

A Degree-first Greedy Search Algorithm for the Evaluation of Structural Controllability of Real World Directed Complex Networks

Debayan Das

Department of ETCE, Jadavpur University, Kolkata-700032, India.

ddebayan50@gmail.com

Ayan Chatterjee

Department of ETCE, Jadavpur University, Kolkata-700032, India.

ayanetce@gmail.com

Nabamita Pal

Department of EE, Techno India, Salt lake, Kolkata-700091, India

nabamita.techno@gmail.com

Amitava Mukherjee

IBM India Private Ltd., Salt lake, Kolkata-700091, India

amitava.mukherjee@in.ibm.com

Mrinal K Naskar

Department of ETCE, Jadavpur University, Kolkata-700032, India.

mrinalnaskar@yahoo.co.in

Received: December 14, 2013

Accepted: March 9, 2014

Published: April 30, 2014

DOI: 10.5296/npa.v6i1.4756

URL: <http://dx.doi.org/10.5296/npa.v6i1.4756>

ABSTRACT

Ubiquitous data flow through a directed complex network requires the complete structural controllability of the network. For evaluating the structural controllability of any network, determination of maximum matching in the network is a cardinal task and has always been a problem of immense concern. Its solution is mandatory in structural control theory for controlling real world complex networks. The existing classical approach through the Hopcroft-Karp algorithm and other proposed algorithms require the determination of the bipartite equivalent graph (i.e., network), which belongs to the NP-complete class of problems. In this article, we propose a degree-first greedy search algorithm to determine maximum matching in unipartite graphs without determining its bipartite equivalent. Thus this classical problem of the NP-Complete class can be solved using the heuristic, with

reduced complexity. This algorithm can be efficiently used to find maximum matching in most of the real world complex networks that follow Erdős-Rényi model. Simulation results obtained using our heuristic reveal that dense and homogenous networks can be controlled with fewer controller nodes popularly termed as driver nodes, compared to the sparse inhomogeneous networks.

Keywords: Augmenting path, complex networks, driver nodes, Erdős-Rényi model, maximum matching, structural controllability, unipartite and bipartite graph.

1. INTRODUCTION

A wide variety of real life structures can be naturally represented by complex networks. Any social network, different networks within our body (neural networks), various kind of Biological networks, World Wide Web etc can be effectively modelled in complex networks. These realistic systems are composed of highly dynamical units. The very first loom to capture the global properties of these practical systems as complex network model is to assume the dynamic units as the nodes of the complex network and the interconnections between those units as links. By help of Statistical Mechanics and different area of graph theory in Mathematics the structural behaviour of complex network can be analysed properly.

Complex networks have a vast scope of study which has made it a budding field of research. Controllability of these Complex networks is a very interesting and relevant topic of research among other areas regarding complex network study [1-3].

The contents of our paper are following: Section 2 describes the generic concepts that are required to design our heuristic. Some important definitions and generic concepts have been illustrated in section 2. Section 3 proposes heuristic followed by an example that explains this proposed heuristic. Section 4 discusses results and other inferences, while section 5 evaluates the complexity of this heuristic. Section 6 describes the scope of future work and concludes the paper.

2. RELATED WORK AND MOTIVATION

2.1 Controllability of a real-world network

Majority of the real-world systems are driven by non-linear processes, but the controllability of those non-linear systems is structurally similar to that of linear systems in many aspects. The canonical linear, time-invariant dynamics for a linear system is represented by next equation

$$\frac{dx(t)}{dt} = Ax(t) + Bu(t) \quad (1)$$

where $x(t)$ is the state vector of dimension $1 \times N$, where N is the number of nodes at time t . A is the system matrix of dimension $N \times N$, that describes the system's wiring diagram and the interaction strength between the components. B is the $N \times M$ input matrix ($M \leq N$) which identifies the nodes controlled by an external controller, and $u(t)$ is input matrix of dimension $M \times 1$.

If there exists an input $u(t)$ which transfers the initial state X_0 to the state X_1 in a finite time t_1 , then the state X_0 is called controllable. Analytically it is said that if the system is fully controllable then Kalman's Controllability test matrix Q_c

$$Q_c = [B: AB: A^2B \dots A^{n-1}B] \quad (2)$$

will be of Rank N , that is

$$\text{rank}(Q_c) = N \quad (3)$$

So, from the definition of controllability it can be said that, in a controllable system by the help of any external input the internal states of a system can be controlled. But for various reasons this classical definition of controllability fails to get acceptance in the case of complex networks. First and foremost, the classical definition of controllability is applicable for undirected networks whereas almost all real-world complex networks are directed. Secondly, complex networks normally consist of thousands of nodes and branches, it becomes almost impossible to calculate Kalman's controllability test matrix Q_c from these complex graphs [1-3]. These reasons dramatically limit the acceptability of the classical controllability theory for complex networks [4].

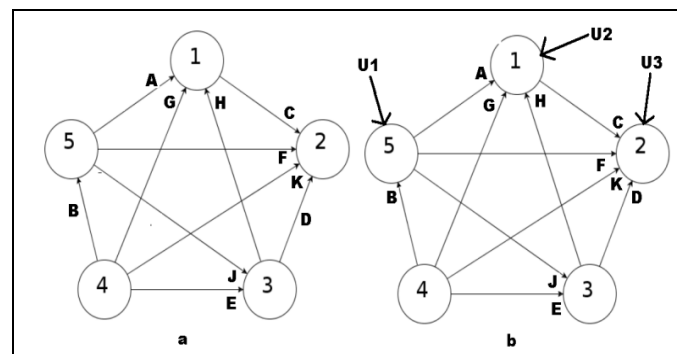
Hence, to define the controllability of complex networks a new concept "Structural Controllability" evolved [5]. This concept was introduced by scientist Lin in the year 1970. A structurally controllable system can be shown to be controllable for almost all weight combinations. Thus, structural controllability helps us to overcome our inherently incomplete knowledge of the link weights in the matrix A [5].

