degree

4.1 *Fraction of driver nodes in SSNs with k-regular out-degree*

It is shown in Liu *et al.* [2] that the minimum number of driver nodes can be obtained by using the following set of generating functions

$$G_{\mathrm{out}}(x) = \sum_{k_{\mathrm{out}}=0}^{\infty} P_{\mathrm{out}}(k_{\mathrm{out}}) x^{k_{\mathrm{out}}}, \tag{4.1}$$

$$G_{\mathrm{in}}(x) = \sum_{k_{\mathrm{in}}=0}^{\infty} P_{\mathrm{in}}(k_{\mathrm{in}}) x^{k_{\mathrm{in}}}, \tag{4.2}$$

$$H_{\text{out}}(x) = \sum_{k_{\text{out}}=1}^{\infty} \frac{k_{\text{out}} P_{\text{out}}(k_{\text{out}})}{\langle k_{\text{out}} \rangle} x^{k_{\text{out}}-1}, \tag{4.3}$$

$$H_{\text{in}}(x) = \sum_{k_{\text{in}}=1}^{\infty} \frac{k_{\text{in}} P_{\text{in}}(k_{\text{in}})}{\langle k_{\text{in}} \rangle} x^{k_{\text{in}}-1}, \tag{4.4}$$

where $P_{\text{out}}(\cdot)$ and $P_{\text{in}}(\cdot)$ denote the probability distribution function of the out- and in-degree, respectively, and $\langle k_{\text{out}} \rangle$ and $\langle k_{\text{in}} \rangle$ denote the average out- and in-degree, respectively.

Using those generating functions, the general expression for the minimum fraction N_D of driver nodes derived by Liu *et al.* [2] reads

$$n_D = \frac{N_D}{N} = \frac{1}{2}\{G_{\text{in}}(w_2) + G_{\text{in}}(1-w_1) - 2 + G_{\text{out}}(\hat{w}_2) + G_{\text{out}}(1-\hat{w}_1) + k(\hat{w}_1(1-w_2) + w_1(1-\hat{w}_2))\}, \tag{4.5}$$

where $w_1, w_2, \hat{w}_1$ and $\hat{w}_2$ satisfy

$$w_1 = H_{\text{out}}(\hat{w}_2), \tag{4.6}$$

$$w_2 = 1 - H_{\text{out}}(1-\hat{w}_1), \tag{4.7}$$

$$\hat{w}_1 = H_{\text{in}}(w_2), \tag{4.8}$$

$$\hat{w}_2 = 1 - H_{\text{in}}(1-w_1). \tag{4.9}$$

By construction, the out-degree distribution for the SSN suggested in [16] is a Dirac delta function, that is,

$$P_{\text{out}}(k_{\text{out}}) = \delta(k - k_{\text{out}}), \tag{4.10}$$

where k is the fixed out-degree for every node. Thus, the average out-degree $\langle k_{\text{out}} \rangle$ equals the out-degree k of each node. It is also shown in [16] that, for sufficiently large SSNs, the in-degree distribution closely resembles a Poisson distribution, with average k, that is,

$$P_{\text{in}}(k_{\text{in}}) = \frac{k^{k_{\text{in}}}}{k_{\text{in}}!} e^{-k}. \tag{4.11}$$

Using the degree distributions in Eqs (4.1)–(4.4), it follows

$$G_{\text{out}}(x) = x^k, \tag{4.12}$$

$$G_{\text{in}}(x) = e^{-k(1-x)}, \tag{4.13}$$

$$H_{\text{out}}(x) = x^{k-1}, \tag{4.14}$$

$$H_{\text{in}}(x) = e^{-k(1-x)}. \tag{4.15}$$

Therefore, the parameters $w_1, w_2, \hat{w}_1$ and $\hat{w}_2$ satisfy

$$w_1 = \hat{w}_2^{k-1}, \tag{4.16}$$

$$w_2 = 1 - (1 - \hat{w}_1)^{k-1}, \tag{4.17}$$

$$\hat{w}_1 = e^{-k(1-w_2)}, \tag{4.18}$$

$$\hat{w}_2 = 1 - e^{-kw_1}. \tag{4.19}$$

For the trivial case $k = 0$, it is easy to see that the above set of equations leads to $n_D = 1$, that is, all agents in the swarm need to be controlled, which makes sense because the out-degree of every node is 0 in this case. Also, for the case $k = 1$, Eqs (4.16)–(4.19) are solved for $w_1 = 1, w_2 = 0, \hat{w}_1 = e^{-1}$ and $\hat{w}_2 = 1 - e^{-1}$. Hence, for $k = 1$, it holds that $n_D = e^{-1}$.

For the case $k > 1$, Komareji and Bouffanais [16] argue that the smallest solution of the pair of equations (4.16) and (4.19) is given by $w_1 = \hat{w}_2 = 0$, and assuming that w_1 and $\hat{w}_2$ are indeed zero, the following expression for the fraction of driver nodes is derived:

$$n_D = \frac{1}{2}\{(1 - e^{-k(1-w_2)})^k - 1 + e^{-k(1-w_2)} + k(1 - w_2)e^{-k(1-w_2)}\}, \tag{4.20}$$

where w_2 is the solution of the implicit equation

$$1 - w_2 = (1 - e^{-k(1-w_2)})^{k-1}. \tag{4.21}$$

From Eq. (4.20), the asymptotic behaviour of n_D for large k can also be derived:

$$n_D \approx \frac{1}{2}e^{-k}. \tag{4.22}$$

However, upon simulation of SSNs, determining the fraction of driver nodes by applying the maximum matching algorithm, as described in [2], we found a discrepancy between Eq. (4.20) and the simulation results shown in Fig. 2. We generate 10000 directed networks with $N = 10000$ nodes each having an out-degree k whose value ranges from 1 to 8. The fraction n_D of driver nodes is the average fraction of driver nodes over 10000 networks for each out-degree k. As shown in Fig. 2, the result from Eq. (4.20) fits well with the simulation result at $k = 1$. However, the difference between Eq. (4.20) and simulation results is obvious for other values of k. For example, at all points $k > 1$, the results from the simulation are about two times the results given by Eq. (4.20).

The discrepancy is due to the assumption that one can choose the solution of equations (4.16) and (4.19) given by $w_1 = \hat{w}_2 = 0$. One can also argue that the pair of equations (4.16) and (4.19) is equivalent to the pair of equations (4.17) and (4.18). If we assume

$$w_1 = 1 - w_2, \tag{4.23}$$

$$\hat{w}_2 = 1 - \hat{w}_1, \tag{4.24}$$

then the pair of equations (4.17) and (4.18) follows from the pair of equations (4.16)–(4.19). As a result, applying Eq. (4.5) leads to the following expression for the fraction of driver nodes:

$$n_D = ((1 - e^{-k(1-w_2)})^k - 1 + e^{-k(1-w_2)} + k(1 - w_2)e^{-k(1-w_2)}), \tag{4.25}$$

where w_2 is still the solution of equation (4.21).

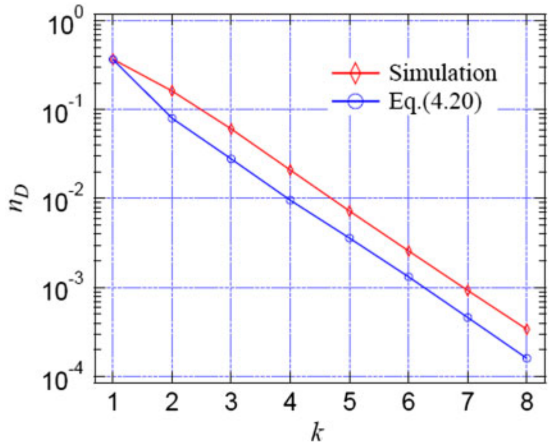

FIG. 2. Fraction of driver nodes n_D for different values of the out-degree k: Eq. (4.20) versus simulation results.

