


ITC 1/49 Information Technology and Control Vol. 49 / No. 1 / 2020 pp. 100-112 DOI/10.5755/j01.itc.49.1.23612	A Method for Finding Constrained Driver Nodes in Target Control of Network	
	Received 2019/03/01	Accepted after revision 2019/11/19
	 http://dx.doi.org//10.5755/j01.itc.49.1.23612	

HOW TO CITE: Qian, Y., Qiu, Z., Lu, F. (2020). A Method for Finding Constrained Driver Nodes in Target Control of Network. *Information Technology and Control*, 49(1), 100-112. <https://doi.org//10.5755/j01.itc.49.1.23612>

A Method for Finding Constrained Driver Nodes in Target Control of Network

Yuhua Qian, Zhipeng Qiu and Furong Lu

School of Computer and Information Technology; Shanxi University; Wucheng Road 92, Taiyuan, China;
Institute of Big Data Science and Industry, Shanxi University, Wucheng Road 92, Taiyuan, China;
e-mail: jinchengqyh@126.com

Corresponding author: jinchengqyh@126.com

One of the ultimate goals of studying network dynamics and its properties is to control it. In the past 20 years, with the joint efforts of many scientists, the theory of controlling a whole network through a small set of nodes, which are called driver nodes (DN) has been greatly developed. However, in some situations, it is very difficult and not necessary to control a whole network, which motivates scientists to explore target control theory, i.e., the efficient control for a subset of nodes in a network through a small node set. In a real network, there is another common situation, which is that not each node can be easily accessed. Therefore, it is meaningful to explore the target control strategy under the condition in which the driver nodes are constrained. In this paper, an effective method is proposed to make more DN be included in the constrained node set (CNs). We adopt the strategy of greedy algorithm to gradually constrain DN into CNs in the iterative process of target control. In each iteration, we adjust the strategy of how to choose driver nodes according to some network properties. A few experiments were presented to prove that the proposed method can make DN into CNs effectively in both Scale-free networks (SF) and Erdős-Rényi (ER) networks. Then, we explored the performance of this method in the case of local and random selection of target nodes, respectively. In addition, some factors that will affect the effects of this method were also explored in this paper. In the end, this method is proved valid through the verification of the real network data sets.

KEYWORDS: complex network; constrained nodes; target control.

1. Introduction

1.1. Network Science Introduction in Recent Years

In the past few decades, network science has been widely applied in various fields: human brain intelligence [17, 23], economy [9, 21], society [15, 21], physics [3, 26], biology [27, 29], etc. Through nearly 20 years of unremitting efforts, great progress has been made in the understanding of network topology and its dynamical properties. In recent years, link prediction [14, 25], community discovery [1, 22], structure and dynamics of network [4, 13] have received extensive attention. These research achievements create the knowledge foundation based on which the controllability issues of network can be investigated effectively. Among these issues, topological [30] and dynamical [6] properties of network are the two most important aspects, which can determine the number of the external inputs and the energy of control, etc. However, the dynamics of real-world networks are so complex that we can not capture them easily. Thus, in this paper, we explore the control process in a complex network with linear dynamics.

1.2. Literature Review on Network Controllability

In the field of network controllability, controlling one complex network means steering the state variables associated with its nodes to arbitrary states of them using some suitable control inputs in finite time. In 2011, Liu *et al.* [16] developed an analytical framework to solve the issue of complex networks' controllability, which is based on Lin's classical structure control theory (*SCT*) [2]. In that paper, Liu *et al.* answered the following question: how to find the minimal number of *DN*. Subsequently, Yuan *et al.* [28] proposed a method called exact control of complex network that can calculate the minimum number of *DN* in an undirected network based on *PBH* rank criterion [10]. The target control theory [7] raised by Gao *et al.* is applicable to a situation that the complex network does not need to be fully controlled, i.e., only a part nodes of this network need to be controlled.

In recent years, some scientists have done some works in nonlinear control and closed loop control. Sun *et al.*

[24] developed a framework to solve the problem of finite-time, closed-loop control. They also explored the mathematics and physical foundations of the trade-off between the control time and energies. Kim *et al.* [12] revealed the relationship between the connectivity of a subset of structural connections and the minimum energy required to control the system through linear network control theory. A method called the minimum-cost fixed-flow [20] was proposed by Shana *et al.* to solve the minimum cost constrained input selection for controlling state-space structured systems. Ning *et al.* [5] explored the controllability of weighted complex networks with noises in the sense of Kalman controllability. Tommaso *et al.* [18] raised a theory for structural controllability of networks with symmetric, constrained weights. The problem of how to choose the minimum cost constrained inputs, outputs and feedback pattern structured systems [19] was solved by Shana *et al.* through an order optimal approximation algorithm. These works deal with many problems in the area of linear controllability of complex network.

1.3. Motivation of Our Work

The theoretical framework of structural control and target control provide solutions for fully controlling network and partially controlling network, respectively. For a common complex network, target control strategy uses a greedy algorithm to find a minimum *DN* based on the K-walk theory. However, in a networked system of the real world, many social networks, biological networks, and technological networks have various constraints, so control signals cannot be applied at any node in the networks like these. For example, in a genetic network, we cannot control the gene which we need to change from any part of it, and we also cannot build epidemic prevention station anywhere in an infectious disease network. Therefore, it is of great practical significance to propose a method of converging the *DN* into a *CN*.

However, there are very few works related to the issue that the *DN* are constrained. Guo *et al.* [8] has raised a framework to solve this problem based on the nodes which point to the target nodes. They solved this problem based on integer linear programming (ILP)

and random Markov chain (MC) sampling. Guo's method has two main problems, presented as follows:

- 1 The ILP they need to solve is an NP-hard problem. If the size of a network is relatively large, it will be very difficult to obtain the solution of the ILP in equation.
- 2 They get some possible maximum matchings for the target nodes of a network through random Markov chain sampling. The sampling results will directly affect the selection of the final driver nodes. Hence, the method will bring greater randomness, especially for a large size network. That is because for a network with large size, random MC sampling can obtain only a small part of all the different maximum matchings.

Compared with Guo's method, our method can be applied to a large size network easily. However, for that our method does not need to sample maximum matchings, it shows great robustness in both large and small size networks. The comparison of our method and Guo's method in time complexity are presented as follows:

Table 1

Comparison of time complexity

Name	Guo's Method	DC Method
Time complexity	$\alpha(m*r*\sqrt{\ V\ *}\ E\)$	$\alpha(r*\sqrt{\ V\ *}\ E\)$

where m represents the times of MC sampling, r , $\|V\|$, $\|E\|$ represents the iteration times in target control process, the number of nodes and the number of edges, separately. For our method, we only need sort the matching sequence before the first iteration (more details will be shown in section 4), and then in each iteration, we only need to make some small changes. Hence, the main time complexity comes from the target control process, i.e. $\alpha(r*\sqrt{\|V\|*}\|E\|)$. From this table, we can see that our method has a great advantage in time complexity.

