# Controllability and epidemic spreading

Larry Lam, Guillaume Rostaing, Alexander Schaub
llam268, rostaing, axschaub

***Abstract*—In this paper, we explore control theory as a solution to stop epidemic outbreaks in networks. Control theory deals with the behavior of dynamical systems, specifically networks in this case. We investigate the process of finding sets of nodes required to control a network, and applying an input to these nodes in order to drive the system from a given state to another. We explore the task of minimizing the energy needed to control a specific network. We develop and test algorithms on a subset of a US airports network as well as randomly generated graphs using a SIS model for epidemic spreading.**

## I. Introduction

With people creating and becoming part of increasingly complex and connected networks, finding efficient ways to address potential outbreaks in these networks is of prime importance. Over the past few years, epidemic outbreaks have been monitored with systematic attention, mostly due to highly perilous diseases such as the Ebola hemorrhagic fever or the bird flu. Being able to stop these epidemics (nay pandemics) is tantamount to saving valuable resources such as time, money, and most importantly human lives.
We want to apply control theory to epidemic spreading. Indeed, this phenomenon can be described as a dynamic process, and the control that is applied on the network could be achieved, for example, by vaccinating some specific patients or distributing medicine to cities, depending on the network we consider. By finding the set of driver nodes (i.e. the nodes on which we apply an action to control the system) that minimizes the efforts required to stop the epidemic, we could determine which are the most valuable nodes to target in case of an outbreak.
We work with one dataset in particular, the US airports network. The rationale is that diseases easily spread through the airports network, and distributing medicine to the concerned cities may lower the infection rates. After finding the set of driver nodes, we should be able to completely eradicate the disease by only targeting a definite set of cities.

## II. Prior Work

### A. Finding the Driver Nodes

In (Liu et al., 2011), the authors propose tools to analyze the controllability of arbitrary complex directed networks driven by dynamic processes. The main takeaway from this paper is that a set of $N_d$ driver nodes can be determined, which are the inputs to the command signals. This can be done after ensuring that a network is controllable through Kalman's rank condition.

The authors also detail some behaviors of the $N_d$ driver nodes. First, maximum matching is used to find the minimum set size $N_d$ needed to maintain full control of the network. Also, driver nodes tend to avoid the hubs which appears as a counter-intuitive result. Then, $N_d$ is mainly determined by the in-degrees and out-degrees of the nodes regardless to where the links point. Along these lines, small changes in the average degree distribution induce large variations in $N_d$.

The robustness of the control of the network is analyzed with the definition of critical, redundant and ordinary links. Removing critical links implies increasing $N_d$ while redundant links have no impact on $N_d$. To induce network robustness on link failures, it is sufficient to make the critical links redundant.

### B. Minimizing the Energy to Control a Network

In (Pasqualetti et al., 2014), the authors propose an algorithm to find a subset of control nodes in a network that minimizes the (worst-case) energy required to control the network (that is, to make it reach any given state). With this information, one can go about proving which kinds of networks allow to be controlled with a "reasonable" energy (that is, an energy that does not scale exponentially with the size of the network).

The base model used in the paper is that of a dynamic system in discrete time, controlled by a weighted adjacency matrix A, that can be controlled by injecting a signal u that will affect nodes $(k_1, k_2, ...., k_n)$. Therefore, the network is governed by the equation $x(t+1) = Ax(t) + B * u(t)$, where $B$ is the matrix representing the control nodes ($B = [e_{k_1}, e_{k_2}, ..., e_{k_m}]$)

The network is supposed to be controllable in finite time, therefore, $u(t > T) = 0$. The energy is defined as the square of the time-norm of $u(t)$ (i.e. the sum of the squared norms at each time). This energy is the inverse of the smallest eigenvalue of the so-called Gramian matrix $W = C * C^{\mathsf{T}}$ where $C = \left[B, AB, ..., A^{T-1}B\right]$ (the network is controllable by the nodes defined by $B$ if and only if $C$ has full rank). Other metrics that have been proposed are the trace of the inverse of the Gramian and the Determinant of the Gramian, as well as the trace of the Gramian (although this last metric does not ensure controllability).