In case of structural controllability we assume that $G(A)$ is a directed graph drawn from an equivalent directed network. This graph $G(A)$ can also be defined by the linear equation (1). Here the state or system matrix $A = (a_{ef})_{n \times n}$, here the element a_{ef} is 0 if no link exists between the nodes e and f in the graph $G(A)$. Otherwise, a_{ef} represents the weight of the link between the nodes e and f . It will be positive or negative depending on the direction of the link i.e. it is from e to f or from f to e [5].

The set of nodes in the directed graph $G(A)$ (Fig. 1a) can be divided into two categories, i.e. $N = N_A \cup N_B$, i) the set of state nodes N_A (Fig. 1b) 3 & 4, and ii) the set of input nodes N_B ,

(Fig:1b) 1,2 and 5. If there is m number of inputs in a directed graph then the number of input nodes will also be m . The edges of the graphs are also divided into two categories, i.e. $E=E_A \cup E_B$, i) the set of edges between the state nodes E_A (from Fig. 1b: E) and ii) the set of edges between the state nodes and input nodes or between two input nodes E_B (from Fig: 1b A, B, C, D, F, G, H, J &K). The input nodes of a digraph are also called the Origin and the state nodes directly connected with the input nodes are called controlled nodes. The number of controlled nodes is normally always greater than the number of origins because one origin can be connected with several controlled nodes. The input nodes are called Driver nodes N_D (from Fig: 1b 1, 2 and 5). It is obvious that the number of driver nodes in a digraph will be equal to the number of total inputs (from Fig. 1b: U_1, U_2 and U_5) to the graph.

From the classical definition of controllability it can be said that a system can be controlled by external inputs if and only if the system is fully controllable and from the definition of driver nodes we can say that the number of driver nodes are always equal to the number of inputs to the system. Now, if we summarize these two facts then it can be concluded that if minimum number of driver nodes for a digraph can be calculated then the number of inputs needed to control the system can also be evaluated. We are mainly interested in identifying the minimum number of driver nodes, N_D , whose control is sufficient to control the system's dynamics.



1(a)

1(b)

Fig: 1 Different nodes and links in a digraph

Now, in the basic concept of ‘Structural Controllability’ it is considered that the system obeys the linear equation (1) and the matrixes A & B are assumed to be structured. (A, B) are structured means the elements of A and B are either 0 (represents absence of connection between parameters) or independent free parameters (weights of the links) [13]. The System will be called ‘Structurally controllable’ if the independent free parameters of the matrix pair A and B can be fixed to certain values for which A and B can satisfy Kalman’s controllability test. When a system is controllable for any value of the independent free parameters of (A, B) then the system is called ‘Strongly Structurally controllable’ [6].

2.2 Lin's structural controllability theorem:

Scientist Lin has given a structural controllability theorem [5] where he introduced two conditions a) inaccessibility and b) dilation. In a digraph a node is called inaccessible if that node can't be influenced by an external input. For example, an isolated node is an inaccessible node. In a digraph, if there is a driver node which is not the starting node of a directed path then the nodes upstream to the driver node will be the examples of inaccessible nodes as those nodes can't be accessed by the external input connected to the driver node. In Fig 2, node 4 is inaccessible node.

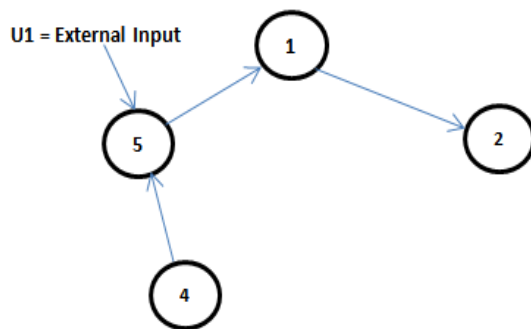


Fig: 2 Inaccessible node

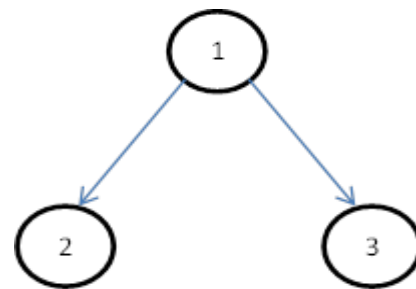


Fig 3: Dilation

In a digraph if there exist greater number of subordinate nodes than the number of superior nodes then it is said that the graph contains dilation. In Fig 3, node 2 and 3 are subordinate nodes which are controlled by one superior node 1. So, here dilation exists in the digraph. In Lin's structural controllability theorem it is stated that a digraph will be fully structurally controllable if and only if it contains no inaccessible nodes and dilation.

Now to evaluate structural controllability of a certain digraph we have to determine the number of external inputs needed to control the network. It will be highly economical if we can be able to determine the minimum number of inputs to control it. For this purpose Minimum input theorem can be followed.

2.3 Minimum inputs theorem

The Minimum Inputs theorem states that, in a controlled network, the minimum number of inputs equals the minimum number of driver nodes needed to fully control the network. If the minimum number of driver nodes is one, i.e. only one input is necessary to fully control the network, then the controlled network is said to be perfectly matched.

2.4 Maximum matching in undirected networks [7]

Another important concept in connection with determination of minimum number of external input is maximum matching. Let $G = (V, E)$ be a finite undirected graph (without

loops or multiple edges) having the vertex set V and the edge set E . An edge incident with vertices v and w is denoted as $\{v, w\}$. A set M belonging to E is a matching if no vertex $v \in V$ is incident with more than one edge in M . A matching of maximum cardinality is called a maximum matching.

2.5 Maximum matching in directed networks

A matching M of the graph G is an edge set such that no two edges of M share their endpoints. Given a bipartite graph $G = (V, E)$, maximum matching is a matching that contains the largest possible number of edges [4,8]. For a given directed complex network, the largest set of edges, without common start and end nodes is said to be the maximum matching for that network. If maximum matching of a network yields only one start node and end node, then it is called complete matching (Fig. 4) [9].