TABLE 1 *Comparison of Eqs (4.25) and (4.26) with simulation results*

k	Eq. (4.25)		Eq. (4.26)		Simulations
	Value	r (%)	Value	r (%)	
1	0.367879	0.0079	0.367879	0.0079	0.36782
2	0.161903	0.40	0.135335	16.07	0.162003
3	0.060759	0.29	0.049787	17.82	0.06068
4	0.020916	0.28	0.018316	12.18	0.020943
5	0.007262	0.93	0.006738	6.35	0.007221
6	0.002578	0.23	0.002479	4.06	0.002561
7	0.00093	2.76	0.000912	0.77	0.000929
8	0.000339	5.93	0.000335	4.69	0.000346

The asymptotic behaviour of n_D for large k becomes:

$$n_D \approx e^{-k}. \tag{4.26}$$

Note that Eq. (4.25) also holds for $k = 1$, another indication of its correctness.

Table 1 shows the comparison between the approximations in Eqs (4.25) and (4.26) and the simulations.

Like previously, we generate 10000 directed networks with $N = 10000$ for each out-degree k whose value ranges from 1 to 8. The fraction of driver nodes n_D is the average fraction of driver nodes in 10000 networks. Then we calculate the analytical results from Eqs (4.25) and (4.26) and also the corresponding absolute relative error r. As shown in Table 1, the absolute relative errors of our approximation are less than 1% for k from 1 to 6. For the case where $k = 7$ and $k = 8$, the absolute relative errors are still small—less than 6%. When the values of k are small, the absolute relative errors of equation (4.26) are large.

We conclude from Table 1 that the simulations are an excellent fit for our approximation in Eq. (4.25). Also, the asymptotic approximation Eq. (4.26) is increasingly accurate for increasing k.

4.2 *Fraction of driver nodes under random link failures*

In this section, we generalize the results of the previous section by considering SSNs with k-regular out-degree, but now, we assume that a fraction p of the links is removed at random. This assumption is in accordance with some real-life scenarios, such as the communication disconnection between robots in swarm robotic networks because of the limited range of communication.

In what follows, we show that the analysis that led to our implicit approximations is still valid and applicable for this case. A crucial step is to find expressions for the generating functions Eqs (4.1)–(4.4) for this specific case involving a fraction of link failures.

The following lemma is instrumental in establishing the key results for this case—see [28] which gives an expression for the degree distribution, after removing m links uniformly at random.

LEMMA 4.1 After removing m links in a uniform and random way from a network $G_0(N, L)$, with degree distribution $Pr[D_{G_0} = j]$, the degree distribution $Pr[D_G = i]$ of the new network G satisfies:

$$Pr[D_G = i] = (1-p)^i \sum_{j=i}^{N-1} \binom{j}{i} p^{j-i} Pr[D_{G_0} = j], \tag{4.27}$$

where $p = \frac{m}{L}$ denotes the fraction of removed links in the original network G_0.

THEOREM 4.1 Consider a directed network with a k-regular out-degree and a Poisson in-degree distribution with an average k. Upon removing uniformly and at random a fraction p of the links, the generating functions $\bar{G}_{\text{out}}(x)$ and $\bar{G}_{\text{in}}(x)$ of the out- and in-degree, respectively, satisfy

$$\bar{G}_{\text{out}}(x) = (p + (1-p)x)^k, \tag{4.28}$$

$$\bar{G}_{\text{in}}(x) = e^{-k(1-p)(1-x)}. \tag{4.29}$$

The proof of Theorem 4.1 is given in Appendix A. Note that for the case without link removals, that is, $p = 0$, Eqs (4.28) and (4.29) reduce to Eqs (4.12) and (4.13). Also, we can deduce from Eqs (4.28) and (4.29) directly that both the average out- and in-degree after link removals, which is denoted by $\bar{k}$, equal

$$\bar{k} = k(1-p). \tag{4.30}$$

THEOREM 4.2 Consider a directed network with a k-regular out-degree and a Poisson in-degree with an average k. Then, after removing uniformly at random a fraction p of the links, the generating functions $\bar{H}_{\text{out}}(x)$ and $\bar{H}_{\text{in}}(x)$ of the excess out- and in-degree, respectively, satisfy

$$\bar{H}_{\text{out}}(x) = (p + (1-p)x)^{k-1} \tag{4.31}$$

$$\bar{H}_{\text{in}}(x) = e^{-k(1-p)(1-x)} \tag{4.32}$$

The proof of Theorem 4.2 is given in Appendix A. Note that for the case without link removals, that is, $p = 0$, Eqs (4.31) and (4.32) reduce to Eqs (4.14) and (4.15).

The results in Theorems 4.1 and 4.2 can also be directly deduced by using a result from [28]: if the generating function for the degree distribution for a network is given by $G(x)$, then the generating function $\bar{G}(x)$ for the resulting network after a fraction p of links are randomly removed, satisfies $\bar{G}(x) = G(p + (1-p)x)$. In addition, Theorem 4.2 can also be established directly by applying Eq. (3.5) to Eqs (4.28) and (4.29).

We are now in a position to state the following result.

THEOREM 4.3 Consider a directed network with a k-regular out-degree and a Poisson in-degree distribution with an average k. Then, after removing uniformly at random a fraction p of its links, the fraction of the minimum number of driver nodes is given by

$$\begin{aligned} n_D = (p + (1-p)(1 - e^{-k(1-p)(1-w_2)}))^k - 1 + e^{-k(1-p)(1-w_2)} \\ + k(1-p)(1-w_2)e^{-k(1-p)(1-w_2)}, \end{aligned} \tag{4.33}$$

where w_2 satisfies

$$1 - w_2 = (p + (1-p)(1 - e^{-k(1-p)(1-w_2)}))^{k-1}. \tag{4.34}$$

The asymptotic behaviour of n_D for large k is given by

$$n_D \approx e^{-k(1-p)}. \tag{4.35}$$

It is worth noting that for the particular case without link removals, that is, $p = 0$, Eqs (4.33)–(4.35) reduce to Eqs (4.25)–(4.21)–(4.26), respectively. The proof of Theorem 4.3 is given in Appendix A.

Table 2 shows the comparison between the approximations in Eqs (4.33) and (4.35) and simulations, for the cases $p = 0.2$ and $p = 0.5$. Specifically, we generated 1000 directed networks with $N = 10000$ with out-degree k, where $k \in \{1, 2, 3, 4, 5, 6, 7, 8\}$. For each network with the same out-degree k, we randomly removed a fraction p of links and get the value of n_D, and then repeat this process one thousand times. Thus, the fraction of driver nodes n_D for a combination (k, p) is the average fraction of driver nodes in 10^6 realizations.

As shown in Table 2, the absolute relative errors r of our approximation Eq. (4.33) are small—less than 4% for any k value when $p = 0.2$ or $p = 0.5$. In contrast, the relative errors of the asymptotic approximation Eq. (4.35) are large for most cases.

We conclude from Table 2 that the simulations yield an excellent fit with our approximation Eq. (4.33). Unsurprisingly, the asymptotic approximation Eq. (4.35) is increasingly accurate for increasing k.