The paper is organized as follows. First, we introduce some concepts and models of network control. Second, we propose a method for finding DN constrained under target control of network. Third, some factors which may influence the effects of this method are presented in different target nodes selected strategies. Fourth, we verified the effectiveness of the meth-

od on some real network data sets. In the end, we conclude the applicable scenarios of the method and find some future research points.

2. Structural Control Theory

2.1. Linear Model

In this section, we will describe the linear model, i.e., linear time invariant complex system (LTI):

$$\dot{x} = Ax + Bu, \quad (1)$$

where \dot{x} means the rate of change for x ; $A \in R^{N \times N}$ describes the system's wiring diagram among nodes in a network; In this equation, a_{ij} which means the element of i th row intersecting with j th column in A , which represents the strength of influence that the component x_j put on x_i ($a_{ij}=0$ means have no influence); $x = (x_1, x_2, \dots, x_N)^T$ characterizes the state of a system of N nodes; $B \in R^{N \times M}$ denotes the input matrix of control signals, and u represents M dimensions input signals.

2.2. Control of Complex Network

The controllability of a network can be defined as that for any initial state $x(t_0)$, the state of the system can reach any state in time t_1 through some appropriate inputs. Here, the mathematic condition is given as follows: the system is controllable if and only if the $N \times NM$ controllability matrix:

$$C = (B, AB, A^2B, \dots, A^{N-1}B) \quad (2)$$

is full rank, that is to say

$$rank(C) = N, \quad (3)$$

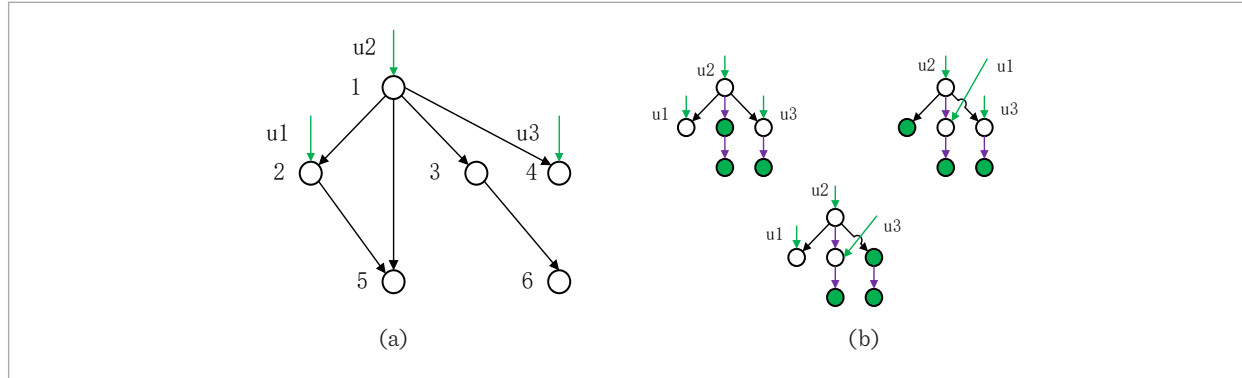
which is called Kalman's Controllability Rank Condition [11]. Meanwhile, there is another rank condition: if the matrix

$$[\lambda I_N - A, B] \quad (4)$$

is full rank with λ can be any value, the network can be controlled fully. This is named PBH Rank Condition [10]. Where I_N is an N -dim identity matrix; And other symbols have the same meanings as the above.

Figure 1

Structure Control Graphs. (a) The small network can be controlled by an input vector which can be captured by $u(t) = (u_1(t), u_2(t), u_3(t))$, allowing us to change it from its initial state to some desired final state in the state space. (b) Maximum matching of the small network. Matching edges are shown in purple, matched nodes are green and unmatched nodes are white. The maximum matching includes all links, none of which shares a common starting or ending node. Only three links can be part of a maximum matching for the network, yielding three unmatched nodes (The number of ND is 3). There are totally three different maximum matchings for this network



In the area of controlling complex network, the most important task is to find m driver nodes, which directly determine the control energy. By applying appropriate input signals to the driver nodes, the entire network will reach the target state in the case of linear dynamics in a finite time interval. The control process is presented as (Figure 1(a)).

Since the number of driver nodes in a network has great influence on control energy, how to determine the minimum set of DN is a critical thing. The algorithm of maximum matching in graph theory is introduced to solve this problem by Liu *et al.*, and a set of driver nodes can be obtained through an iterative process. They proposed a framework in light of the fact that if a network is controllable, the driver nodes is the unmatched nodes in the maximum matching of it (the white nodes in Figure 1(b)). The maximum matching represents that all the edges in the network that have no common starting nodes and ending nodes. Here, we call the source node of a matching edge “matching node” (white nodes in Figure 1(b)), and the target node “matched node” (corresponding to the green nodes in Figure 1(b)). In a network, the number of the maximum matching edges is definite, however, the matching edges are not constant, i.e., we can obtain at least one minimum driver nodes set.

3. Target Control Theory

Target control, which can be considered as a special case of full control, is a very practical control style in the real world. Here, our study is based on the target control of *LTI* dynamical system raised by Gao *et al* [7]. This control model is presented as follows:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}, \quad (5)$$

where C represents the output matrix, which is used to identify the target nodes we want to control, and y means the state of them. The definitions of other symbols are consistent with Equation (1).

For a network, $Net = \{v_1, v_2, \dots, v_N\}$ represents all the nodes of it, and a set of target nodes $C_n = \{c_1, c_2, \dots, c_n\}$ with size $S = |C_n| = fN$ is what we want to control. In the output matrix $C = [I(c_1), I(c_2), \dots, I(c_s)]$, $I(i)$ denotes the i th row of an $N \times N$ identity matrix. The system (A, B, C) is target controllable when the set C can be driven to any desired final state in finite time by a time-dependent input vector $u(t) = (u_1(t), u_2(t), \dots, u_M(t))^T$. Hence, the system (A, B, C) is target controllable when the following condition is satisfied:

$$d(A, B, C) \equiv \text{rank}[CB, CAB, CA^2B, \dots, CA^{N-1}B] = S. \quad (6)$$

If a network meets the above conditions, given any initial state $x(0)$, the input signal values $u(t)$ that change over time can be calculated, but how to compute them is beyond the scope of this article.