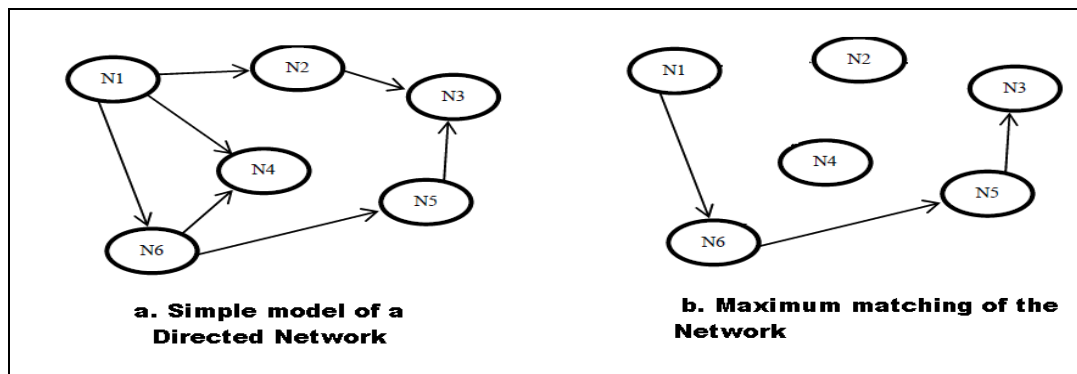


Fig: 4 Maximum Matching

2.6 Driver node and augmenting path

In a maximum matched undirected network, driver nodes are designated by the end points of matched paths, called the augmenting paths. In case of directed networks, driver node is the starting node of the augmenting path, where desired input is applied for structurally controlling the whole path corresponding to it. In the Fig. 5, $N1$ and $N2$, and $N4$ are the driver nodes, and the directed path (marked in red) is the augmenting path corresponding to driver node $N1$. Nodes $N2$ and $N4$ are isolated nodes. Hence they must be controlled independently. So, $N2$ and $N4$ are also treated as driver nodes.

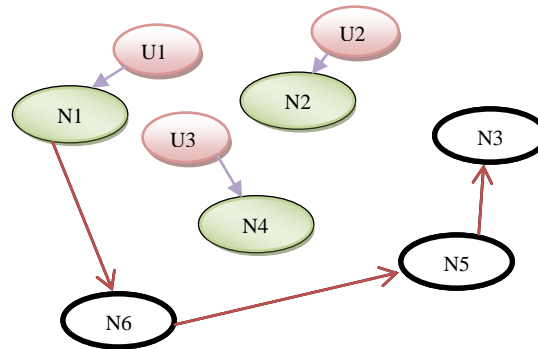


Fig 5. Driver nodes (N1, N2, and N4) with their augmenting paths (marked in red)

3. HEURISTIC FOR MAXIMUM MATCHING

In this section, we propose a heuristic for finding maximum matching in directed graphs directly from its unipartite form, i.e., without obtaining the bipartite equivalent. The main advantage of our approach over the previously existing ones is that it not only determines the number of driver nodes, but also specifies them along with their corresponding augmenting paths [25]. Thus all information required for full controllability of the network is obtained. In directed complex networks, determination of augmenting paths refer to the sets of directed links which share no common start or end point.

The steps given below are to be followed for determining maximum matching in directed networks using our heuristic:

Step 1: Firstly, determine the node in the part of the network under consideration, which possesses the maximum out degree. Let the node be denoted by N_a . This node N_a is considered as the *driver node*.

Step 2: Next, determine the node connected to N_a (driver node) having the maximum out degree. Let the node be denoted as N_b .

Step 3: Now, consider the link from node N_a to N_b , and discard all other links connected to N_a , thus setting its degree to 1. Include N_a to the augmenting path and discard it from further consideration.

Step 4: Finally, assign N_b to N_a and repeat step 2 and step 3 till the degree of N_a becomes 1.

After completion of step 4, the augmenting path having N_a (from step 1) as its driver node is obtained.

Step 5: Repeating steps 1 to 4, the next augmenting path is obtained.

Step 6: Repeat the whole process till the entire network has been considered.

Now, we explain our algorithm with a simple example. We consider a network with 6 nodes and 7 edges given below:

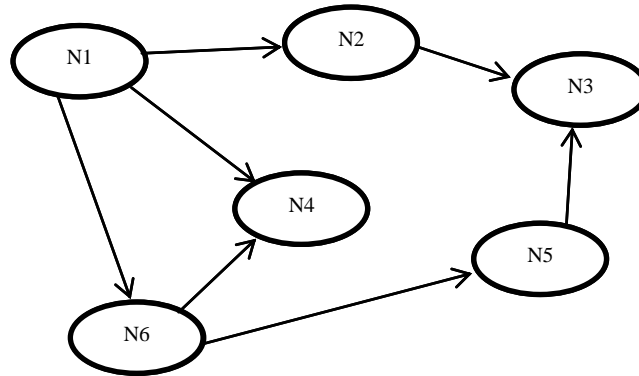


Fig 6(a)

We explain our heuristic with Fig. 6. Fig. 6(a) shows the directed network on which we implement our algorithm.

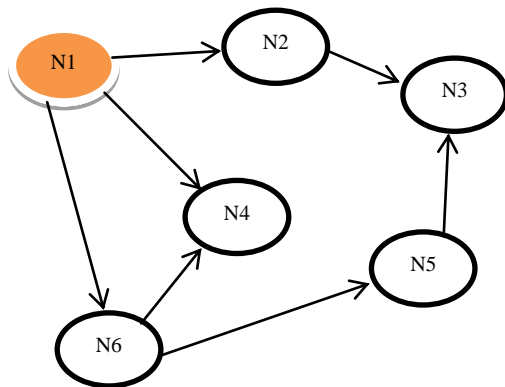


Fig 6(b)

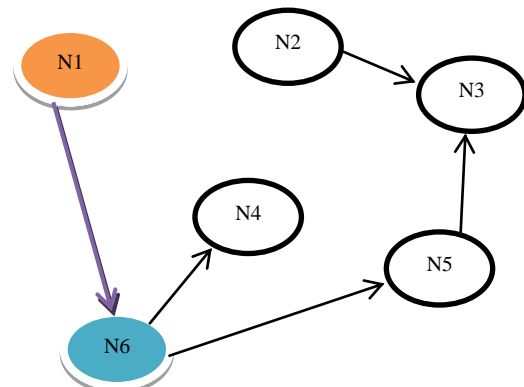


Fig. 6(c)

Applying Step 1 of our heuristic on the network 6(a), the maximum out-degree node is obtained as N1. N1 is considered as a driver node (marked orange), as shown in Fig. 6(b). Now applying Step 2 on the Fig. 6(b), N6 is obtained as the maximum out-degree node connected to the driver node N1. According to Step 3, all the links of N1 are discarded except the one from N1 to N6 (marked in violet), as shown in Fig. 6(c).