Finally, Fig. 3 shows the fraction of driver nodes n_D as a function of the out-degree k for several values of p. The value of n_D decreases as the degree of networks increases for a specific p. Note that for the same k value, a larger value of p leads to a larger value of n_D.

TABLE 2 *Comparison of Eqs (4.33)–(4.35) with simulation results*

k	Eq. (4.33)				Eq. (4.35)				Simulations	
	$p = 0.2$	r (%)	$p = 0.5$	r (%)	$p = 0.2$	r (%)	$p = 0.5$	r (%)	$p = 0.2$	$p = 0.5$
1	0.442926	1.41	0.584101	3.72	0.449329	0.019	0.606531	0.021	0.449321	0.606622
2	0.238827	0.30	0.410116	0.12	0.201897	15.21	0.367879	10.20	0.238905	0.410229
3	0.116278	0.30	0.279218	0.24	0.090718	21.75	0.22313	19.89	0.116176	0.279108
4	0.050341	0.38	0.183439	0.19	0.040762	18.72	0.135335	26.08	0.050167	0.183421
5	0.021143	0.96	0.112696	0.29	0.018316	12.53	0.082085	26.95	0.021215	0.112680
6	0.009002	0.13	0.065394	0.55	0.00823	8.70	0.049787	23.45	0.009041	0.065339
7	0.003902	0.20	0.037384	1.18	0.003698	5.41	0.030197	18.92	0.003915	0.03736
8	0.001714	2.50	0.021502	0.20	0.001662	5.5	0.018316	14.65	0.001706	0.021533

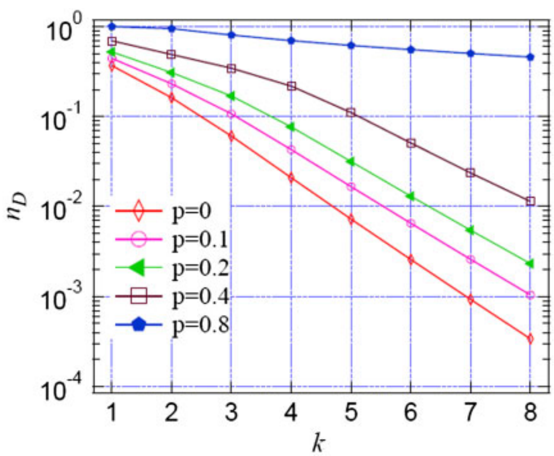

FIG. 3. Fraction of driver nodes as function of the out-degree k for several values of the fraction of removed links p.

5. SSNs with a bi-modal out-degree

5.1 *Fraction of driver nodes in SSNs with a bi-modal out-degree*

In this section, we generalize the results of one of the previous sections by considering SSNs with a bi-modal out-degree distribution, that is, we assume that for a fraction α of nodes the out-degree is k_1, while for the remaining fraction $1 - \alpha$ of nodes, the out-degree equals k_2. We will assume $k_1 \neq k_2$ and both k_1 and k_2 are strictly larger than 0.

THEOREM 5.1 Consider a directed network with a bi-modal out-degree distribution $\alpha\delta(k_{\text{out}} - k_1) + \alpha\delta(k_{\text{out}} - k_2)$, with average out-degree

$$k = \alpha k_1 + (1 - \alpha)k_2, \tag{5.1}$$

and a Poisson in-degree distribution with an average k. The generating functions $\hat{G}_{\text{out}}(x)$ and $\hat{G}_{\text{in}}(x)$ of the out- and in-degree, respectively, satisfy

$$\hat{G}_{\text{out}}(x) = \alpha x^{k_1} + (1-\alpha)x^{k_2}, \tag{5.2}$$

$$\hat{G}_{\text{in}}(x) = e^{-k(1-x)}. \tag{5.3}$$

The proof of Theorem 5.1 is given in Appendix B.

THEOREM 5.2 Consider a directed network with bi-modal out-degree $\alpha\delta(k_{\text{out}} - k_1) + \alpha\delta(k_{\text{out}} - k_2)$, with an average out-degree

$$k = \alpha k_1 + (1-\alpha)k_2 \tag{5.4}$$

and a Poisson in-degree distribution with average k. Then, the generating functions $\hat{H}_{\text{out}}(x)$ and $\hat{H}_{\text{in}}(x)$ of the excess out- and in-degree, respectively, satisfy

$$\hat{H}_{\text{out}}(x) = \frac{\alpha k_1 x^{k_1-1} + (1-\alpha)k_2 x^{k_2-1}}{k}, \tag{5.5}$$

$$\hat{H}_{\text{in}}(x) = e^{-k(1-x)}. \tag{5.6}$$

The Proof of Theorem 5.2 is given in Appendix B. The proof also can be established by applying Eq. (3.5) directly to Eqs (5.2) and (5.3). Note for the case $k_1 = k_2 = k$, where the out-degree reduces to a Dirac function, Eqs (5.2)–(5.6) reduce to Eqs (4.12)–(4.15).

THEOREM 5.3 Consider a directed network with a bi-modal out-degree $\alpha\delta(k_{\text{out}} - k_1) + \alpha\delta(k_{\text{out}} - k_2)$, with average out-degree $k = \alpha k_1 + (1-\alpha)k_2$ and a Poisson in-degree with average k. Then, the fraction of the minimum number of driver nodes is given by

$$n_D = \alpha(1-e^{-k(1-w_2)})^{k_1} + (1-\alpha)(1-e^{-k(1-w_2)})^{k_2} - 1 + e^{-k(1-w_2)} + ke^{-k(1-w_2)}(1-w_2), \tag{5.7}$$

where w_2 satisfies

$$1 - w_2 = \frac{\alpha k_1(1-e^{-k(1-w_2)})^{k_1-1} + (1-\alpha)k_2(1-e^{-k(1-w_2)})^{k_2-1}}{k}. \tag{5.8}$$

The asymptotic behaviour of n_D for large k is given by

$$n_D \approx e^{-k}. \tag{5.9}$$

Note for the case $k_1 = k_2 = k$, where the out-degree reduces to a Dirac function, Eqs (5.7)–(5.9) reduce to Eqs (4.25)–(4.21)–(4.26), respectively. It is worth noting that Eqs (5.7)–(5.8) reveal a complex dependency of n_D with respect to the parameters characterizing the degree distribution, namely (α, k_1, k_2). This is particularly apparent given the implicit nature of Eq. (5.8). In such cases, identifying the intricate

TABLE 3 *Comparing Eqs (5.7)–(5.9) with simulation results*

k_1	k_2	k	α	Eq. (5.7)		Eq. (5.9)		Simulation
				Value	r (%)	Value	r (%)	
1	3	2.5	0.25	0.107746	0.51	0.082085	23.43	0.107795
1	3	2	0.5	0.183062	0.020	0.135335	26.09	0.181395
1	3	1.5	0.75	0.273670	0.040	0.223130	18.44	0.273455
2	4	3.5	0.25	0.036402	0.56	0.030197	16.58	0.036705
2	4	3	0.5	0.063648	0.27	0.049787	21.57	0.06352
2	4	2.5	0.75	0.106955	0.25	0.082085	23.44	0.106735
2	6	5	0.25	0.007355	1.04	0.006738	9.30	0.007315
2	6	4	0.5	0.022172	0.76	0.018316	16.76	0.022335
2	6	3	0.75	0.071349	0.19	0.049787	30.09	0.071875
2	8	6.5	0.25	0.001555	3.81	0.001503	0.33	0.001595
2	8	5	0.5	0.007556	2.20	0.006738	8.86	0.007745
2	8	3.5	0.75	0.045382	0.35	0.030197	33.69	0.04665
4	6	5.5	0.25	0.004324	0.68	0.004087	4.84	0.004362
4	6	5	0.5	0.007293	0.97	0.006738	6.71	0.007181
4	6	4.5	0.75	0.012357	1.40	0.011109	8.34	0.01228
4	8	7	0.25	0.000931	3.22	0.000912	5.20	0.000962
4	8	6	0.5	0.002593	4.18	0.002479	4.36	0.002706
4	8	5	0.75	0.007354	1.17	0.006738	7.56	0.007269

dependency of n_D with those parameters using the Hopcroft–Karp algorithm would require an impractical brute-force approach.