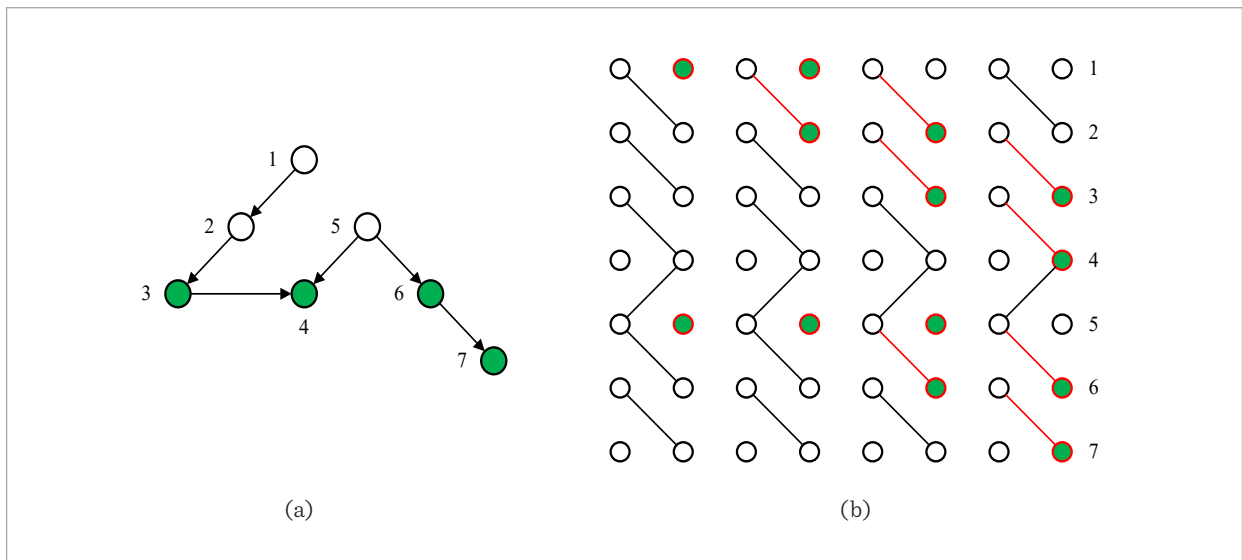
In order to find the minimum set of driver nodes under the premise of target control, Gao *et al.* used a greedy algorithm to approximate the exact set. In their paper, the “K-walk” theory is introduced to solve this question. This main idea of this theory is that a node can control a set of target nodes with the distances from the driver node to each target node are different. Based on this, a set of controllable nodes with respect to a given driver node can be identified. However, in general, a network requires more than one driver node, so is the greedy strategy applied in this case. As shown in Figure 2, in the process of identifying driver nodes, the nodes which point to the given target node set is first identified. If a node has no

in-bound edges or the edges pointing to it have not been included in maximum matching, then the node itself should be put into the final driver nodes set. Those matching nodes become a new round of target nodes which should be matched in the next iteration. Through the iteration process, the minimum driver nodes set could be found in finite time, until all the matching nodes are not changed any more.

However, under this strategy, the result is an approximation of the set to minimum driver nodes for a network. Because although the minimum number of matching nodes can be obtained in each iteration, the choice of matching nodes in the previous iteration will affect that in this time. If we want to solve this problem thoroughly, we should compute all kinds of matching strategies in each iteration, which is very complex in time and space. To solve this problem, Zhang *et al.* proposed one method to find less driver nodes by changing the iteration strategy [31].

Figure 2

Target control process. (a) In the small network ($N = 7$), $\{x_3, x_4, x_6, x_7\}$ are target nodes (represented by green nodes). (b) Use greedy algorithm to solve the target control problem of this network. First, the network is depicted by a bipartite graph. Target nodes are $\{x_3, x_4, x_6, x_7\}$. Through the first iteration (red lines), we find the matching nodes $\{x_2, x_3, x_5, x_6\}$. Next, these nodes are regarded as target nodes in the second iteration. Repeat the above process, we will get the second round matching nodes $\{x_1, x_2, x_5\}$ and an unmatched node x_5 , i.e., a driver node; In the last iteration, we find another unmatched node x_1 . Hence, the minimum set of driver nodes $\{x_1, x_5\}$ is obtained



4. A Method for Finding Driver Nodes Constrained under Target Control

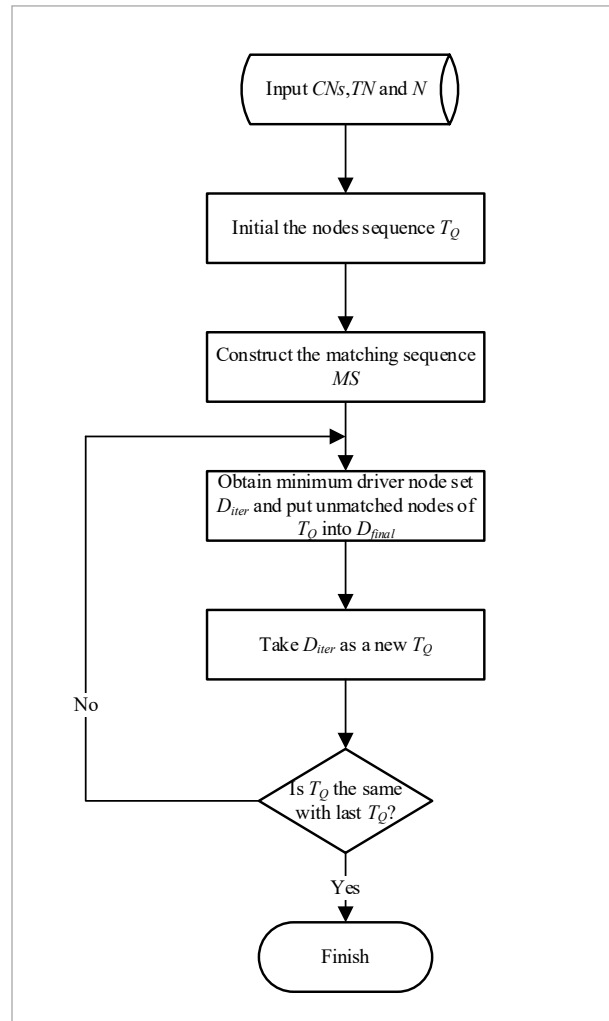
In many real networks, the locations of the DN are often constrained by some factors. How to control a network as efficiently as possible is an important issue when the driver nodes' positions are constrained. Here, we proposed an algorithm to solve the problem in this situation under the premise of target control, which is named DC . The main ideas of the algorithm are summarized as follows:

