Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Lecture with Computer Exercises:
Modelling and Simulating Social Systems with MATLAB

Project Report

Optimal Strategy for Disaster Spreading

Guangyu Li, Jiawen Luo, Zhichao Han

ETH Zurich
December 2016

Agreement for free-download

We hereby agree to make our source code for this project freely available for download from the web pages of the SOMS chair. Furthermore, we assure that all source code is written by ourselves and is not violating any copyright restrictions.

Name 1 Jiawen Luo Name 2 Zhichao Han Name 3 Guangyu Li

Contents

1 Abstract

Efficient response and decision making are very important when disaster happens. In order to control disaster spreading, several heuristic strategies have been proposed to distribute external resources optimally [1]. We examine these heuristic strategies on different type of networks. Moreover, we model this problem as a PDE-constrained optimization problem, which is solvable by the adjoint method when topology properties of the network is given a-prior. Solution of this optimization problem gives an optimal strategy for resource distribution with respect to a certain objective function. Results show that the converged optimal strategy is much more effective than these existed heuristic strategies.

2 Individual contributions

- **Modeling:** we read paper and find out what to do together. Particularly, Jiawen proposed the adjoint method and derivated the formula used for optimization.
- **Implementing:** Jiawen and Zhichao implement the simulation of disaster spreading independently. After validing the correctness of our code, Jiawen implement the codes solving optimization problem.
- **Writing:** we write the report together.

3 Introduction and Motivations

Disastrous events spread in networks and might lead to total breakdown of the system, for example, road infrastructures, power system and social networks. Emergency response and recovery call for external resources are crucial for survival and determine whether the affected areas will overcome the consequences of catastrophes or not [1]. The usual situation is that we have limited resources and we need to make a decision on the resource distribution. A good strategy is very important for overcoming a disaster and may prevent us from enormous economic loss.

In this work, we study the problem of effectively utilizing limited external resources to control disaster spreading in a network. In the previous work of Buzan, et. al [1], they compare six heuristic strategies to distribute resources when disaster happens and analyze the “best” one among these six strategies for different scenarios. However, there exist lots of other heuristic strategies since the ways to distribute resources are infinite. So a natural question is: what is the optimal strategy? To our knowledge, no previous work has answered this question.

We find this problem could be modeled as a PDE-constrained optimization problem given the network topology. We proposed two different objective functions for different optimal criteria. By using adjoint method to solve this optimization problem, we finally obtain the optimal way to distribute resources when disaster happens. Numerical results show that the optimal strategy is much more effective than these heuristic strategies studied in [1].

This report is organized as the following way: We firstly introduce the disaster spreading model in Sec. 4. In Sec. 5, we discuss the heuristic strategies proposed in [1] and formulate our optimization problem. After this, in Sec. 6 we introduce the adjoint method in a slightly general way to avoid complex settings in our model. Sec. 7 is our experiments design, followed by results in Sec. 8. We propose some interesting work that can be done in further studies in Sec. 9. We present the adjoint system of our model in Appendix I and explain our code structure in Appendix II.

4 Disaster Spreading Model Description

Generally speaking, a network system contains many interconnected components. Each component may influence / be influenced by other components and each component has a recovery rate. The recovery rate is related with the external resources assiged to this component. Different strategies have different ways to assign external resources, thus leading to different results. Before we discuss these strategies, let us first specify the model of disaster spreading and the influence of external resources.

4.1 the dynamic model of disaster spreading

Our model of disaster spreading is slightly modified from the model proposed in [1]. The system is modeled as a directed graph $G = (V, E)$. Each node represents a component in this system and there is a directed edge from node i to node j if component i has influence to component j. For node $i \in \{1, 2, \dots, N\}$, its state at time t is represented by a continuous function $x_i(t) \in \mathbb{R}^+$. When node i is in normal state, the corresponding state variable x_i is 0, while if node i is deviated from the normal state, its corresponding state variable $x_i > 0$.