The Proof of Theorem 5.3 is given in Appendix B.

Table 3 shows the comparison between the approximations in Eqs (5.7) and (5.9) and simulations.

We generate 10000 directed networks with $N = 10000$ for each out-degree combination (k_1, k_2, α) and obtain the average fraction n_D of driver nodes. As shown in Table 3, the absolute relative errors r of our approximation Eq. (5.7) are small, thereby indicating a good fit with simulations. The absolute relative errors of Eq. (5.9) are larger, especially for small average degree $k = \alpha k_1 + (1-\alpha)k_2$.

We conclude from Table 3 that the simulations constitute a very good fit with our approximation Eq. (5.7). Also, the asymptotic approximation Eq. (5.9) is increasingly accurate for increasing k.

5.2 *Fraction of driver nodes under random link failures*

In this section, we generalize the results of the previous section by considering again SSNs with a bi-modal out-degree, but now, we assume that a fraction p of the links is removed at random. We show that the analysis that led to our implicit approximations can also be conducted for this case. Similar to the case with a regular out-degree, a crucial step is to find expressions for the generating functions Eqs (4.1)–(4.4) for this particular case.

Based on Lemma 1, we get:

THEOREM 5.4 Consider a directed network with a bi-modal out-degree $\alpha\delta(k_{\text{out}}-k_1)+(1-\alpha)\delta(k_{\text{out}}-k_2)$, with average out-degree

$$k = \alpha k_1 + (1-\alpha)k_2 \tag{5.10}$$

and a Poisson in-degree with an average k. Then, after removing uniformly at random a fraction p of the links, the generating functions $\tilde{G}_{\text{out}}(x)$ and $\tilde{G}_{\text{in}}(x)$ of the out- and in-degree, respectively, satisfy

$$\tilde{G}_{\text{out}}(x) = \alpha(p+(1-p)x)^{k_1} + (1-\alpha)(p+(1-p)x)^{k_2}, \tag{5.11}$$

$$\tilde{G}_{\text{in}}(x) = e^{-k(1-p)(1-x)}. \tag{5.12}$$

By applying the generating function $\bar{G}(x)$ for the resulting network after a fraction p of links are randomly removed [28], the theorem also follows directly from $\tilde{G}_{\text{out}}(x) = \hat{G}_{\text{out}}(p+(1-p)x)$ and $\tilde{G}_{\text{in}}(x) = \hat{G}_{\text{in}}(p+(1-p)x)$. Note that for the case without link removals, that is, $p = 0$, Eqs (5.11) and (5.12) reduce to Eqs (5.2) and (5.3). Also, we can deduce from Eqs (5.11)–(5.12) directly that both the average out- and in-degree after link removals, which we denote by $\tilde{k}$, and satisfies

$$\tilde{k} = k(1-p). \tag{5.13}$$

THEOREM 5.5 Consider a directed network with a bi-modal out-degree $\alpha\delta(k_{\text{out}}-k_1)+(1-\alpha)\delta(k_{\text{out}}-k_2)$, with average out-degree

$$k = \alpha k_1 + (1-\alpha)k_2, \tag{5.14}$$

and a Poisson in-degree with an average k. Then, after removing uniformly at random a fraction p of the links, the generating functions $\bar{H}_{\text{out}}(x)$ and $\bar{H}_{\text{in}}(x)$ of the excess out- and in-degree, respectively, satisfy

$$\tilde{H}_{\text{out}}(x) = \frac{\alpha k_1(p+(1-p)x)^{k_1-1} + (1-\alpha)k_2(p+(1-p)x)^{k_2-1}}{k}, \tag{5.15}$$

$$\tilde{H}_{\text{in}}(x) = e^{-k(1-p)(1-x)}. \tag{5.16}$$

The Proof of Theorem 5.5 can readily be obtained by combining the Proofs of Theorems 4.2 and 5.2. By applying the generating function $\bar{G}(x)$ for the resulting network after a fraction p of links are randomly removed [28], the theorem also follows directly from $\tilde{H}_{\text{out}}(x) = \hat{H}_{\text{out}}(p+(1-p)x)$ and $\tilde{H}_{\text{in}}(x) = \hat{H}_{\text{in}}(p+(1-p)x)$. Note that for the case without link removals, that is, $p = 0$, Eqs (5.15) and (5.16) reduce to Eqs (5.5) and (5.6).

After obtaining expressions for all required generation functions, we are now in the position to state the following result.

THEOREM 5.6 Consider a directed network with a bi-modal out-degree $\alpha\delta(k_{\text{out}}-k_1)+(1-\alpha)\delta(k_{\text{out}}-k_2)$, with average out-degree $k = \alpha k_1 + (1-\alpha)k_2$ and a Poisson in-degree with an average k. Then, after

TABLE 4 *Comparison of approximation Eq. (5.17) with simulation results*

k_1	k_2	k	α	Eq. (5.17)				Simulation	
				$p = 0.2$	r (%)	$p = 0.5$	r (%)	$p = 0.2$	$p = 0.5$
1	3	2.5	0.25	0.251484	0.50	0.541569	0.19	0.252746	0.540569
1	3	2	0.5	0.340662	0.41	0.627028	1.29	0.342065	0.619028
1	3	1.5	0.75	0.431100	0.29	0.709013	2.21	0.432370	0.693714
2	4	3.5	0.25	0.122113	0.25	0.410770	0.29	0.121813	0.409569
2	4	3	0.5	0.183813	1.32	0.476848	0.43	0.186273	0.474822
2	4	2.5	0.75	0.247667	0.80	0.535501	0.43	0.245692	0.537824
2	6	5	0.25	0.033257	0.87	0.299464	0.52	0.033549	0.297913
2	6	4	0.5	0.094631	1.86	0.435961	1.48	0.096426	0.429607
2	6	3	0.75	0.216405	0.93	0.514376	3.06	0.218443	0.499125
2	8	6.5	0.25	0.008650	0.06	0.101497	17.49	0.008655	0.123010
2	8	5	0.5	0.037450	0.81	0.406573	0.15	0.037150	0.405974
2	8	3.5	0.75	0.204397	0.017	0.505728	1.00	0.204363	0.510815
4	6	5.5	0.25	0.020441	5.68	0.163736	0.14	0.021671	0.163504
4	6	5	0.5	0.032167	4.57	0.229064	0.44	0.033706	0.228061
4	6	4.5	0.75	0.050380	1.00	0.288043	0.80	0.049880	0.285759
4	8	7	0.25	0.005504	1.47	0.058532	0.33	0.005586	0.058338
4	8	6	0.5	0.013368	0.077	0.135230	0.42	0.013357	0.134664
4	8	5	0.75	0.033187	0.27	0.265665	0.93	0.033275	0.263211

removing uniformly at random a fraction p of the links, the fraction of minimum number of driver nodes is given by:

$$\begin{aligned} n_D = \alpha(p + (1-p)(1 - e^{-k(1-\omega_2)}))^{k_1} + (1-\alpha)(p + (1-p)(1 - e^{-k(1-\omega_2)}))^{k_2} \\ -1 + e^{-k(1-p)(1-\omega_2)} + k(1-p)e^{-k(1-\omega_2)}(1-\omega_2) \end{aligned} \tag{5.17}$$

where ω_2 satisfies

$$\begin{aligned} & 1-\omega_2 \\ = \frac{\alpha k_1 (p + (1-p)(1 - e^{-k(1-p)(1-\omega_2)}))^{k_1-1} + (1-\alpha)k_2(p + (1-p)(1 - e^{-k(1-p)(1-\omega_2)}))^{k_2-1}}{k}&. \end{aligned} \tag{5.18}$$