- 1 Because the matching nodes produced in each iteration will affect the final result, how to make them into the constrained nodes set (CNs) is the key to the problem. In one iteration, the unmatched nodes of target nodes in this round will be the members of the final driver nodes set, and the matching nodes of this round will be the target nodes of next iteration. Hence, if we can constrain more matching nodes of each round into CNs , the rate of driver nodes in CNs will be higher. In this step, we will put the CNs in the front part of a matching sequence (MS), and the other nodes of the network should be put behind them in MS . Through this step, the CNs can be selected to be the matching nodes preferentially.
- 2 For that we prefer to make DN into the CNs , we sort the target nodes by placing the intersection of target nodes (TN) and the CNs in the latter part of TN . Through this way, even if the nodes in the latter part of TN become the final driver nodes because they are not matched, they have higher probabilities in CNs .
- 3 Under the premise of guaranteeing the first condition, the nodes of CNs and the other nodes in MS are sorted according to the descending order of node in-degree values, separately. Because if the in-degree of a node is larger, the more matching edges it can select, the easier it is to form a matching. Therefore, the number of DN will be reduced as many as possible.
- 4 In one iteration, if a node has become a driver node, the matching priority of this node should be increased, for that, if this node is selected as the driver node of any other node again, it will not increase the number of driver nodes. According to this, we

can change the MS in each iteration to make this algorithm more effective.

Here, according to the above thoughts, we designed our DC algorithm. The flow chart is presented in Figure 3.

Figure 3
Flow diagram of DC method



The algorithm is presented here:

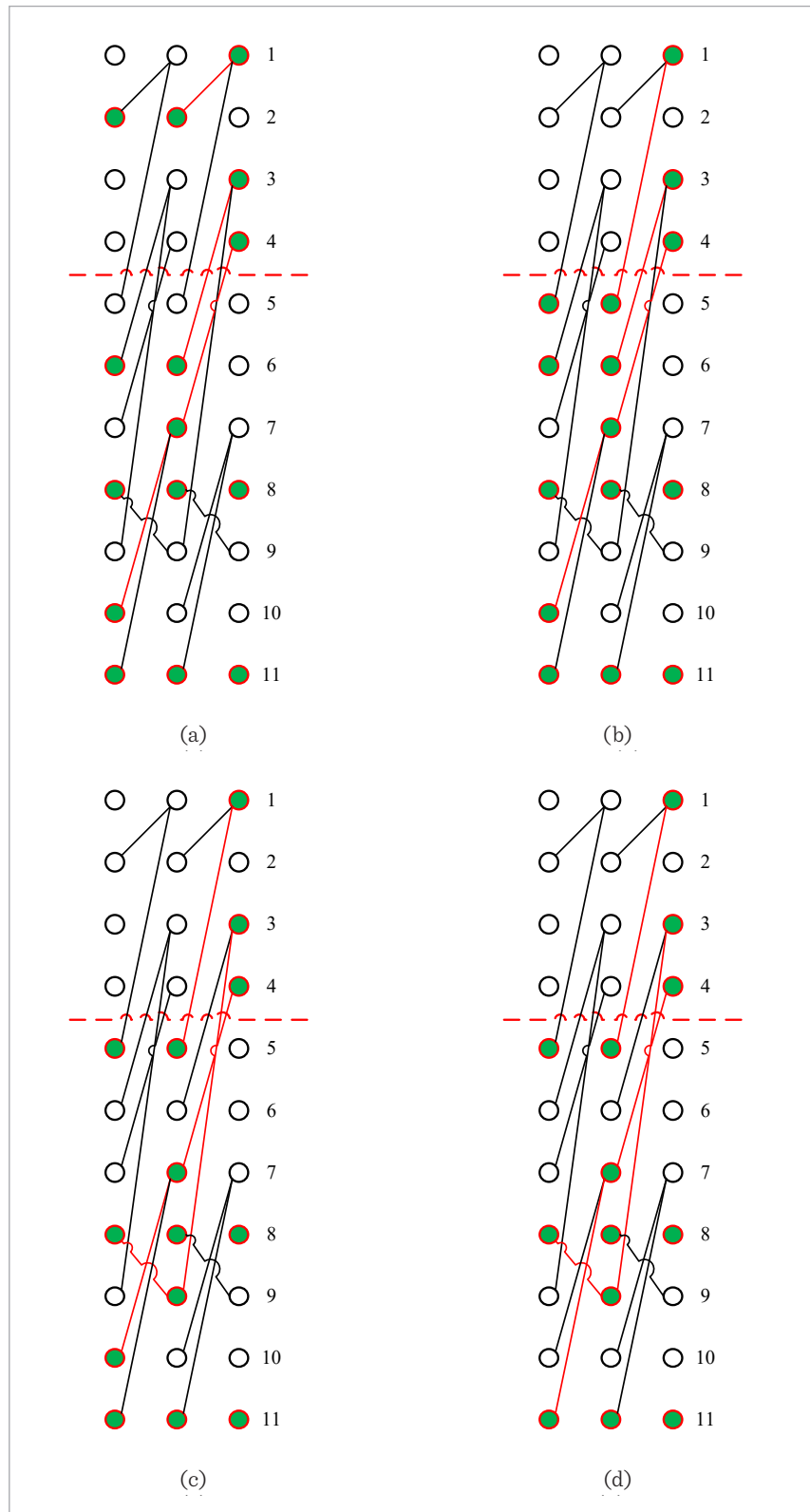
Step 1. The set of constrained nodes, target nodes and total nodes in a network are represented by the symbols CNs , TN and N .

Step 2. Create a nodes sequence T_Q of the set TN . Put the nodes in the sequence T_Q , which belong to $TN \cap CNs$, into the latter part of T_Q .

Figure 4

Constrained driver method.
The nodes below the red line
(Constrain line) indicate a pre-
specified constrained set.

(a) Common target control
method. (b) Prefer to the
nodes which are in *CNs*. In this
subgraph, it refers to node x_5 .
(c) Prefer to nodes with large
in-degree. The matching node of
node x_3 changes from node x_6 to
node x_9 . (d) Prefer to the nodes
those have become the driver
nodes. Node x_7 selects node x_{11}
which has been a driver node as
the matching node



Step 3. Initialize the matching sequence (*MS*) by placing the nodes of N randomly. Get a new sequence *MS* by placing *CNs* in the former part of *MS*. Sort the two parts of the queue *MS* (constrained part and the other) according to the in-degree of nodes from high to low separately.

Step 4. In one iteration, the minimum driver node set D_{iter} of T_Q is found by the maximum matching method according to the matching order *MS*. Next, the unmatched nodes of T_Q in *MS* are changed position to the former of *MS* and put into the final driver node set D_{final} .

Step 5. Take D_{iter} as a new T_Q , and repeat step 4, until T_Q do not change in the two adjacent iterations. Finally, D_{final} is the approximate minimum driver node set.