The whole system exhibits a natural level of resistance to challenges, which is reflected by the threshold $\theta_i \geq 0$ of node i. We say node i is failed when $x_i \geq \theta_i$ and node i is challenged but not failed when $0 < x_i < \theta_i$. So the state of node i could be divided into these three cases:

$$\text{node } i \text{ is:} \begin{cases} \text{normal} & \text{if } x_i = 0 \\ \text{challenged} & \text{if } 0 < x_i < \theta_i \\ \text{failed} & \text{if } \theta_i \leq x_i \end{cases} \tag{1}$$

The interactions between components are spreaded via edges, which is characterized by connection strength M_{ij} and transmission delay t_{ij}. The dynamic of node i is modeled by equation:

$$\frac{dx_i}{dt} = -\frac{x_i}{\tau_i} + \Theta_i(\sum_{i \neq j} \frac{M_{ji} x_j(t - t_{ji})}{f(O_j)} e^{-\beta t_{ji}}) \tag{2}$$

where the first term in right-hand-side models the self-recovery ability of node i and τ_i, known as the recovery time, is related to the resources distributed to node i. We will discribe τ_i in detail afterwards. The second term characterizes the influence of other nodes to node i. $M_{ij} = 0.5$ if there is an edge from node i to node j and $\beta = 0.025$, as is used in [1]. The time delay t_{ij} obeys a χ^2 distribution, with degree freedom $\mu = 4$. And the distribution is strectched by multiplying a factor 0.05 and shifted with 1.2. Finally the average delay $\langle t_{ij} \rangle = 1.4$. The function $f(O_j) = (aO_j)/(1 + bO_j)$ is used to reflect that the impact of highly connnected neighboring nodes is "dissipated" among all out-edges. O_j is the out-degree of node j. a is set to 4 and b is set to 3. $\Theta(\cdot)$ is the coupling function in sigmoid form:

$$\Theta_i(y) = \frac{1 - \exp(-\alpha y)}{1 + \exp(-\alpha(y - \theta_i))} \tag{3}$$

where α is set to 20 in our report.

We follow all the parameter settings in the original paper [1], except for the time delay t_{ij}. Instead we set all t_{ij} equal to $\langle t_{ij} \rangle = 1.4$.

4.2 self-recovery and external resources

In 4.1, Eq. (2) characters how disaster spreads in a system. The recovery rate $1/\tau_i$ is defined as:

$$\frac{1}{\tau_i} = \frac{1}{(\tau_{start} - \beta_2)e^{-\alpha_2 R_i(t)} + \beta_2} \tag{4}$$

where $\tau_{start} = 4$, $\alpha_2 = 0.58$ and $\beta_2 = 0.2$ are model parameters. Based on the assumption of [1], resources will remain at a certain node and will not be assigned again once they have been assignment to some node. $R_i(t)$ is the cumulative resources of node i at time t.

Eq. (2) and (4) reflect that external resources could be reduce disaster spreading by increasing the recovery rate of its corresponding node. If no external resources are assigned to node i, it can recover from derivation with its internal recovery rate $1/\tau_{start} = 0.25$. Otherwise, the recovery rate of node i (Eq. (4)) increases if more resources are assigned to node i. Furthermore, the maximal recovery rate is limited by $1/\beta_2 = 5$.

Now, we define the mobilization of external resources. According to [1], the amount of resources reached into the system by time t is modeled by the shape of a continuous function $r(t) = a_1 t^{b_1} e^{-c_1 t}$. $a_1 = 530$, $b_1 = 1.6$ and $c_1 = 0.22$ are constant parameters. The mobilization then corresponds to the growing part of $r(t)$. The curve is furthermore normalized to fit the total external resources and simulation time horizon. Details will be discussed in section 7.1.

5 Different Strategies

In this section, we review these six heuristic strategies studied in [1] and discuss how to model the disaster spreading as an optimization problem.

5.1 Some heuristic strategies to control disaster

In [1], the authors examine these six heuristic stratеties to distribute external resources:

S1 *uniform dissemination:* each node gets the same amount of resources

S2 *out-degree based dissemination:* the resources are distributed over nodes proportionally to their out-degrees

S3 *uniform reinforcement of challenged nodes:* all nodes $i \in \{1, 2, \ldots, N\}$ with $x_i > 0$ are equally provided with resources

S4 *simple targeted reinforcement of destroyed nodes:* damaged nodes are equally provided with resources with priority, while challenged nodes are uniformly reinforced if no damaged nodes exist

S5 *simple targeted reinforcement of highly connected nodes:* a fraction q of highly connected nodes is uniformly provided with resources by using the fraction k of all resources, while the remaining resources are applied according to strategy **S4**

S6 *out-degree-based targeted reinforcement of destroyed nodes:* application of strategy **S4**, but with a distribution of resources proportional to the out-degrees of nodes rather than a uniform distribution

Among these six strategies, **S6** seems to be the most efficient one in many situations[1], but it is almost for sure that there must be some optimal strategy that will have a better performance than **S6**. In the following subsection, we discuss how to get the optimal strategy from optimization.