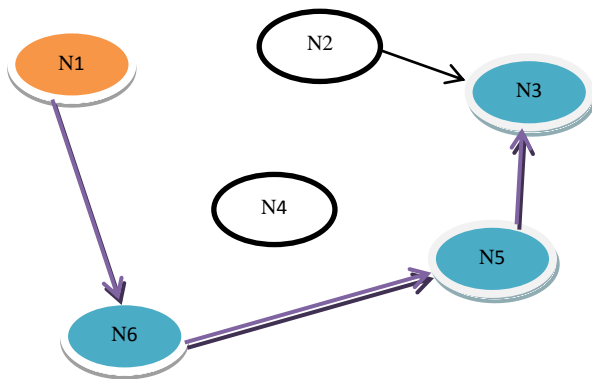


Fig 6(d)

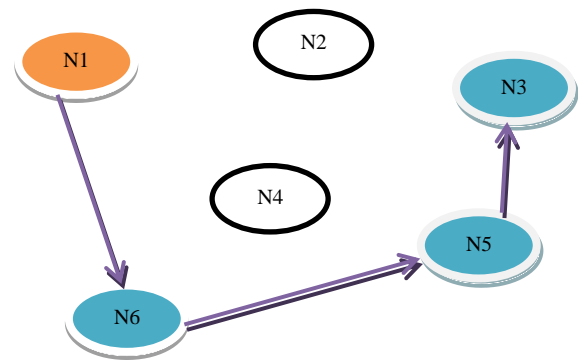


Fig 6(e)

As in Step 4, the above process is repeated for the node N6, N5 and N3, till degree of those nodes becomes 1, as in Fig. 6(d). In Fig. 6(e), the largest set of edges having no common start and end nodes is obtained. Thus fig. 6(e) gives the maximum matching of the original network. Here, nodes N4 and N2 are isolated nodes. Hence, both N2 and N4 are considered as driver nodes. The network contains only one augmenting path (marked in violet) with N1 as the driver node, and N6, N5, N3 as the matched nodes (marked in blue).

Step 5 is skipped here, as there clearly exists no other augmenting paths. In Step 6, we ensure that the whole network has been traversed once.

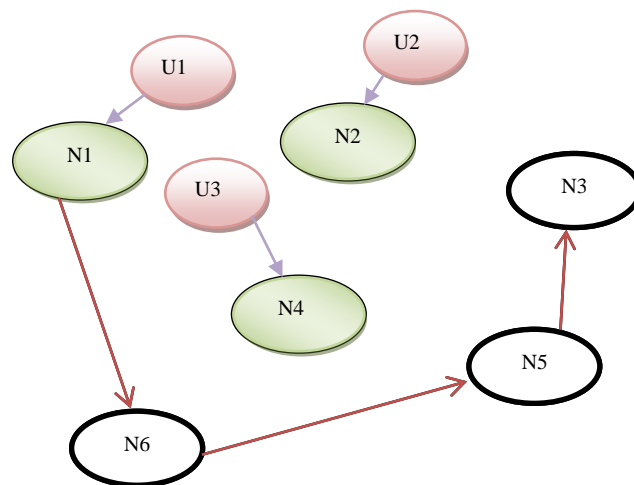


Fig 6(f)

Finally, the required controlled network is obtained, which is controlled by three input vertices u_1, u_2, u_3 (marked in pink) applied to the state vertices N_1, N_2 and N_4 (marked in brown) respectively, as shown in Fig 6(f).

4. SIMULATION RESULTS

4.1. Simulation Environment

Our algorithm has been mainly applied to networks which follow Erdős-Rényi model. The networks under consideration has been generated by a simulator namely Cytoscape (Version 2.8.3). Cytoscape is an open-source bio-informatics software platform for visualising molecular interaction networks and integrating with gene expression profiles and other state data. Here, we have used the random network generation plugin with the latest version of Cytoscape, released in May, 2012. We have simulated our algorithm on the generated networks using Eclipse, a Java integrated development environment. Eclipse contains a base workspace and an extensible plug-in system for customizing the environment.

During generation, the networks were specified by the parameters N (number of nodes) and L (number of links). In our simulation results, the degree of the nodes has been considered as the parameter measuring the criticality or priority of the nodes. It should be kept in mind that other characteristic (attributes and variables) of nodes can be treated as the parameter. Our algorithm has been applied to different types of networks, specified by graphs with particular number of links and nodes. From the simulation results we also arrive at some intuitive interpretations. We have classified the graphs as dense, semi-dense and sparse. The graphs which have comparatively large number of links with respect to the number of nodes are considered to be dense. On the other hand, the graphs which have very small number of links with respect to the size of the network are considered sparse. The graphs whose density of links with respect to number of nodes lies in between these two classes have been classified as semi-dense. It is clear from the definition mentioned above that this classification is not robust in practical sense.

From the number of driver nodes obtained from our simulation results it is clear that lesser number of driver nodes is suffice to structurally control a denser network and the number increases as the network tends to become sparse. It is also intuitively clear that when number of links in a network increases (dense network), number of augmenting paths should decrease, as length of such paths increase with increase in connectivity among the nodes. In another paper [8], Barabasi et.al have shown that n_d varies as $e^{-\langle k \rangle}$, where k is the Poisson variable where $P(k)$ represents the probability of a link to exist between two nodes, and n_d is the density of driver nodes, obtained as the ratio of the minimum driver nodes required and

the total number of nodes in the entire network. As a network becomes denser, $\langle k \rangle$ increases and thus n_d decreases, which leads to similar result obtained from our simulation.