Taking Figure 4 as an example, the picture shows the results after considering the 3 factors described above. Because the effect of step 2 is obvious, we will not explain too much here. The *TN* in the figure are $T = \{x_1, x_3, x_4, x_8, x_{11}\}$. The nodes below the red line are *CNs*, that is, $C = \{x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}\}$. Figure 3(a) shows that the driver node set is $D = \{x_2, x_6, x_8, x_{10}, x_{11}\}$ without taking any strategy. After using step 1 above, the driver nodes become set $D = \{x_5, x_6, x_8, x_{10}, x_{11}\}$, as shown in Figure 4(b). It can be seen that *DN* are in *CNs*. After adopting the third strategy, the result is shown in Figure 4(c) as a collection $D = \{x_5, x_8, x_{10}, x_{11}\}$. When taking step 4, the final result is a set of nodes $D = \{x_5, x_8, x_{11}\}$. By step 3 and 4, we can see that the number of the driver nodes decreases obviously. Therefore, from theory aspect, this method is effective for solving this problem.

5. Results and Analysis

5.1. N_{con} Index of DC method

In this section, we will give some results and analysis of our *DC* method. In common cases, the *DN* can not totally be the nodes in *CNs*, so we define the following index to evaluate the performance of this algorithm:

$$N_{con} = \frac{N_{dc}}{N_D \times \left(\frac{C}{N}\right)}, \quad (7)$$

where N_{dc} denotes the number of driver nodes in the constrained set. N_D represents that in the whole network. C means the number of the constrained nodes and N means the number of the total nodes. Obviously, the larger the value, the better the performance of this method.

Because the choice of target nodes has an influence on the number of driver nodes, for the selection of target nodes in the experiment, we take the following two ways:

- 1 Random choice strategy: random choose target nodes until the rate of the target nodes reaches the appropriate rate;
- 2 Local choice strategy: choose the nodes which have higher degrees as the target nodes, then use the Breadth-First Search method to find all the target nodes until the rate of target nodes reaches the appropriate rate.

5.2. N_{con} Over Target Nodes Selection Ratio

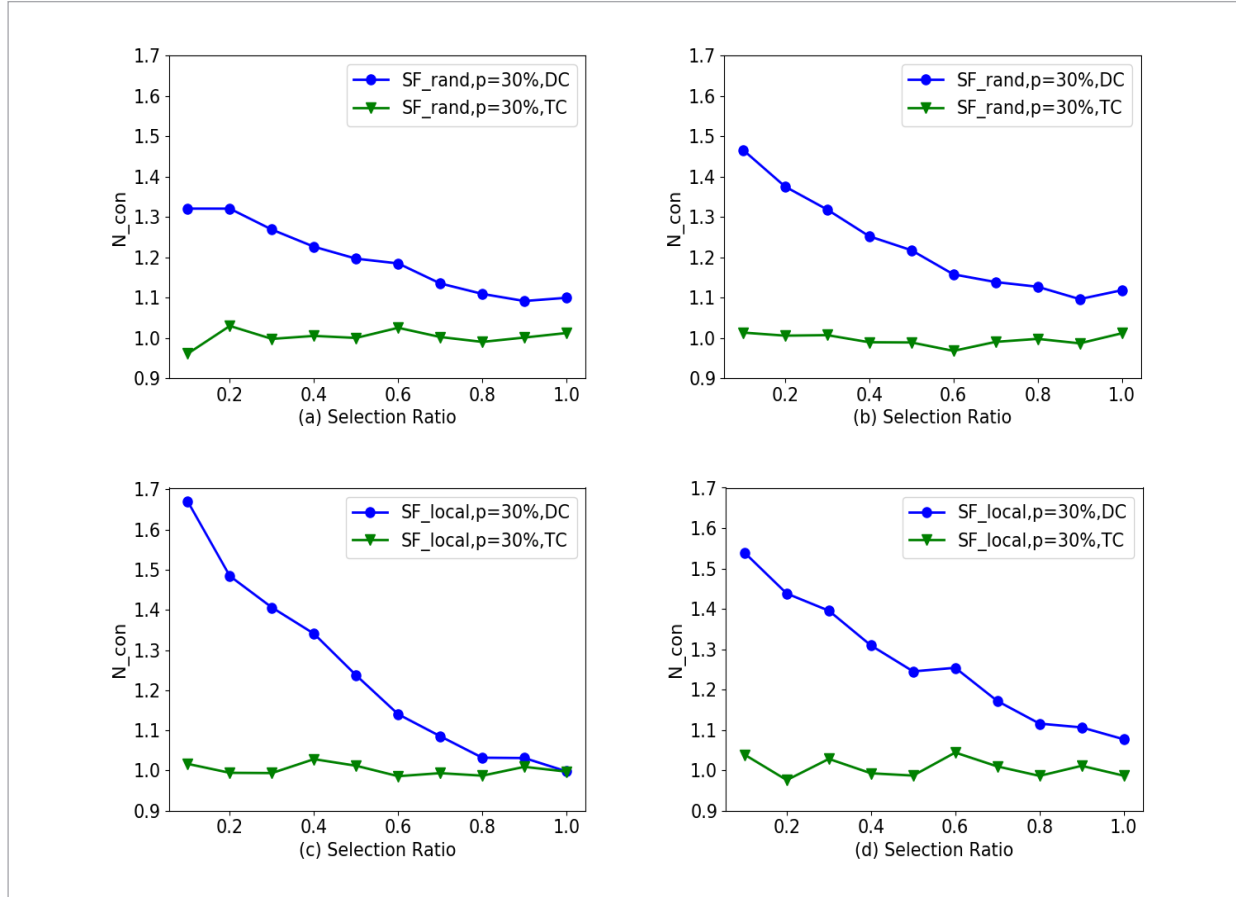
Here, we verified our method on two classical simulation networks, i.e., *SF* and *ER*. Figure 4 shows the results of them as functions of the target nodes selection ratio under different average degree conditions. It can be seen that the strategy has a good effect as expected in these 4 cases. From the overview of these 4 subplots, N_{con} of this method is gradually decreased with the selection ratio increasing. This phenomenon can be explained theoretically, because with the increase of the selection ratio, target control will degenerate into complete structural control gradually. The pre-given *CNs* in the entire network will be vied by the gradually increasing target nodes. From the comparison of Figure 5(a) and Figure 5(c) (or Figure 5(b) and Figure 5(d)), we can see that the result of the local strategy is better than that of the random strategy. Because when the degree of a target node is larger, it will have more choices when selecting the driver nodes.