The asymptotic behaviour of n_D for large k is given by

$$n_D \approx e^{-k(1-p)}. \tag{5.19}$$

For the case without link removals, that is, $p = 0$, Eqs (5.17)–(5.19) reduce to Eqs (5.7)–(5.9).

The Proof of Theorem 5.6 is given in Appendix C.

As a final step, to verify our approximation Eq. (5.17), we generate 1000 directed networks with $N = 10000$ for each out-degree combination (k_1, k_2, α). For each network with the same out-degree combination (k_1, k_2, α), we randomly remove a fraction p of links and get the value of n_D, and then repeat this process for 1000 times. Thus, the fraction of driver nodes n_D for a combination (k_1, k_2, α, p) is the average fraction of driver nodes in 10^6 realizations.

Table 4 shows the comparison between Eq. (5.17) and simulations. In most cases, the relative errors between Eq. (5.17) and simulations are small. We conclude from Table 4 that the simulations are a very robust fit with our approximation Eq. (5.17).

6. Conclusion

In this article, we correct the formula given in [16] for the minimum number of driver nodes for a specific class of SSNs, which are characterized by a regular out-degree. We then generalize the results by considering SSNs with a regular out degree k where a fraction p of the links is unavailable. For this case, we derive an implicit equation, whose solution leads to the minimum number of driver nodes. We find that our approximation fits well with the simulation results. Finally, we relax the condition that the out-degree is regular and look into bi-modal out-degree distributions. For this case, we also consider scenarios with unavailable links. We derive an implicit equation and verify its accuracy. We find that our approximation for bi-modal out-degree distribution fits well with simulation results.

Appendix A

Here, we will give the proof of Theorem 4.1. The out-degree distribution $P_{\text{out}}(\cdot)$ for the unperturbed network is given in Eq. (4.10). Let us denote the out-degree distribution for the perturbed network by $\bar{P}_{\text{out}}(\cdot)$. Then it follows from Lemma 4.1 and Eq. (4.10) that

$$\bar{P}_{\text{out}}(k_{\text{out}}) = (1-p)^{k_{\text{out}}} \sum_{j=k_{\text{out}}}^{N-1} \binom{j}{k_{\text{out}}} p^{j-k_{\text{out}}} \delta(k-j). \tag{A.1}$$

Therefore, we obtain

$$\bar{P}_{\text{out}}(k_{\text{out}}) = 0, \tag{A.2}$$

if $k_{\text{out}} > k$ and

$$\bar{P}_{\text{out}}(k_{\text{out}}) = (1-p)^{k_{\text{out}}} \binom{k}{k_{\text{out}}} p^{k-k_{\text{out}}} \tag{A.3}$$

if $k_{\text{out}} \leq k$. From this, we get

$$\begin{aligned} \bar{G}_{\text{out}}(x) &= \sum_{k_{\text{out}}=0}^{\infty} \bar{P}_{\text{out}}(k_{\text{out}}) x^{k_{\text{out}}} = \sum_{k_{\text{out}}=0}^{k} (1-p)^{k_{\text{out}}} \binom{k}{k_{\text{out}}} p^{k-k_{\text{out}}} x^{k_{\text{out}}} \\ &= \sum_{k_{\text{out}}=0}^{k} \binom{k}{k_{\text{out}}} ((1-p)x)^{k_{\text{out}}} p^{k-k_{\text{out}}} = (p + (1-p)x)^k. \end{aligned} \tag{A.4}$$

This proves that Eq. (4.28) holds.

We assumed that the in-degree distribution of the original graph follows a Poisson distribution, see (4.11) but for finite N the actual distribution is binomial. However, for $N \longrightarrow \infty$ the limiting distribution is indeed Poissonian. Therefore, for proving that Eq. (4.29) holds, we will use Lemma 4.1 with $N = \infty$. The in-degree distribution $P_{\text{in}}(\cdot)$ for the unperturbed network is given in Eq. (4.11). Let us denote the in-degree distribution for the perturbed network by $\bar{P}_{\text{in}}(\cdot)$. Then, it follows from Lemma 4.1 and Eq. (4.11) that

$$\bar{P}_{\text{in}}(k_{\text{in}}) = (1-p)^{k_{\text{in}}} \sum_{j=k_{\text{in}}}^{\infty} \binom{j}{k_{\text{in}}} p^{j-k_{\text{in}}} \frac{k^j}{j!} e^{-k}. \tag{A.5}$$

From this, we get

$$\begin{aligned}
\bar{G}_{\text{in}}(x) = \sum_{k_{\text{in}}=0}^{\infty} \bar{P}_{\text{in}}(k_{\text{in}}) x^{k_{\text{in}}} &= \sum_{k_{\text{in}}=0}^{\infty} (1-p)^{k_{\text{in}}} \sum_{j=k_{\text{in}}}^{\infty} \binom{j}{k_{\text{in}}} p^{j-k_{\text{in}}} \frac{k^j}{j!} e^{-k} x^{k_{\text{in}}} \\
&= e^{-k} \sum_{k_{\text{in}}=0}^{\infty} \left(\frac{(1-p)x}{p}\right)^{k_{\text{in}}} \sum_{j=k_{\text{in}}}^{\infty} \binom{j}{k_{\text{in}}} \frac{(pk)^j}{j!} \\
&= e^{-k} \sum_{k_{\text{in}}=0}^{\infty} \left(\frac{(1-p)x}{p}\right)^{k_{\text{in}}} \sum_{j=k_{\text{in}}}^{\infty} \frac{1}{k_{\text{in}}!} \frac{(pk)^j}{(j-k_{\text{in}})!} \\
&= e^{-k} \sum_{k_{\text{in}}=0}^{\infty} \left(\frac{(1-p)x}{p}\right)^{k_{\text{in}}} \frac{1}{k_{\text{in}}!} \sum_{j=k_{\text{in}}}^{\infty} \frac{(pk)^{j-k_{\text{in}}} (pk)^{k_{\text{in}}}}{(j-k_{\text{in}})!} \\
&= e^{-k} \sum_{k_{\text{in}}=0}^{\infty} \left(\frac{(1-p)x}{p}\right)^{k_{\text{in}}} \frac{(pk)^{k_{\text{in}}}}{k_{\text{in}}!} \sum_{\hat{j}=0}^{\infty} \frac{(pk)^{\hat{j}}}{\hat{j}!} \\
&= e^{-k} \sum_{k_{\text{in}}=0}^{\infty} \frac{(k(1-p)x)^{k_{\text{in}}}}{k_{\text{in}}!} e^{pk} \\
&= e^{-k} e^{k(1-p)x} e^{pk} = e^{-k(1-p)(1-x)}.
\end{aligned} \tag{A.6}$$

This proves that Eq. (4.29) holds.