In the tables below, N: Number of nodes, L: Number of links, n_d : Density of driver nodes, N_d : Number of driver nodes. ($N_d = N * n_d$)

In table 1, we show the experimental results for dense networks like World Wide Web, as obtained using our algorithm. We see that these dense networks can be controlled by very small number of driver nodes (N_d) compared to the total number of nodes in the network (N). We plot our results for these dense networks in fig. 7.

TABLE 1: Dense networks

| Type | Name | N | L | L/N | N_d | $n_d^{experimental}$ | Run time (in mSec) |
|----------------------|-----------------|------|-------|----------|-------|----------------------|--------------------|
| World Wide Web | Political blogs | 1224 | 19025 | 15.5433 | 76 | 0.062091503 | 658 |
| Intra-organizational | Freemans-2 | 34 | 695 | 20.44118 | 2 | 0.058823529 | 128 |
| | Freemans-1 | 34 | 830 | 24.41176 | 1 | 0.029411765 | 146 |
| Regulatory | TRN-Yeast-1 | 441 | 12873 | 29.19048 | 15 | 0.034013605 | 532 |

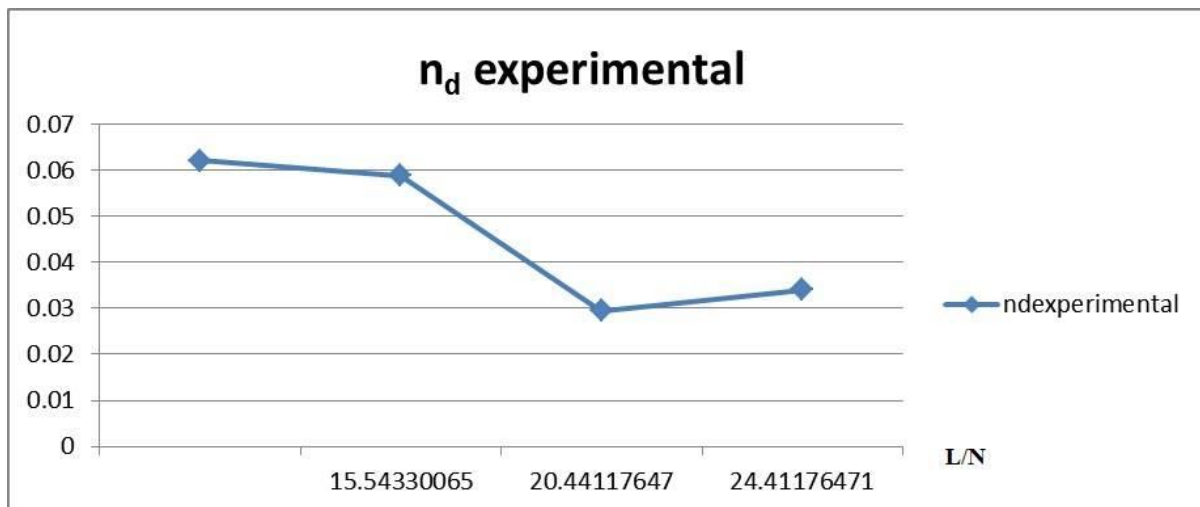


Figure 7: $n_d^{experimental}$ vs L/N for dense networks

In table 2, we show the experimental results obtained by applying our heuristic to semi-dense networks like the food web. We see that controlling a semi-dense network with same number of driver nodes as a dense network requires more number of driver nodes compared to the dense network. This is due to the reason that L/N ratio is less in case of these semi-dense networks. We show the variation of driver node density (n_d) with the density of the network L/N in fig 8.

TABLE 2: Semi-dense networks

| Type | Name | N | L | L/N | N _d | n _d ^{experimental} | Run time (in mSec) |
|----------------------|------------------------|------|--------|----------|----------------|--|--------------------|
| Food Web | Ythan | 135 | 601 | 4.451852 | 25 | 0.185185185 | 169 |
| | Little Rock | 183 | 2494 | 13.62842 | 15 | 0.081967213 | 257 |
| | Seagrass | 49 | 226 | 4.612245 | 8 | 0.163265306 | 70 |
| Neuronal | Caenorhabditus elegans | 297 | 2345 | 7.895623 | 35 | 0.117845118 | 260 |
| Social Communication | Uclocline | 1899 | 20296 | 10.68773 | 176 | 0.092680358 | 780 |
| | Email Epoch | 3188 | 39256 | 12.31368 | 252 | 0.079046424 | 1257 |
| Trust | Wikivote | 7115 | 103689 | 14.5733 | 486 | 0.068306395 | 4025 |