5.3. N_{con} Over Average Degree of Network

From Figure 5, we can initially see that the average degree of network has a certain impact on the method. Hence, the next step is to explore the influence of network average degree. Figure 6 shows the relationship between them. From the overall picture of the four images, the growth in average degree will indeed bring about an improvement in the control effect N_{con} . In Figure 6(a) and 6(b), the increase in *ER* network is

Figure 5

Algorithm efficiency of different scale target nodes. The metric of the driver node N_{con} is a function of the target nodes selection ratio in the graph. The blue and green lines represent our *DC* method and common target control method, respectively. (a) taking the random strategy under the average degree of 5, (c) taking the local strategy under the average degree of 5. And (b) and (d) show the cases with an average of 7 degree. The size of *SF* networks is $N = 3000$. All the simulation results are obtained by averaging over 20 independent networks realizations



slightly higher than that in *SF* network. However, in Figure 6(b), the curve is nearly no rising with the average degree increasing. When the average degree of the *SF* network increases, a small part of nodes with high degrees will have more chances of driver nodes selection, and most nodes have no obvious changes. Figure 6(c) and 6(d) also show the same phenomenon. However, in *ER* networks, the increase in the average degree of a network leads to higher returns for that it leads to an increase in the chance of selection for each node.

5.4. Network Designation Suggestions

At the beginning of the design of a network, we can reserve a part of the nodes as possible driver nodes to

control the target nodes as efficient as possible. Thus, how to choose this part of the nodes is a very important issue. It can be seen that the nodes with larger degrees have higher possibility to control more nodes from the process of selecting the driver nodes of target control. Therefore, we use the nodes with larger degrees as the pre-designated *CNs*. Because this instance shows the difference in the effects of the two strategies (one is *CNs* with higher degrees, the other is *CNs* chosen randomly), the size of the networks is only $N = 1000$. As shown in Figure 7, under either local or random strategy, nodes with higher degrees of selection will have better results. As the network average degree increases, the pre-selected constrained

Figure 6

Driver efficiency under different network average degrees. This figure captures the functional relationship between the N_{con} and the network average (k). The blue and green lines represent our method and target control method, respectively. The selection ratio of target nodes in the networks is 20%, the constrained node of the network is $C_p = 30\%$, and the scale of the network is $N = 3000$. All the simulation results are obtained by averaging over 20 independent networks realizations. (a) ER network, random strategy; (b) SF network, random strategy; (c) SF network, local strategy; (d) ER network local strategy

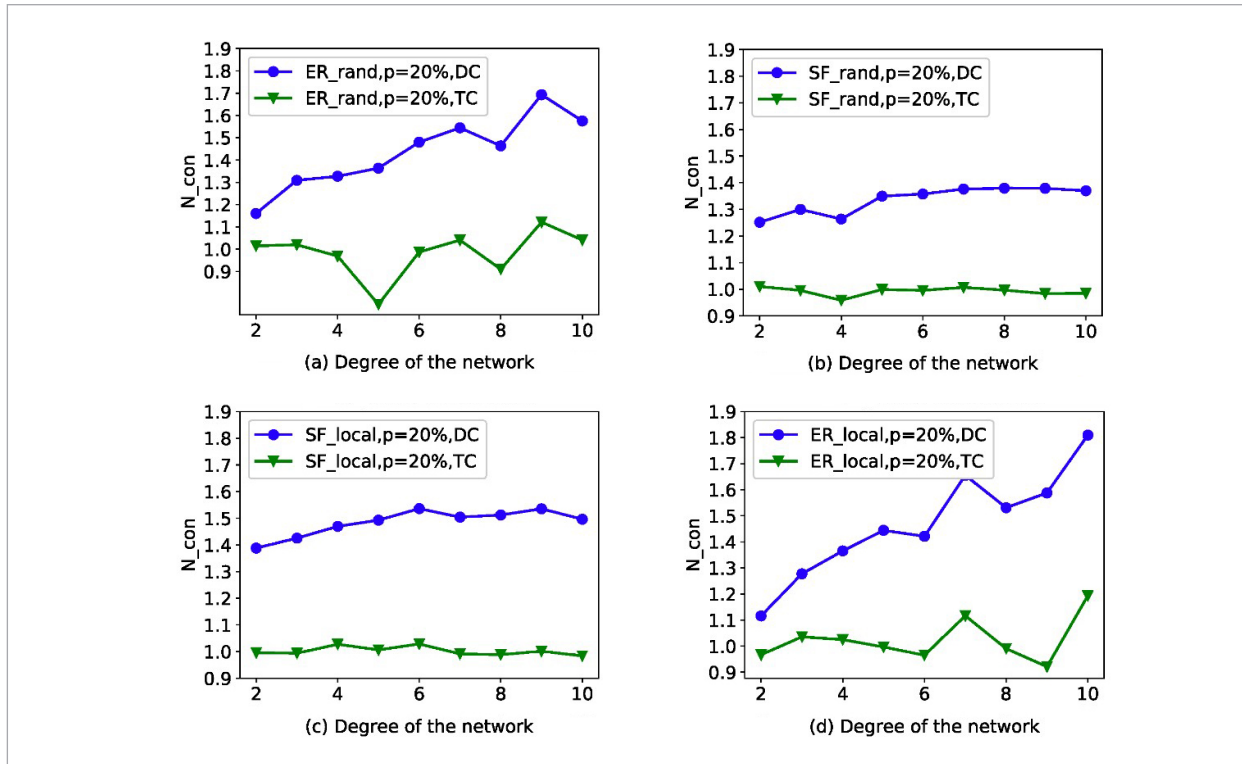
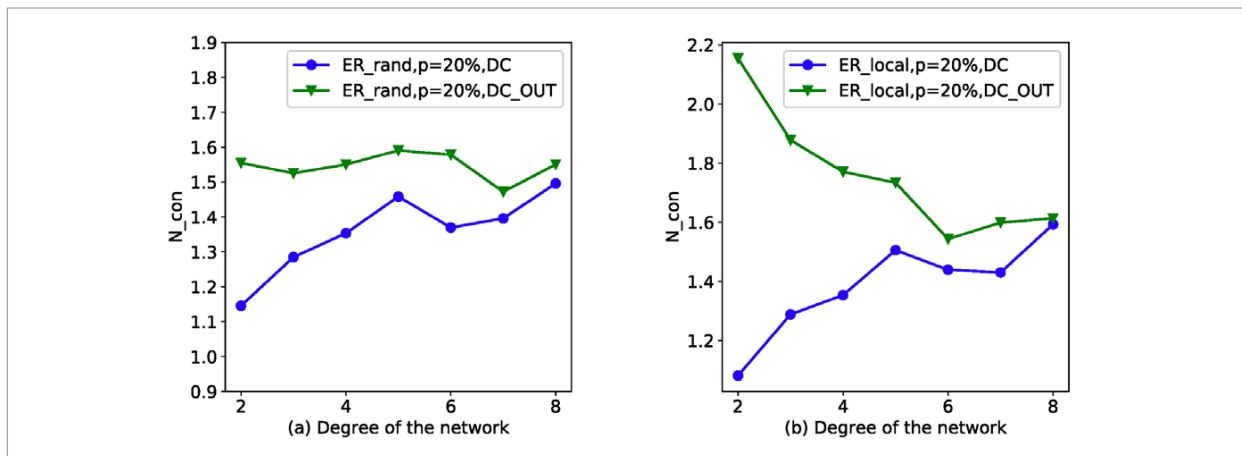