Next, we will prove Theorem 4.2. Using the same notation as before, it follows from Eq. (4.3) that for the perturbed system the generating function $\bar{H}_{\text{out}}(x)$ is given by

$$\bar{H}_{\text{out}}(x) = \sum_{k_{\text{out}}=1}^{\infty} \frac{k_{\text{out}} \bar{P}_{\text{out}}(k_{\text{out}})}{\langle k_{\text{out}} \rangle} x^{k_{\text{out}}-1} \tag{A.7}$$

Then, using Eqs (A.2) and (A.3), we obtain

$$\begin{aligned}\bar{H}_{\text{out}}(x) &= \sum_{k_{\text{out}}=1}^{k} \frac{k_{\text{out}}(1-p)^{k_{\text{out}}}\binom{k}{k_{\text{out}}}p^{k-k_{\text{out}}}}{k(1-p)} x^{k_{\text{out}}-1} \\ &= \sum_{k_{\text{out}}=1}^{k} \binom{k-1}{k_{\text{out}}-1} p^{k-k_{\text{out}}}((1-p)x)^{k_{\text{out}}-1} \\ &= \sum_{m=0}^{k-1} \binom{k-1}{m} p^{k-1-m}((1-p)x)^{m} = (p+(1-p)x)^{k-1}.\end{aligned} \tag{A.8}$$

Finally, we prove Eq. (4.32).

Using the same notation as before, it follows from Eq. (4.4) that for the perturbed system the generating function $\bar{H}_{\text{in}}(x)$ is given by

$$\bar{H}_{\text{in}}(x) = \sum_{k_{\text{in}}=1}^{\infty} \frac{k_{\text{in}}\bar{P}_{\text{in}}(k_{\text{in}})}{\langle k_{\text{in}}\rangle} x^{k_{\text{in}}-1}. \tag{A.9}$$

Then, using Eq. (A.5), we obtain

$$\begin{aligned}\bar{H}_{\text{in}}(x) &= \sum_{k_{\text{in}}=1}^{\infty} \frac{k_{\text{in}}(1-p)^{k_{\text{in}}}}{k(1-p)} \sum_{j=k_{\text{in}}}^{\infty} \binom{j}{k_{\text{in}}} p^{j-k_{\text{in}}} \frac{k^j}{j!} e^{-k} x^{k_{\text{in}}-1} \\ &= e^{-k} \sum_{k_{\text{in}}=1}^{\infty} \frac{k_{\text{in}}(k(1-p)x)^{k_{\text{in}}}}{xk(1-p)k_{\text{in}}!} e^{pk} = e^{-k+pk} \sum_{k_{\text{in}}=1}^{\infty} \frac{(k(1-p)x)^{k_{\text{in}}-1}}{(k_{\text{in}}-1)!} \\ &= e^{-k+pk} \sum_{m=0}^{\infty} \frac{(k(1-p)x)^{m}}{m!} = e^{-k+pk+k(1-p)x} = e^{-k(1-p)(1-x)}.\end{aligned} \tag{A.10}$$

This finishes the proof of Theorem 4.2.

Proof of Theorem 4.3

Using Theorems 4.1 and 4.2, the set of equations (4.6)–(4.9) becomes

$$w_1 = (p+(1-p)\hat{w}_2)^{k-1} \tag{A.11}$$

$$\hat{w}_2 = 1 - e^{-k(1-p)w_1} \tag{A.12}$$

$$w_2 = 1 - (p+(1-p)(1-\hat{w}_1))^{k-1} \tag{A.13}$$

$$\hat{w}_1 = e^{-k(1-p)(1-w_2).} \tag{A.14}$$

By setting $\hat{w}_2 = 1 - \hat{w}_1$ and $w_1 = 1 - w_2$, it follows that the pair of Eqs (A.11) and (A.12) is equivalent to the pair of Eqs (A.13) and (A.14).

From this it follows that n_D in Eq. (4.5) becomes

$$n_D = \bar{G}_{\rm out}(1-\hat{w}_1) + \bar{G}_{\rm in}(w_2) - 1 + k(1-p)\hat{w}_1(1-w_2) \tag{A.15}$$

Using Eqs (4.28), (4.29) and (A.14), this leads to Eq. (4.33). Furthermore, Eq. (4.34) follows from the substitution of $\hat{w}_1$ given in Eq. (A.14) into Eq. (A.13).

Finally, we prove that Eq. (4.35) holds. First, we rewrite Eq. (4.33) as

$$n_D = (p + (1-p)(1-\hat{w}_1))^k - 1 + \hat{w}_1 + k(1-p)(1-w_2)\hat{w}_1, \tag{A.16}$$

where $\hat{w}_1$ satisfies

$$\hat{w}_1 = e^{-k(1-p)(p+(1-p)(1-\hat{w}_1))^{k-1}}. \tag{A.17}$$

Therefore, for large k, we obtain

$$\hat{w}_1 \approx e^{-k(1-p)}, \tag{A.18}$$

while from Eq. (A.13), we get

$$1 - w_2 = (p + (1-p)(1-\hat{w}_1))^{k-1} \approx 1 - (1-p)(k-1)\hat{w}_1. \tag{A.19}$$

Then plugging Eqs (A.18) and (A.19) into Eq. (A.16) yields

$$\begin{aligned} n_D &\approx 1 - (1-p)k\hat{w}_1 - 1 + \hat{w}_1 + k(1-p)(1-(1-p)(k-1)\hat{w}_1)\hat{w}_1 \\ &= 1 - (1-p)k\hat{w}_1 - 1 + \hat{w}_1 + k(1-p)\hat{w}_1 - (1-p)^2k(k-1)\hat{w}_1^2 \approx \hat{w}_1 \approx e^{-k(1-p)}. \end{aligned} \tag{A.20}$$

This completes the Proof of Theorem 4.3.

Appendix B

Proof of Theorem 5.1

Let us denote the out-degree distribution for the considered network by $\hat{P}_{\rm out}(\cdot)$. Then, it holds that

$$\hat{P}_{\rm out}(k_{\rm out}) = \alpha\delta(k_{\rm out} - k_1) + (1-\alpha)\delta(k_{\rm out} - k_2). \tag{B.1}$$

Then, denoting the generating function for the out-degree distribution by $\hat{G}_{\rm out}$, we get

$$\begin{aligned} \hat{G}_{\rm out}(x) &= \sum_{k_{\rm out}=0}^{\infty} \hat{P}_{\rm out}(k_{\rm out})x^{k_{\rm out}} \\ &= \sum_{k_{\rm out}=0}^{\infty} (\alpha\delta(k_{\rm out} - k_1) + (1-\alpha)\delta(k_{\rm out} - k_2))x^{k_{\rm out}} = \alpha x^{k_1} + (1-\alpha)x^{k_2}. \end{aligned} \tag{B.2}$$

Let us denote the in-degree distribution for the considered network by $\hat{P}_{\text{in}}(\cdot)$, which for large N will approach a Poisson distribution with average $k = \alpha k_1 + (1-\alpha)k_2$. Then, it holds that

$$\hat{P}_{\text{in}}(k_{\text{in}}) = \frac{k^{k_{\text{in}}}}{k_{\text{in}}!}e^{-k}. \tag{B.3}$$

Then, denoting the generating function for the in-degree distribution by $\hat{G}_{\text{in}}$, we get

$$\begin{aligned}\hat{G}_{\text{in}}(x) &= \sum_{k_{\text{in}}=0}^{\infty} \hat{P}_{\text{in}}(k_{\text{in}})x^{k_{\text{in}}} = \sum_{k_{\text{in}}=0}^{\infty} \frac{k^{k_{\text{in}}}}{k_{\text{in}}!}e^{-k}x^{k_{\text{in}}} \\ &= e^{-k}\sum_{k_{\text{in}}=0}^{\infty}\frac{(kx)^{k_{\text{in}}}}{k_{\text{in}}!} = e^{-k}e^{kx} = e^{-k(1-x)}.\end{aligned} \tag{B.4}$$

This finishes the proof of Theorem 5.1.