### C. *Immunization in Complex Networks*

In (Pastor-Satorras et al., 2002), immunization was explored as a method to prevent epidemic outbreaks in both homogeneous and scale-free networks. Two methods of immunization (uniform and targeted) were tested on the two kinds of complex networks. Uniform immunization, a random introduction of immune individuals into the network, proved to be ineffective for scale-free networks, but had a noticeable effect on the spreading rate in homogeneous networks (reducing spread rate by a factor of 1-g). Targeted immunization, a process in which nodes with the highest connectivity are immunized, performs poorly in homogeneous networks, but works well in the case of scale-free networks.

Seeing that there is no effective strategy for immunization to treat both homogeneous and scale-free networks, we look to controllability as a way to treat networks after an outbreak occurs rather than a purely preventative approach.

## III. Algorithms

### A. *Control energy*

We suppose that the dynamic system we will analyze is described by an equation of the type $x(t+1) = Ax(t) + B * u(t)$ where $A$ and $B$ are square matrices, and $x$ and $u$ are vectors (note that we slightly modify the definition of $B$ and $u$ from the one given by Pasqualetti in the paper and expand $B$ to a square matrix by adding enough empty columns). The goal is to minimize the *control energy* as stated above. However, we would like to minimize the *weighted* control energy (to be able to work on more complex networks with for instance populations on the nodes). Let $\Sigma$ be a symmetric positive-definite matrix. Then the *weighted control energy* is defined as

$$\tilde{E}(u,T) = \sum_{\tau=0}^{T-1} u(\tau)^T \Sigma u(\tau)$$

This is the temporal norm associated with the norm induced by $\Sigma$. When $\Sigma = I_n$, then this is the control energy as defined in this paper.

Let $S$ be the symmetric, positive-definite square root of $\Sigma$. $\tilde{E}(u,T)$ can also be expressed as $E(Su,T)$ since

$$||Su||_2^2 = (Su)^T Su = u^T S^T Su = u^T S^2 u = u^T \Sigma u$$

Now, we can see that the dynamic equation can also be expressed as

$$\begin{aligned} x(t+1) &= Ax(t) + B * S^{-1} * S * u(t) \\ &= Ax(t) + \tilde{B} * \tilde{u}(t) \end{aligned}$$

$$\tilde{W} = C\Sigma^{-1}C^T$$

and the minimum energy is the inverse of the smallest eigenvalue of $\tilde{W}$. Moreover, the associated input would be

$$\tilde{u}(\tau) = S^{-1}B(A^T)^{t-\tau-1}\tilde{W}^{-1}x_f$$

where $x_f$ is the desired final state of the system.

### B. Interpretation of the formula

If $\Sigma$ is simply a diagonal matrix with positive terms, $\Sigma = diag(\sigma_i^2)$, then $S^{-1}B$ is simply a matrix where the canonical vectors $e_i$ are replaced with $\frac{e_i}{\sigma_i}$. Therefore, in order to have the same effect on the system, the $i-th$ component has to be multiplied by $\sigma_i$, thus inducing a larger norm (if $\sigma_i > 1$).
The non-diagonal case is maybe more complex to apprehend. It would make sense to use non-diagonal weight matrices if the energy would not only depend on the $u_i(t)^2$ but also on the squares of linear combinations of the $u_i(t)$. Since any symmetric positive-definite matrix can be diagonalized, the value of $u(t)^T\Sigma u(t)$ can always be expressed as a linear combination of the squares of linear combinations of the $u_i$.

### C. Epidemic spreading

For the SIS-model, the (simplified) equation describing the infection probability $p_i$ (valid for the beginning of the outbreak with small values of $p_i$) of node $i$ is given by

$$\dot{(p_i)} = -\alpha_i p_i + \beta_i \sum_{j\in N_i} a_{ij} p_j$$

where $A = [a_{ij}]$ is the adjacency matrix, $\alpha_i$ and $\beta_i$ are the curing and infection rates.

This equation can be discretized : if $h$ is some small parameter, then we have that

$$\frac{p_i(t+h) - p_i(t)}{h} \simeq -\alpha_i p_i + \beta_i \sum_{j\in N_i} a_{ij} p_j$$
$$\implies p_i(t+h) \simeq (1 - h\alpha_i) p_i(t) + h\beta_i \sum_{j\in N_i} a_{ij} p_j$$

Finally, if we discretize the time parameter and only consider time steps of value $h$, and furthermore rewrite this equation in terms of the whole vector $\mathbf{p}$, the we have that

$$\mathbf{p}(t+1) = (I_n - h\alpha) + h\beta A$$

where $I_n$ is the $n \times n$ identity matrix, and $\alpha, \beta$ are diagonal matrices with values $\alpha_i$ and $\beta_i$respectively.