Figure 7

Algorithm efficiency under constrained nodes with higher out-degrees. X axis is the average of networks, and Y axis is N_{con} index. The size of the network becomes $N = 1000$. The percent of target nodes is 20%. The green lines represent the result of constrained nodes with higher out-degree, and the blue lines represent that with random out-degree. (a) Random strategy; (b) Local strategy



nodes with larger degrees will no longer have advantages, and the two curves will gradually approach. This can be clearly observed from the picture.

6. Conclusion

In real networks, the input signals may not be applied to some nodes for many different reasons. This paper explored a target control framework which is used in complex network under the condition that the DN are constrained. We raise this method through changing the matching strategy in the iteration process. Our method is proved effective in the experiments on SF and ER networks. We also verified our method on real data sets. Through this method, we solve the target control under constrained driver nodes situation prob-

lems in some degree. This can be used to constrain the driver nodes in real networks to save costs. Or through this method, we can control some networks which we consider are not target controllable originally. In addition, we also explored the factors that have influence on our DC method. We found that the rate of target nodes selection and average degree of network have great influence on the effect of our DC method.

However, the problem still have many points to study, for example: how to make more driver nodes into the constrained nodes set; how to change the structure of a complex network to make the driver nodes all in CNs ; the factors that influence the effectiveness of constrain DN to the nodes set and the most important factor of these kind of problems, etc. These questions can bring more and more deep insights to the control mechanism of complex networks.

Table 2

Real data sets results. N means the number of nodes in a network; L means that of edges; $\langle k \rangle$ represents the average degree; N_{TCL} , N_{DCL} , N_{TCR} , N_{DCR} represent N_{con} in the following 4 situations: common target control method under local strategy, DC method under local strategy, common target control method under random strategy, DC method under local strategy. The selection ratio of the target nodes in the networks is 20%, the rate of constrained nodes of the network is $C_p = 30\%$. Each data item is the average value of 10 times repeated experiments

Dataset	N	L	$\langle k \rangle$	N_{TCL}	N_{DCL}	N_{TCR}	N_{DCR}
s208	122	189	3.10	1.14683	1.65079	1.02579	1.28889
s420	252	399	3.17	0.964866	1.46818	0.964103	1.16429
s838	512	819	3.20	0.922009	1.60465	0.833248	1.23499
Grasslands	88	137	3.11	0.97535	1.37458	0.986148	1.52381
Littlerock	183	2494	27.26	1.05007	1.27903	1.02967	1.22355
Seagrass	49	226	9.22	1.00317	1.3254	1.59444	2.44444
ythan	135	601	8.90	1.03962	1.17897	1.16862	1.618
celegans-metabolic	453	2040	9.01	1.00722	1.07502	0.961172	1.10581
Metabolic-CE	1173	2864	4.88	1.07957	1.72941	1.04025	1.48642
Metabolic-EC	2275	5763	5.07	0.996128	1.66706	0.991521	1.66804
Metabolic-SC	1511	3833	5.07	0.949767	1.62123	1.02787	1.63297
TRN-EC-Alon	481	519	2.16	1.00146	1.24606	0.946631	1.14837
prison	67	182	5.43	0.905556	1.33333	1.92222	2.65556
polblogs	1490	19025	25.54	1.04765	1.13332	1.11063	1.22623
Tr1	4440	12873	5.80	0.944061	1.11228	1.02702	1.1925
Tr2	688	1079	3.14	1.04979	1.14423	0.993555	1.21228

Acknowledgement

This work was supported by the National Natural Science Fund of China (Nos 61672332, 61322211, 61432011, U1435212), Program for New Century Excellent Talents in University (No. NCET-12-1031), National Key Basic Research and Development Program of China(973) (Nos 2013CB329404,

2013CB329502), Program for the Outstanding Innovative Teams of Higher Learning Institutions of Shanxi, and Program for the Young San Jin Scholars of Shanxi and the Overseas Returnee Research Program of Shanxi Province under Grant (No. 2017023). We would like to thank the anonymous reviewers for their valuable comments and suggestions.