Proof of Theorem 5.2

Using the same notation as before, it follows from Eq. (4.3) that the generating function $\hat{H}_{\text{out}}(x)$ is given by

$$\hat{H}_{\text{out}}(x) = \sum_{k_{\text{out}}=1}^{\infty} \frac{k_{\text{out}}\hat{P}_{\text{out}}(k_{\text{out}})}{\langle k_{\text{out}}\rangle}x^{k_{\text{out}}-1}. \tag{B.5}$$

Then, using Eq. (B.1), we obtain

$$\begin{aligned}\hat{H}_{\text{out}}(x) &= \sum_{k_{\text{out}}=1}^{\infty} \frac{k_{\text{out}}(\alpha\delta(k_{\text{out}}-k_1) + (1-\alpha)\delta(k_{\text{out}}-k_2))}{k}x^{k_{\text{out}}-1} \\ &= \frac{\alpha k_1 x^{k_1-1} + (1-\alpha)k_2 x^{k_2-1}}{k}.\end{aligned} \tag{B.6}$$

Finally, we prove Eq. (5.6).

Using the same notation as before, it follows from Eq. (4.4) that the generating function $\hat{H}_{\text{in}}(x)$ is given by

$$\hat{H}_{\text{in}}(x) = \sum_{k_{\text{in}}=1}^{\infty} \frac{k_{\text{in}}\hat{P}_{\text{in}}(k_{\text{in}})}{\langle k_{\text{in}}\rangle}x^{k_{\text{in}}-1}. \tag{B.7}$$

Then, using Eq. (B.3), we obtain

$$\begin{aligned}\bar{H}_{\text{in}}(x) &= \sum_{k_{\text{in}}=1}^{\infty} \frac{k_{\text{in}}k^{k_{\text{in}}}e^{-k}x^{k_{\text{in}}-1}}{kk_{\text{in}}!} = e^{-k}\sum_{k_{\text{in}}=1}^{\infty}\frac{k^{k_{\text{in}}-1}x^{k_{\text{in}}-1}}{(k_{\text{in}}-1)!} \\ &- e^{-k}\sum_{i=0}^{\infty}\frac{(kx)^i}{i!} = e^{-k}e^{kx} = e^{-k(1-x)}.\end{aligned} \tag{B.8}$$

This finishes the Proof of Theorem 5.2.

Proof of Theorem 5.3

Using Theorems 5.1 and 5.2, the set of equations (4.6)–(4.9) becomes

$$w_1 = \frac{\alpha k_1 \hat{w}_2^{k_1-1} + (1-\alpha)k_2 \hat{w}_2{}^{k_2-1}}{k} \tag{B.9}$$

$$\hat{w}_2 = 1 - e^{-kw_1} \tag{B.10}$$

$$w_2 = 1 - \frac{\alpha k_1 (1-\hat{w}_1)^{k_1-1} + (1-\alpha)k_2(1-\hat{w}_1)^{k_2-1}}{k} \tag{B.11}$$

$$\hat{w}_1 = e^{-k(1-w_2)} \tag{B.12}$$

By setting $\hat{w}_2 = 1 - \hat{w}_1$ and $w_1 = 1 - w_2$, it follows that the pair of Eqs (B.9)–(B.10) is equivalent to the pair of equations (B.11) and (B.12). From this, it follows that n_D in Eq. (4.5) becomes

$$n_D = \hat{G}_{\text{out}}(1-\hat{w}_1) + \hat{G}_{\text{in}}(w_2) - 1 + k\hat{w}_1(1-w_2) \tag{B.13}$$

Using Eqs (5.2), (5.3) and (B.12), this leads to Eq. (5.7). Furthermore, Eq. (5.8) follows from the substitution of $\hat{w}_1$ given in Eq. (B.12) into Eq. (B.11).

Finally, we prove that Eq. (5.9) holds. First, we rewrite Eq. (5.7) as

$$n_D = \alpha(1-\hat{w}_1)^{k_1} + (1-\alpha)(1-\hat{w}_1)^{k_2} - 1 + \hat{w}_1 + k(1-w_2)\hat{w}_1, \tag{B.14}$$

where $\hat{w}_1$ satisfies

$$\begin{aligned} \hat{w}_1 &= e^{-(\alpha k_1(1-\hat{w}_1)^{k_1-1}+(1-\alpha)k_2(1-\hat{w}_1)^{k_2-1})} \\ &\approx e^{-(\alpha k_1+(1-\alpha)k_2)+(\alpha k_1(k_1-1)+(1-\alpha)k_2(k_2-1))\hat{w}_1} \\ &= e^k e^{(\alpha k_1(k_1-1)+(1-\alpha)k_2(k_2-1))\hat{w}_1} \end{aligned} \tag{B.15}$$

Therefore, for large k, we obtain

$$\hat{w}_1 \approx e^{-k}, \tag{B.16}$$

while from Eq. (B.11), we get

$$\begin{aligned} w_2 &\approx 1 - \frac{\alpha k_1(1-(k_1-1)\hat{w}_1 + (1-\alpha)k_2(1-(k_2-1)\hat{w}_1}{k} = \\ &1 - \frac{k - (\alpha k_1(k_1-1) + (1-\alpha)k_2(k_2-1))\hat{w}_1}{k} = \\ &\frac{\alpha k_1(k_1-1) + (1-\alpha)k_2(k_2-1)}{k}\hat{w}_1 \equiv \sigma\hat{w}_1. \end{aligned} \tag{B.17}$$

Then, plugging Eqs (B.16) and (B.17) into Eq. (B.14) yields

$$\begin{aligned} n_D &\approx \alpha(1-k_1\hat{w}_1)+(1-\alpha)(1-k_2)\hat{w}_1-1+\hat{w}_1+k(1-\sigma\hat{w}_1)\hat{w}_1 \\ &= \alpha-\alpha k_1\hat{w}_1+1-\alpha-k_2(1-\alpha)\hat{w}_1-1+\hat{w}_1+k\hat{w}_1-k\sigma\hat{w}_1^2 \approx \hat{w}_1 = e^{-k}. \end{aligned} \tag{B.18}$$

This completes the Proof of Theorem 5.3.