### D. Choosing the driver nodes

The aforementioned formulas enable us to compute the minimal control energy for given set of driver nodes. Now, the objective is to find the set of driver nodes that minimize the minimum control energy. The paper by Pasqualetti et al. proposes a method to choose a set of driver nodes in a decentralized method by clustering the network and choosing the driver nodes among the boundary nodes of the clusters, and trying to minimize the the minimum input energy of each cluster. The idea is to partition the network using Fiedler bisections, to select the boundary nodes (i.e. the nodes that have a neighbor node that is not in their cluster), and bisect the least controllable cluster until enough driver nodes have been selected.

While this won't find the set of control nodes that will minimize the minimum control energy, it could yield interesting results. Moreover, this method is quite fast.

A second, more straightforward way to select the nodes would be to use a greedy approach. While this works quite well for maximizing the rank of the matrix and several metrics [8], it is not sure whether it produces a good result on the metric we choose, the inverse of the smallest eigenvalue of the Gramian.

Our algorithm works as follows : at each round, we select one more node until the number of desired driver nodes has been found. Let $l_i$ the list of $i$ previously selected nodes. For every node $j \notin l_i$, we compare the rank and the energy corresponding to the set of driver nodes $l_i \cup j$. We select $j$ such that $l_i \cup j$ maximizes the rank and minimizes the energy.

## IV. Experiments

### A. Implementation

We implemented the algorithm proposed by Pasqualetti et al.. However, computing the Gramian (weighted or not) proved to be more complicated than expected.

We implemented the algorithm in Python, using the numpy module. However, for the airport network (about 650 nodes), the values of the elements in the Gramian overflow when using standard 64-bits integer representation. Neither Matlab/Octave or numpy can handle arbitrary precision floating point numbers. The implementation using another Python module, named mpmath, which is able to handle higher-precision floats, is too slow to be usable. We will therefore re-run our algorithms on a smaller part of the airport network and try to find a faster algorithm using mpmath, especially for computing the smallest eigenvalue of a matrix.

Finally, we decided that we would keep working with numpy but only consider a smaller network than the initial 600-nodes network to avoid overflowing problems. Therefore, there might be rounding errors due to the finite precision we were working on.

## *B. Networks*

*1) Generic graphs:* First, we decide to generate random graphs. We use both the geometric preferential attachment model (Flaxman, Frieze and Vera, 2006) and the Erdos-Renyi random graph model. The generated graphs are undirected, the edges are not weighted and we weight the nodes using the degree measure.

*2) Airport network:* To experiment the algorithms, we created a US airport network obtained from data of the Bureau of Transportation Statistics. The aim of the network is to provide a realistic framework concerning epidemic spreading. The potential action on the nodes could be providing drugs to cure the population.

The raw data contains all the trips between any two American airports aggregated by airline and month, along with the number of passengers on this specific route for this specific airline. We solely focus on the year 2014, and further aggregate the data to get the total number of passengers per year flying between any two airports. We normalize this figure to a daily count and discard routes with less than 500 passengers a day on average.

This operation enables us to get a lean network of 105 nodes and 1052 weighted edges (compared to the initial network of more than 600 nodes that was created discarding routes with less than 1 passenger a day). We add weights on the nodes, which are just the weighted out-degrees. We assume this represents a rough estimation of the population or importance of a particular node. To account for potential (yet the probability is quite low) nodes with no outgoing edges, we set a minimal weight at an arbitrary nonzero positive value.

At this point the network is fully weighted and can be fed to our controllability algorithms. We decide to generate a visualization of the network by getting the GPS coordinates of all the airports in the dataset. All the airports of the network are displayed on a US map along with the routes.