References

1. Akbari, M., Chua, T. S. Leveraging Behavioral Factorization and Prior Knowledge for Community Discovery and Profiling. *Wsdm'17: Proceedings of the Tenth Acm International Conference on Web Search and Data Mining*, 2017, 71-79. <https://doi.org/10.1145/3018661.3018693>
2. Balas, M. J. Trends in Large Space Structure Control Theory: Fondest Hopes, Wildest Dreams. *IEEE Transactions on Automatic Control*, 1982, 27(3), 522-535. <https://doi.org/10.1109/TAC.1982.1102953>
3. Bianconi, G. Interdisciplinary and Physics Challenges of Network Theory. *Europhysics Letters*, 2015, 111(5), 56001. <https://doi.org/10.1209/0295-5075/111/56001>
4. Boccaletti, S., Alameddini, J. A., Guan, S., Leyva, I., Liu, Z., Sendiña-Nadal, I., Wang, Z., Zou, Y. Explosive Transitions in Complex Networks' Structure and Dynamics: Percolation and Synchronization. *Physics Reports-Review Section of Physics Letters*, 2016, 660, 1-94. <https://doi.org/10.1016/j.physrep.2016.10.004>
5. Cai, N., He, M., Wu, Q., Khan, M. J. On Almost Controllability of Dynamical Complex Networks with Noises. *Journal of Systems Science & Complexity*, 2019, 32(4), 1125-1139. <https://doi.org/10.1007/s11424-017-6273-7>
6. Christakis, N. A., Fowler, J. H. The Collective Dynamics of Smoking in a Large Social Network. *New England Journal of Medicine*, 2008, 358(21), 2249-2258. <https://doi.org/10.1056/NEJMsa0706154>
7. Gao, J., Liu, Y. Y., D'souza, R. M., Barabási, A. L. Target Control of Complex Networks. *Nature Communications*, 2014, 5415. <https://doi.org/10.1038/ncomms6415>
8. Guo, W. F., Zhang, S. W., Wei, Z. G., Zeng, T., Liu, F., Zhang, J., Wu, F. X., Chen, L. Constrained Target Controllability of Complex Networks. *Journal of Statistical Mechanics-Theory and Experiment*, 2017, 2017(6), 063402. <https://doi.org/10.1088/1742-5468/aa6de6>
9. Hahn, A., Kranz, G. S., Sladky, R., Ganger, S., Windischberger, C., Kasper, S., Lanzenberger, R. Individual Diversity of Functional Brain Network Economy. *Brain Connectivity*, 2015, 5(3), 156-165. <https://doi.org/10.1089/brain.2014.0306>
10. Hautus, M. L. J. Controllability and Observability Conditions of Linear Autonomous Systems. *Proceedings of the Koninklijke Nederlandse Akademie Van Wetenschappen Series a-Mathematical Sciences*, 1969, 72(5), 443-448.
11. Kalman, R. E., Falb, P. L., Arbib, M. A. *Topics in Mathematical System Theory*. New York: McGraw-Hill, 1969.
12. Kim, J. Z., Soffer, J. M., Kahn, A. E., Vettel, J. M., Pasqualetti, F., Bassett, D. S. Role of Graph Architecture in Controlling Dynamical Networks with Applications to Neural Systems. *Nature Physics*, 2018, 14(1), 91-98. <https://doi.org/10.1038/nphys4268>
13. Landini, F., Malerba, F., Mavilia, R. The Structure and Dynamics of Networks of Scientific Collaborations in Northern Africa. *Scientometrics*, 2015, 105(3), 1787-1807. <https://doi.org/10.1007/s11192-015-1635-1>
14. Li, J. C., Zhao, D. L., Ge, B. F., Yang, K. W., Chen, Y. W. A Link Prediction Method for Heterogeneous Networks Based on BP Neural Network. *Physica a-Statistical Mechanics and Its Applications*, 2018, 495, 1-17. <https://doi.org/10.1016/j.physa.2017.12.018>
15. Li, X., Jusup, M., Wang, Z., Li, H., Shi, L., Podobnik, B., Stanley, E. H., Havlin, S., Boccaletti, S. Punishment Diminishes the Benefits of Network Reciprocity in Social Dilemma Experiments. *Proceedings of the National Academy of Sciences of the United States of America*, 2018, 115(1), 30-35. <https://doi.org/10.1073/pnas.1707505115>
16. Liu, Y. Y., Slotine, J. J., and Barabási, A. L. Controllability of Complex Networks. *Nature*, 2011, 473(7346), 167-173. <https://doi.org/10.1038/nature10011>

17. Lu, H., Li, Y., Chen, M., Kim, H., Serikawa, S. Brain Intelligence: Go beyond Artificial Intelligence. *Mobile Networks & Applications*, 2018, 23(2), 368-375. <https://doi.org/10.1007/s11036-017-0932-8>
18. Menara, T., Bassett, D.S., Pasqualetti, F. Structural Controllability of Symmetric Networks. *IEEE Transactions on Automatic Control*, 2019, 64(9), 3740-3747. <https://doi.org/10.1109/TAC.2018.2881112>
19. Moothedath, S., Chaporkar, P., and Belur, M. N. Approximating Constrained Minimum Cost Input-Output Selection for Generic Arbitrary Pole Placement in Structured Systems. *Automatica*, 2019, 107, 200-210. <https://doi.org/10.1109/TAC.2018.2797210>
20. Moothedath, S., Chaporkar, P., Belur, M.N. A Flow-Network-Based Polynomial-Time Approximation Algorithm for the Minimum Constrained Input Structural Controllability Problem. *IEEE Transactions on Automatic Control*, 2018, 63(9), 3151-3158. <https://doi.org/10.1016/j.automatica.2019.05.002>
21. Perren, R., Kozinets, R. V. Lateral Exchange Markets: How Social Platforms Operate in a Networked Economy. *Journal of Marketing*, 2018, 82(1), 20-36. <https://doi.org/10.1145/3018661.3018693>
22. Rossetti, G., Cazabet, R. Community Discovery in Dynamic Networks: A Survey. *Acm Computing Surveys*, 2018, 51(2), 35. <https://doi.org/10.1145/3018661.3018693>
23. Schultz, D. H., Cole, M. W. Higher Intelligence Is Associated with Less Task-Related Brain Network Reconfiguration. *Journal of Neuroscience*, 2016, 36(33), 8551-8561. <https://doi.org/10.1509/jm.14.0250>
24. Sun, Y. Z., Leng, S. Y., Lai, Y. C., Grebogi, C., Lin, W. Closed-Loop Control of Complex Networks: A Trade-Off between Time and Energy. *Physical Review Letters*, 2017, 119(19), 198301. <https://doi.org/10.1103/PhysRevLett.119.198301>
25. Wang, P., Xu, B., Wu, Y., Zhou, X. Link Prediction in Social Networks: The State-of-the-art. *Science China-Information Sciences*, 2015, 58(1), 1-38. <https://doi.org/10.1145/3018661.3018693>
26. Xiao, F., Yang, H., Ye, H. B. Physics of Day-to-Day Network Flow Dynamics. *Transportation Research Part B-Methodological*, 2016, 86, 86-103. <https://doi.org/10.1007/s11432-014-5237-y>
27. Yan, G., Vértes, P. E., Towilson, E. K., Chew, Y. L., Walker, D. S., Schafer, W. R., Barabási, A. L. Network Control Principles Predict Neuron Function in the Caenorhabditis Elegans Connectome. *Nature*, 2017, 550(7677), 519-523. <https://doi.org/10.1038/nature24056>
28. Yuan, Z., Zhao, C., Di, Z., Wang, W. X., Lai, Y. C. Exact Controllability of Complex Networks. *Nature Communications*, 2013, 4, 2447. <https://doi.org/10.1038/ncomms3447>
29. Zanudo, J. G. T., Yang, G., Albert, R. Structure-Based Control of Complex Networks with Nonlinear Dynamics. *Proceedings of the National Academy of Sciences of the United States of America*, 2017, 114(28), 7234-7239. <https://doi.org/10.1073/pnas.1617387114>
30. Zhang, W., Hou, L., Wang, J., Geng, S., Wu, W. Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip. *Proceedings of the 2009 WRI Global Congress on Intelligent Systems*, 2009, 329-333. <https://doi.org/10.1109/GCIS.2009.110>
31. Zhang, X., Lv, T., Yang, X., Zhang, B. Structural Controllability of Complex Networks Based on Preferential Matching. *Plos One*, 2014, 9(11), e112039. <https://doi.org/10.1371/journal.pone.0112039>