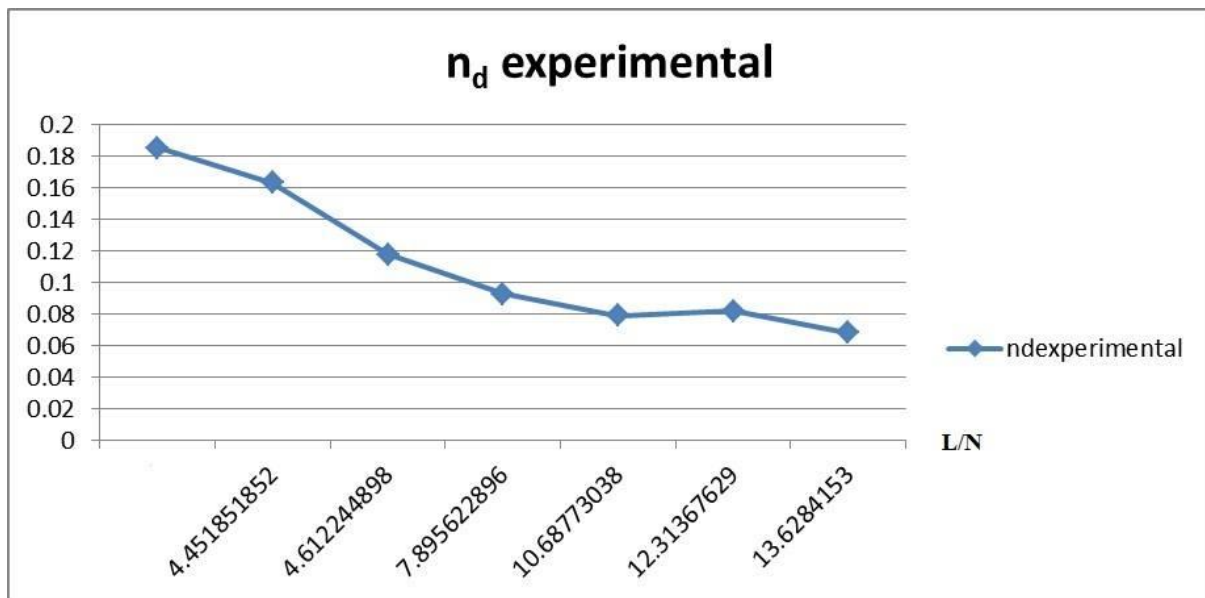


Figure 8: n_d experimental vs L/N for semi-dense networks

Table 3 shows the experimental results obtained for sparse networks. We see that to control a sparse network completely, more number of driver nodes are necessary compared to dense or semi-dense networks. The variation of driver node density (n_d) with the network density (L/N) is shown in fig. 9.

TABLE 3: Sparse networks

| Type | Name | N | L | L/N | N _d | n _d ^{experimental} | Run time (in mSec) |
|---------------------|--------------------------|-------|-------|----------|----------------|--|--------------------|
| Regulatory | TRN-EC-2 | 418 | 519 | 1.241627 | 237 | 0.566985646 | 162 |
| | TRN-EC-1 | 1550 | 3340 | 2.154839 | 636 | 0.410322581 | 499 |
| | TRN-Yeast-2 | 688 | 1079 | 1.568314 | 347 | 0.504360465 | 214 |
| Electronic circuits | S208 | 122 | 189 | 1.54918 | 67 | 0.549180328 | 84 |
| | S420 | 252 | 399 | 1.583333 | 128 | 0.507936508 | 107 |
| Food Web | Grassland | 88 | 137 | 1.556818 | 51 | 0.579545455 | 60 |
| Metabolic | Escherichia coli | 2275 | 5763 | 2.533187 | 844 | 0.370989011 | 609 |
| | Saccharomyces cerevisiae | 1511 | 3833 | 2.536731 | 539 | 0.356717406 | 418 |
| Trust | Prison Inmate | 67 | 182 | 2.716418 | 22 | 0.328358209 | 66 |
| | College Student | 32 | 96 | 3 | 9 | 0.28125 | 47 |
| Internet | P2p-2 | 8846 | 31839 | 3.599254 | 2388 | 0.269952521 | 3541 |
| | P2p-3 | 8717 | 31525 | 3.616497 | 2493 | 0.285992887 | 3593 |
| | P2p-1 | 10876 | 39994 | 3.677271 | 2926 | 0.269032733 | 6461 |

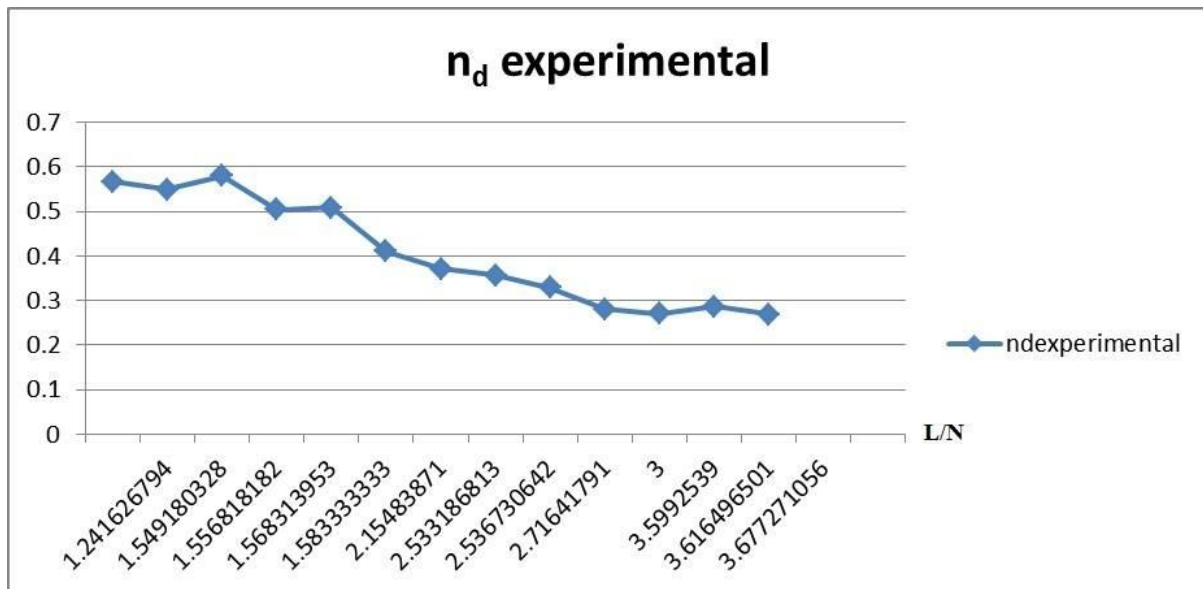


Figure 9: n_d experimental vs L/N for sparse networks

Again from the results in Table 4, we see a good match between the results obtained by applying our heuristic on some real world networks and the density of driver nodes (n_d^{real}) calculated in the practical real-world networks [8].