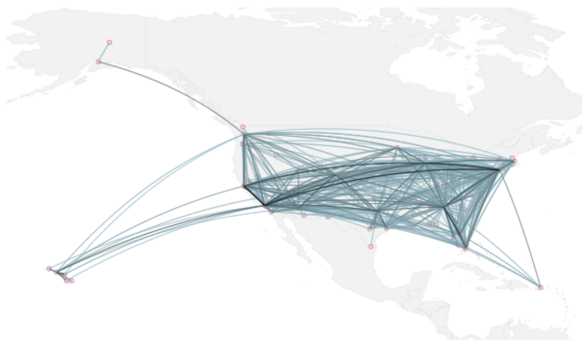


Fig. 1. US airports network

## *C. Results*

For the parameters $\alpha_i$ and $\beta_i$, we generated uniform random vectors. The initial state of the infection was also a uniform random vector. As for the weights, we simply chose the node degrees. The horizon has been set to be equal to the number of nodes.

We finally selected the driver nodes using the greedy approach. The problem with the partitioning approach was that the Fiedler partitioning was unable to separate the graph (all nodes are put in the same partition). The greedy approach would therefore work better.

*1) Generic graphs:* As we can see, the control energy decreases when the number of driver nodes increases. However, for most values of the number of driver nodes, the energy is far too

Fig. 2. Control energy and number of driver nodes for a random 50-nodes Erdos-Renyi graph

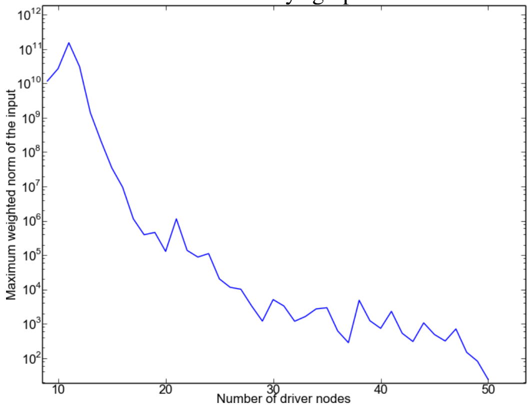
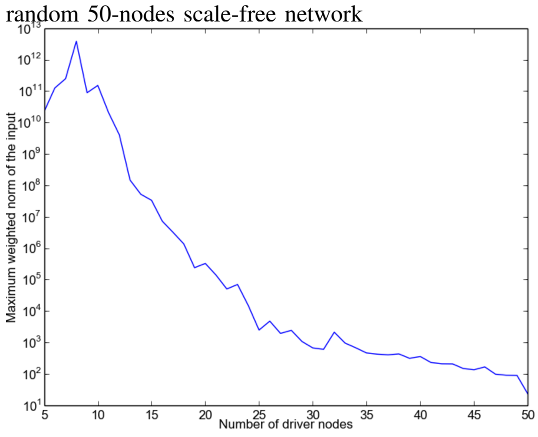


Fig. 3. Control energy and number of driver nodes for a random 50-nodes scale-free network



high to even make sense : the resulting input $u$ takes values that are much higher than 1, and the network would have to go through negative states, which does not make sense in terms of infection rates. The problem is that it is not possible to compute the optimal control input with the constraints in our network ($0 \leq p_i(t) \leq 1$), because the problem would become non linear in this case. Also, we noticed that, when the number of control nodes is small but the network is supposed to be controllable, the computed input does not seem to correctly drive the network towards the desired state, even if we ignore the constraints on the values of $p_i$. This could, however, be caused by rounding errors, the limited precision of floats, or errors in the rank computation (which is also related to the limited precision of floating point numbers).

Fig. 4. Control energy and number of driver nodes for a random 50-nodes Erdos-Renyi graph

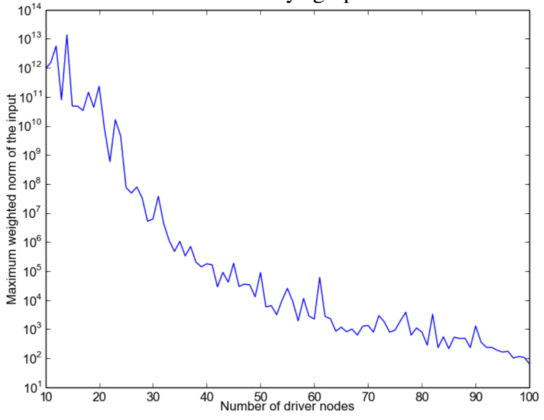
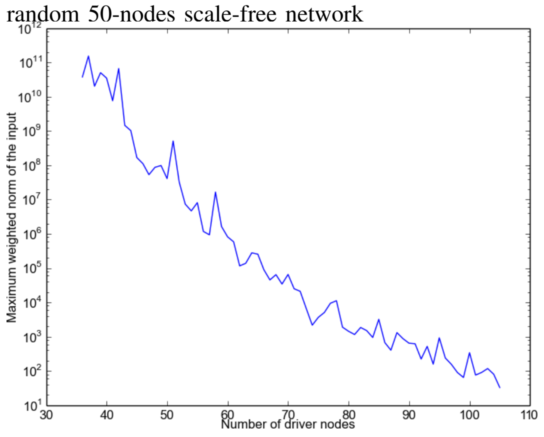