5.2 Disaster spreading as an optimization problem

We formulate this question as an optimization problem to find the optimal strategy. Here we consider two optimization goals. The first one is of the same consideration of the original paper [1]: minimize the number of damaged nodes at end time $t = T$. The other one is to minimize the averaging status of all nodes. As one will see in section 8, these two objectives lead to very different spreading processes. The first principle tends to control the total number of damaged nodes, e.g., a node with status value of 0.6 and a node with 4.0 are both regarded as damaged nodes and get same penalty. This may lead to the situation that total number of damaged nodes is in control while some nodes are in very bad status. On the contrary, the second strategy will lead to a more averaging status of all nodes, as showned in Fig. 4 and Fig.7.

6 Adjoint Method for Optimization

As mentioned in the last section, our goal is to optimize the resource distribution strategy. In this section we introduce the adjoint method in a general setting. The detailed derivation of the adjoint system corresponding to our case will be explained in the Appendix 10. Adjoint method is widely used in many areas, for example, data assimilation [3], full wave inversion in seismology [4] and PDE constrained optimization [2]. The main motivation for utilizing the adjoint method is that the parameters we want to optimize live in very high dimensions. Calculating derivatives of the objective function w.r.t. to N parameters in a finite difference scheme will require N times forward simulations, while the adjoint method will only require one forward and backward simulation, so in general it will reduce a lot of computation time.

Suppose our dynamical system is governed by a partial differential equation,

$$\frac{\partial \mathbf{x}}{\partial t} = \mathscr{T}\mathbf{x}. \tag{5}$$

$\mathbf{x}$ lives in a Hilbert space $\mathscr{H}$ with inner product defined as $\langle\cdot,\cdot\rangle$. To make our discussion simple, let $\mathscr{T}$ be a bounded linear operator: $\mathscr{H} \to \mathscr{H}$. By Riesz representation theorem the adjoint operator of $\mathscr{T}$ exists and is unique, noted as $\mathscr{T}^\dagger$. These two operators have the property,

$$\langle \mathscr{T}\mathbf{x}, \mathbf{y}\rangle = \langle \mathbf{x}, \mathscr{T}^\dagger \mathbf{y}\rangle, \quad \mathbf{x}, \mathbf{y} \in \mathscr{H}. \tag{6}$$

Let our objective function be a scalar function $f(\mathbf{x}_T)$ and the parameter to optimize be $r(t)$ satisfying constraint $g(r) \equiv 0$. g is also a scalar function for simplicity. The objective function value is uniquely controlled by parameter $r(t)$. Note that the operator $\mathscr{T}$ may also depend on $r(t)$. Now we will show how to derive the adjoint system and formulate our algorithm.

Define the Lagrangian as

$$\mathscr{L} = f(\mathbf{x}_T) + \int_0^T \langle \mathbf{x}^\dagger, \frac{\partial \mathbf{x}}{\partial t} - \mathscr{T}\mathbf{x}\rangle dt + \lambda(t) g(r), \tag{7}$$

in which $\mathbf{x}^\dagger$ and $\lambda(t)$ are Lagrangian multipliers and $\mathbf{x}^\dagger$ is also know as the adjoint field of $\mathbf{x}$. T is the time horizon we will consider. Take the variation of $\mathscr{L}$,

$$\delta\mathscr{L} = \langle \nabla_{\mathbf{x}_T}\mathscr{L}, \delta\mathbf{x}_T\rangle + \langle \nabla_{\mathbf{x}^\dagger}\mathscr{L}, \delta\mathbf{x}^\dagger\rangle + \langle \nabla_{\mathbf{x}}\mathscr{L}, \delta\mathbf{x}\rangle + \langle \nabla_{\lambda}\mathscr{L}, \delta\lambda\rangle + \langle \nabla_{r}\mathscr{L}, \delta r\rangle. \tag{8}$$

Integration by parts will give us,

$$\nabla_{\mathbf{x}_T}\mathscr{L} = \nabla f(\mathbf{x}_T) + \mathbf{x}_T^\dagger = 0, \tag{9}$$

$$\nabla_{\mathbf{x}}\mathscr{L} = -\frac{\partial \mathbf{x}^\dagger}{\partial t} - \mathscr{T}^\dagger \mathbf{x}^\dagger = 0, \tag{10}$$

$$\nabla_r \mathscr{L} = \int_0^T \langle \mathbf{x}^\dagger, -\nabla_{r(t)}(\mathscr{T}\mathbf{x})\rangle dt + \lambda(t)\nabla_{r(t)} g(r(t)). \tag{11}$$

Eq. (9) is known as the compatibility condition, which determines the initial (or called the terminal condition in some literatures) of the adjoint field $\mathbf{x}^\dagger$. Eq. (10) is the so called adjoint equation which controls the evolution of $\mathbf{x}^\dagger$. Eq. (11) gives the gradient of $\mathscr{L}$ w.r.t. our parameter $r(t)$ and we will update $r(t)$ using this information. Note that the term $\mathbf{x}^\dagger\delta\mathbf{x}|_{t=0}$ vanishes due to our assumption that the initial condition is a fixed parameter.