TABLE 4: Comparison of our results with that of the real network [8]

| N | L | L/N | N_d | n_d^{real} | $n_d^{experimental}$ |
|------|------|----------|-------|--------------|----------------------|
| 418 | 519 | 1.241627 | 237 | 0.519 | 0.566986 |
| 88 | 137 | 1.556818 | 51 | 0.523 | 0.579545 |
| 2275 | 5763 | 2.533187 | 844 | 0.382 | 0.370989 |
| 1511 | 3833 | 2.536731 | 539 | 0.329 | 0.356717 |
| 297 | 2345 | 7.895623 | 35 | 0.165 | 0.117845 |
| 34 | 830 | 24.41176 | 1 | 0.029 | 0.029412 |

From table 4, we see that for many networks, the mismatch in values of n_d experimental compared to n_d real is notable. Fig. 10 shows the comparison of the results (n_d) using our heuristic and that of the real network. Even if controllability is decided mainly by degree, this mismatch in results might be due to the effect of degree correlations. However, network controllability can be altered only by using betweenness centrality and closeness centrality, without using degree (graph theory) or degree correlations at all.

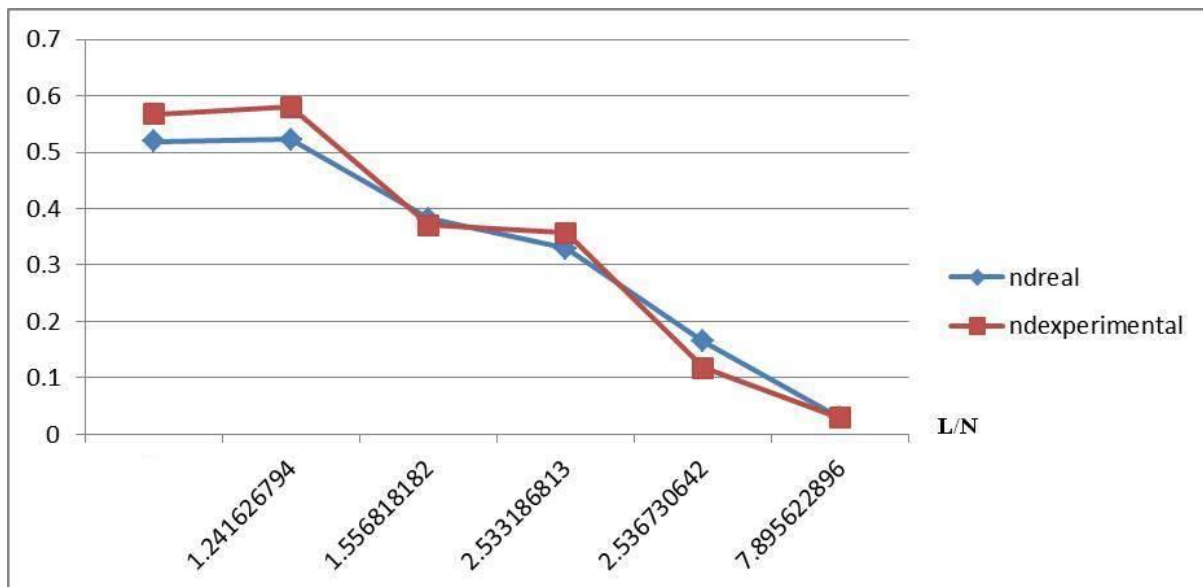


Figure 10: n_d experimental vs L/N for sparse networks

5. COMPLEXITY

In the theory of computational complexity, NP may be defined as the set of decision problems that can be solved in polynomial time [4] on a nondeterministic Turing machine. A problem p in NP is also in NPC if and only if every other problem in NP can be transformed into P in polynomial time. The available approach for solving the problem of determining maximum matching in directed graphs belonged to this class of problems.

In the worst case, it may happen that all links will exist among n number of nodes. So at first we have to search among n nodes for determining the node with the maximum out degree. For the next step, we have to search among $(n-1)$ nodes and so on till the number becomes 1.

Thus, required time complexity in the worst case simply becomes $1+2+3+\dots+n = \sum_{i=1}^n i = n(n+1)/2 = (n^2+n)/2$.

So time complexity of this algorithm is of the order of n^2 .

Comparison with the $O(n^{5/2})$ algorithm proposed by Hopcroft and Karp combined with classical process of obtaining bipartite equivalent of complex networks, reveals the advantage of our heuristic. From our heuristic we are able to determine uniquely the driver nodes and corresponding augmenting paths. Thus approach of structurally controlling any network can be uniquely and efficiently found out.

5. CONCLUSION

After obtaining the driver nodes and corresponding augmenting paths, we apply signals to the driver nodes, whose retrieval indicate the continuity and accessibility of the augmenting paths corresponding to some specific retrieving technique for the applied signals. Any defect in the network can thus be efficiently detected and thereby down time of the whole system can be reduced. This increases the robustness of the network against unwanted failures and breakdowns. [10]

We ended up with two major findings while developing the heuristics for maximum matching. Firstly, we see that the number of driver nodes for a complex network, compared to the total number of nodes in the network increases with the density of the network. Secondly, we also infer from our simulation that a system's controllability is mainly determined by the underlying network's degree distribution, $P(k_{in}, k_{out})$.

Control is the most important issue in the real world complex networks. Till date, the most noted algorithm available for the computation of maximum matching in complex networks is the classical Hopcroft-Karp algorithm [7], and some other algorithms [11-12] have been

proposed, but all of them require that the isomorphic bipartite equivalent of the network be obtained. Again, the bipartite equivalent of all complex networks cannot be determined. Neither does an algorithm exist to determine the bipartite equivalent of any given unipartite complex network. Hence, the proposed algorithm might well serve the purpose of finding the maximum matching, and thus fully controlling a directed weighted real world complex network. Also, the algorithm may be easily modified to solve maxflow problems.

The approach which has been described in this article is very much relevant in the real-life complex networks like power grids, Internet networks, Different biological networks (in brain or in nervous system), ecological networks etc. By using the algorithm we can recognize the set of driver nodes and the time-dependent control of those nodes can guide the network dynamics. From the results which have been shown in this paper we can conclude that for a dense and homogeneous network the number of driver nodes is least. It increases with the decrease in number of links in a network. So the number of driver nodes will be larger for semi-dense and sparse networks than the dense one. The networks with lesser number of driver nodes are easier to control. The developed approach provides a framework to deal with the controllability issue of an arbitrary network which also represents a fundamental step towards the ultimate control strategy of complex systems.