Fig. 5. Control energy and number of driver nodes for a random 50-nodes scale-free network



*2) Airport network:* We plotted the control energy for the pruned airport network (105 nodes) as well as for a scale-free network of similar size (100 nodes). The first thing to note is that the airport network is less "controllable" than the scale-free network, in the sense that it is necessary to select many more control nodes before the Gramian has full rank. Secondly, similar to the previous case, the control energy remains far too high when less than all of the nodes are selected to be driver nodes, making it non-viable to control the epidemic spreading by such means.

## V. Conclusion and Future Work

Having run our controllability algorithms on both generic graphs and the sample airport network, we have seen that controllabililty does not perform well for the purpose of treating epidemic

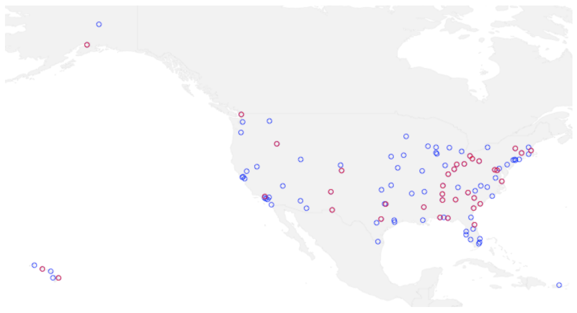


Fig. 6. US airports network, driver nodes are in red. Edges are not represented for clear visibility. This set is the minimum set so that the network is controllable.

outbreaks. What we did not investigate were some edge cases on which the controllability approach could work: for example, when the $\alpha_i$ and $\beta_i$ are in such a relationship that the epidemic would almost stop naturally, or for a smaller initial distribution of the infection probabilities.

Although controllability did not shine in this particular application, there are other fields and opportunities to experiment with the controllability approach. Another area to work on would be fixing the issues regarding floating point numbers in order to run our algorithms on larger networks.

## References

[1] Sepehr Assadi, Sanjeev Khanna, Yang Li, and Victor M Preciado. Complexity of the minimum input selection problem for structural controllability. *IFAC-PapersOnLine*, 48(22):70–75, 2015.

[2] Abraham D Flaxman, Alan M Frieze, and Juan Vera. A geometric preferential attachment model of networks. *Internet Mathematics*, 3(2):187–205, 2006.

[3] Yang-Yu Liu and Albert-Laszló Barabási. Control principles of complex networks. *arXiv preprint arXiv:1508.05384*, 2015.

[4] Yang-Yu Liu, Jean-Jacques Slotine, and Albert-László Barabási. Controllability of complex networks. *Nature*, 473(7346):167–173, 2011.

[5] Alex Olshevsky. Minimal controllability problems. *Control of Network Systems, IEEE Transactions on*, 1(3):249–258, 2014.

[6] Fabio Pasqualetti, Sandro Zampieri, and Francesco Bullo. Controllability metrics, limitations and algorithms for complex networks. *Control of Network Systems, IEEE Transactions on*, 1(1):40–52, 2014.

[7] Romualdo Pastor-Satorras and Alessandro Vespignani. Immunization of complex networks. *Physical Review E*, 65(3):036104, 2002.

[8] Tyler H Summers, Fabrizio L Cortesi, and John Lygeros. On submodularity and controllability in complex dynamical networks. *arXiv preprint arXiv:1404.7665*, 2014.

[9] Piet Van Mieghem, Jasmina Omic, and Robert Kooij. Virus spread in networks. *Networking, IEEE/ACM Transactions on*, 17(1):1–14, 2009.

[10] Zhengzhong Yuan, Chen Zhao, Zengru Di, Wen-Xu Wang, and Ying-Cheng Lai. Exact controllability of complex networks. *Nature communications*, 4, 2013.