Appendix C

Using Theorems 5.4 and 5.5, the set of equations (4.6)–(4.9) becomes

$$\omega_1 = \frac{\alpha k_1(p+(1-p)\hat{\omega}_2)^{k_1-1}+(1-\alpha)k_2(p+(1-p)\hat{\omega}_2)^{k_2-1}}{k} \tag{C.1}$$

$$1-\omega_2 = \frac{\alpha k_1(p+(1-p)(1-\hat{\omega}_1))^{k_1-1}+(1-\alpha)k_2(p+(1-p)(1-\hat{\omega}_1))^{k_2-1}}{k} \tag{C.2}$$

$$\hat{\omega}_1 = e^{-k(1-p)(1-\omega_2)} \tag{C.3}$$

$$1-\hat{\omega}_2 = e^{-k(1-p)\omega_1}. \tag{C.4}$$

By setting $\hat{\omega}_2 = 1-\hat{\omega}_1$ and $\omega_2 = 1-\omega_1$, it follows that the pair of Eqs (C.2) and (C.3) is equivalent to the pair of Eqs (C.1)–(C.4). Then by using Eq. (4.5), we get

$$\begin{aligned} n_D = \alpha(p+(1-p)(1-e^{-k(1-\omega_2)}))^{k_1}+(1-\alpha)(p+(1-p)(1-e^{-k(1-\omega_2)}))^{k_2} \\ -1+e^{-k(1-p)(1-\omega_2)}+k(1-p)e^{-k(1-\omega_2)}(1-\omega_2) \end{aligned} \tag{C.5}$$

where w_2 is the solution of Eqs (C.2) and (C.3). This proves that Eq. (5.17) holds.

Finally, we prove that Eq. (5.19) holds. From Eqs (C.2) and (C.3), it follows that

$$\begin{aligned} \hat{\omega}_1 &= e^{-(1-p)(\alpha k_1(p+(1-p)(1-\hat{\omega}_1))^{k_1-1}+(1-\alpha)k_2(p+(1-p)(1-\hat{\omega}_1))^{k_2-1})} \\ &\approx e^{-k(1-p)}e^{(1-p)^2(\alpha k_1(k_1-1)+(1-\alpha)k_2(k_2-1))\hat{w}_1}. \end{aligned} \tag{C.6}$$

Therefore, for large k, we obtain

$$\hat{w}_1 \approx e^{-k(1-p)}, \tag{C.7}$$

Similarly, from Eq. (C.2), we can deduce

$$w2 \approx \frac{(1-p)(\alpha k_1(k_1-1)+(1-\alpha)k_2(k_2-1))}{k}\hat{w}_1 \equiv \sigma\hat{w}_1 \tag{C.8}$$

Substitution of Eqs (C.7) and (C.8) into Eq. (C.5), we obtain

$$n_D \approx e^{-\bar{k}(1-p)} \tag{C.9}$$

This completes the Proof of Theorem 5.6.

Funding

The first author was sponsored by the China Scholarship Council (grant No. 201706220113).

References

1. Jia, T., Liu, Y.-Y., Csóka, E., Pósfai, M., Slotine, J.-J. & Barabási, A.-L. (2013) Emergence of bimodality in controlling complex networks. *Nat. Commun.*, **4**, 2002.
2. Liu, Y.-Y., Slotine, J.-J. & Barabási, A.-L. (2011) Controllability of complex networks. *Nature*, **473**, 167.
3. Nepusz, T. & Vicsek, T. (2012) Controlling edge dynamics in complex networks. *Nat. Phys.*, **8**, 568.
4. Yuan, Z., Zhao, C., Di, Z., Wang, W.-X. & Lai, Y.-C. (2013) Exact controllability of complex networks. *Nat. Commun.*, **4**, 2447.
5. Lombardi, A. & Hörnquist, M. (2007) Controllability analysis of networks. *Phys. Rev. E*, **75**, 056110.
6. Lin, C.-T. (1974) Structural controllability. *IEEE Trans. Autom. Control*, **19**, 201–208.
7. Cowan, N. J., Chastain, E. J., Vilhena, D. A., Freudenberg, J. S. & Bergstrom, C. T. (2012) Nodal dynamics, not degree distributions, determine the structural controllability of complex networks. *PLoS One*, **7**, 1–5.
8. Ruths, J. & Ruths, D. (2014) Control profiles of complex networks. *Science*, **343**, 1373–1376.
9. Yan, G., Tsekenis, G., Barzel, B., Slotine, J.-J., Liu, Y.-Y. & Barabási, A.-L. (2015) Spectrum of controlling and observing complex networks. *Nat. Phys.*, **11**, 779.
10. Zhang, R., Wang, X., Cheng, M. & Jia, T. (2019) The evolution of network controllability in growing networks. *Physica A*, **520**, 257–266.
11. Lu, Z.-M. & Li, X.-F. (2016) Attack vulnerability of network controllability. *PLoS One*, **11**, e0162289.
12. Lou, Y., Wang, L., Tsang, K.-F. & Chen, G. (2020) Towards optimal robustness of network controllability: an empirical necessary condition. *IEEE Trans. Circuits Syst. I*, **67**, 3163–3174.
13. Wang, S., Yang, Y., Sun, L., Li, X., Li, Y. & Guo, K. (2020) Controllability robustness against cascading failure for complex logistic network based on dynamic cascading failure model. *IEEE Access*, **8**, 127450–127461.
14. Nie, S., Wang, X., Zhang, H., Li, Q. & Wang, B. (2014) Robustness of controllability for networks based on edge-attack. *PLoS One*, **9**, e89066.
15. Sun, P., Kooij, R. E., He, Z. & Van Mieghem, P. (2019) Quantifying the robustness of network controllability. In: *2019 4th International Conference on System Reliability and Safety (ICSRS)*. IEEE, pp. 66–76.
16. Komareji, M. & Bouffanais, R. (2013) Resilience and controllability of dynamic collective behaviors. *PLoS One*, **8**, 1–15.
17. Bouffanais, R. (2016) *Design and Control of Swarm Dynamics*. Singapore: Springer Singapore.
18. Komareji, M. & Bouffanais, R. (2013) Controllability of a swarm of topologically interacting autonomous agents. *Int. J. Complex Systems in Science*, **3**, 11–19. abs/1401.4777.
19. Sekunda, A., Komareji, M. & Bouffanais, R. (2016) Interplay between signaling network design and swarm dynamics. *Netw. Sci.*, **4**, 244–265.
20. Kit, J. L., Dharmawan, A. G., Mateo, D., Foong, S., Soh, G. S., Bouffanais, R. & Wood, K. L. (2019) Decentralized multi-floor exploration by a swarm of miniature robots teaming with wall-climbing units. In: *2019 IEEE International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, pp. 195–201.
21. Mateo, D., Horsevad, N., Hassani, V., Chamanbaz, M. & Bouffanais, R. (2019) Optimal network topology for responsive collective behavior. *Sci. Adv.*, **5**, eaau0999.
22. Horsevad, N., Mateo, D., Kooij, R. E., Barrat, A. & Bouffanais, R. (2022) Transition from simple to complex contagion in collective decision-making. *Nat. Commun.*, **13**, 1–10.
23. Kalman, R. E. (1963) Mathematical description of linear dynamical systems. *J. Soc. Ind. Appl. Math. A*, **1**, 152–192.
24. Yang, Y. & Xie, G. (2016) Mining maximum matchings of controllability of directed networks based on in-degree priority. In: *2016 35th Chinese Control Conference (CCC)*. IEEE, pp. 1263–1267.

25. Hopcroft, J. E. & Karp, R. M. (1973) An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, **2**, 225–231.
26. Newman, M. E. J. (2003) The structure and function of complex networks. *SIAM Rev.*, **45**, 167–256.
27. Newman, M. E. J. (2010) *Networks: An Introduction*. Oxford: Oxford University Press.
28. Van Mieghem, P. (2014) *Performance Analysis of Complex Networks and Systems*. Cambridge: Cambridge University Press.