In the long run, we need to solve numerous questions that are being raised. For example, we need to ensure whether the networks are efficient and robust against all sorts of failures. i.e. site or bond. Does these complex networks help in information flow? How to predict the health of the system represented by the network?

Speculation is that the solutions to these questions will mark the opening of new dimension, in studying the controllability of complex networks.

References

1. H. G. Tanner, "On the controllability of nearest neighbor interconnections" *43rd IEEE Conference on Decision and Control*, (CDC) 2004. **3**, **246**. <http://dx.doi.org/10.1109/CDC.2004.1428782>
2. A. Lombardi and M. Hornquist, *Phys. Rev. E* **75**, **056110**(2007). <http://staffwww.itn.liu.se/~micho/research/2007Reglernatverk-07.pdf>
3. B. Liu, T. Chu, L. Wang, and G. Xie, "Controllability of a Leader-Follower Dynamic Network With Switching Topology" *IEEE Trans. Auto. Contr.* **53**, **1009**(2008). <http://dx.doi.org/10.1109/TAC.2008.919548>
4. Kalman, R.E. Mathematical description of linear dynamical systems. *J. Soc Indus. Appl. Math Ser.* **A1**, 152-192 (1963). <http://dx.doi.org/10.1137/0301010>
5. C. T. Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, No. 3, pp. 201-208, June 1974. <http://dx.doi.org/10.1109/TAC.1974.1100557>

6. H. Mayeda and T. Yamada, “Strong structural controllability”, *SIAM J. Control and Optimization*, vol. 17, pp. 123–138, 1979. <http://dx.doi.org/10.1137/0317010>
7. Hopcroft, John E. & Karp, Richard M. “An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs”, *SIAM J. Comput* **2** (1973). <http://dx.doi.org/10.1109/SWAT.1971.1>
8. Liu, Y. Y., Slotine J.J. & Barabasi A.L. Controllability of complex networks. *Nature* **473**, 167-173 (2011). <http://dx.doi.org/10.1038/nature10011>
9. Zdeborova, L. & Mezard, M. The number of matchings in random graphs. (2006). <http://dx.doi.org/10.1088/1742-5468/2006/05/P05003>
10. Albert, R., Jeong, H. & Barabasi, A.-L. Error and attack tolerance of complex networks. *Nature* **406**, 378-382 (2000). <http://dx.doi.org/10.1038/35019019>
11. Hanckowiak M., Karonski M., Panconesi A., On the Distributed Complexity of Computing Maximal Matchings, *ACM-SIAM SODA* (1998). <http://dx.doi.org/10.1137/S0895480100373121>
12. Galil Z., Efficient Algorithms for finding Maximum Matching in Graphs, *ACM Computing Surveys* (1980). <http://dx.doi.org/10.1145/6462.6502>
13. Evans, T. S. Complex Networks. *Article for Contemporary Physics*, **Imperial/TP/3-04/12** (2004). <http://dx.doi.org/10.1080/00107510412331283531>
14. Faloutsos, M., Faloutsos, P. & Faloutsos, C. On power-law relationships of the internet topology. *Comp. Comm. Rev.* **29**, 251-262 (1999). <http://dx.doi.org/10.1145/316188.316229>
15. Broder, A. *et al.* Graph structure in the web. *Comput. Netw.* **33**, 309-320 (2000). [http://dx.doi.org/10.1016/S1389-1286\(00\)00083-9](http://dx.doi.org/10.1016/S1389-1286(00)00083-9)
16. Strogatz, S.H. Exploring complex networks. *Nature* **410**, 268-276(2001). <http://dx.doi.org/10.1038/35065725>
17. Lovasz, L. & Plummer, M.D. *Matching Theory* (American Mathematical Society, 2009). <http://www.ams.org/publications/authors/books/postpub/chel-367>
18. Erdős, P. & Rényi, A. On the evolution of random graphs. *Publ. Math. Inst. Hung. Acad. Sci.* **5**, 17-61 (1960). https://www.renyi.hu/~p_erdos/1960-10.pdf
19. Barabasi A. L. Scale-Free Networks: A Decade and Beyond. *Science* **325**, 412-413 (2009). <http://dx.doi.org/10.1126/science.1173299>
20. Barabasi, A. L. & Albert, R. Emergence of scaling in random networks. *Science* **286**, 509-512 (1999). <http://dx.doi.org/10.1126/science.286.5439.509>
21. Caldarelli, G. *Scale-Free Networks: Complex Webs in Nature and Technology* (Oxford Univ. Press, 2007). <http://ukcatalogue.oup.com/product/9780199211517.do>
22. Cowan, N. J., Chastain, E. J., Vilhena, D.A., Freudenberg J. S., Bergstrom, C. T. Nodal dynamics, not degree distributions, determine the structural controllability of complex networks. (2012). <http://dx.doi.org/10.1371/journal.pone.0038398>
23. Jean-Loup Guillaume and Matthieu Latapy. Bipartite Graphs as Models of Complex Networks. *CAAN*, 127-139 (2004). <http://dx.doi.org/10.1016/j.physa.2006.04.047>

24. Boccaletti S, Latorab, V., Morenod, Y., Chavezf, M., D.-U. Hwanga. Complex networks: Structure and dynamics. *Elsevier. Physics Reports* **424**, 175 – 308 (2006). <http://dx.doi.org/10.1016/j.physrep.2005.10.009>
25. Chatterjee, A., Das, D., Naskar, M.K., Pal, N., Mukherjee, A., “Heuristic for maximum matching in directed complex networks” International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2013. Mysore, 22-25 Aug. 2013. <http://dx.doi.org/10.1109/ICACCI.2013.6637339>

Copyright Disclaimer

Copyright reserved by the author(s).

This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/3.0/>).