With all these formulas, we are able to construct our optimization algorithm as:

1. Perform one forward simulation of Eq. (5);
2. Calculate $\mathbf{x}_T^\dagger$ according to Eq. (9);

3. Perform once backward simulation according to Eq. (10);
4. Calculate $\nabla_r \mathscr{L}$ from Eq. (11);
5. Update $r(t)$ as $r(t) \leftarrow r(t) + \Delta \nabla_r \mathscr{L}$, Δ is step size for updating;
6. If not converged, return to the first step.

Note that $\lambda(t)$ will be calculated from the constraint $g(r) = 0$.

7 Implementation

Here, we explain some practical details in designing the experiments.

7.1 Practical details in model implementation

The forward and adjoint equations are solved in a finite difference scheme using Euler's method. The time horizon T is set to be 100 and $\Delta t = 0.01$. To save computation time, we assume that resources will only arrive at $t = 1, 2, 3, \ldots$, which makes $R(t)$ a piecewise function. The shape of $R(t)$ is shown in Fig. 1. Resource will only arrive after the reaction time t_D. Tests have been done during our implementation, and justify that a coarser discretization of resource mobilization has negligible influence on the qualitative properties of the spreading process. In our implementation, we fix time delay t_{ij} to 1.4 instead of a random variable. The purpose of this setting is to speed up computation. In fact, this is not an important factor for the spreading process model in [1] and we can save a lot of computation time with this simplification.

7.2 Experiments setup

In this report, we first benchmarked our code with results in [1]. Then we studied effects of the choice of sigmoid coupling function in spreading model (Eq. (3)) and replaced it with a linear function. Results showed that these two setups will lead to very different behaviors of the process. A sigmoid function with a large α prompts an easier spreading of disaster.

After reproducing the results of original paper, we evaluate our optimization model. We choose two different objective functions (the first term in Eq. (19)): $\frac{1}{n}\sum_{i=1}^{n} x_T^{(i)}$ and $\sum_{i=1}^{n} \Theta(x_T^{(i)})$. $\Theta(\cdot)$ is the same sigmoidal function as defined in Eq. (3). The first choice comes from that we want to minimize the averaging damage level of all nodes, while the second choice will lead to a minimization of total number of damaged nodes at time T. Detailed comparison of these two setups will be in Section 8.

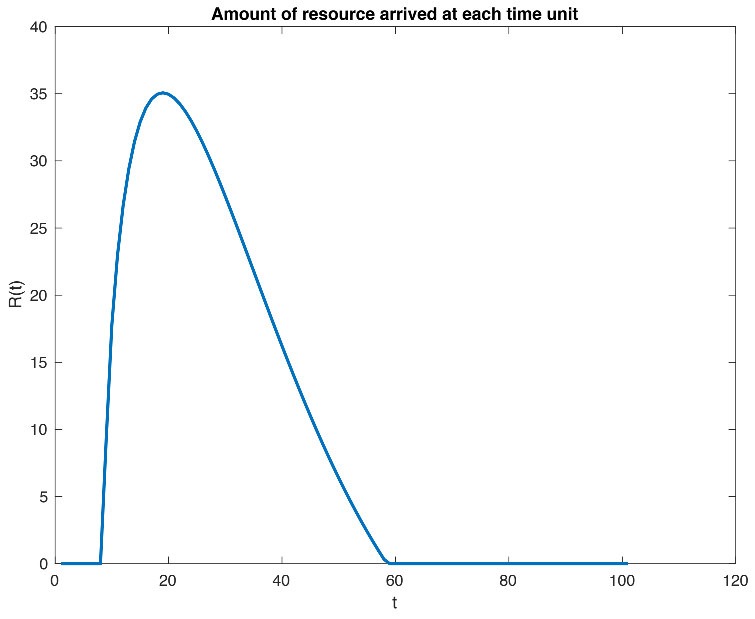


Figure 1: Resource arriving at each time unit. $t_D = 8$ and total resource is set to 1000.

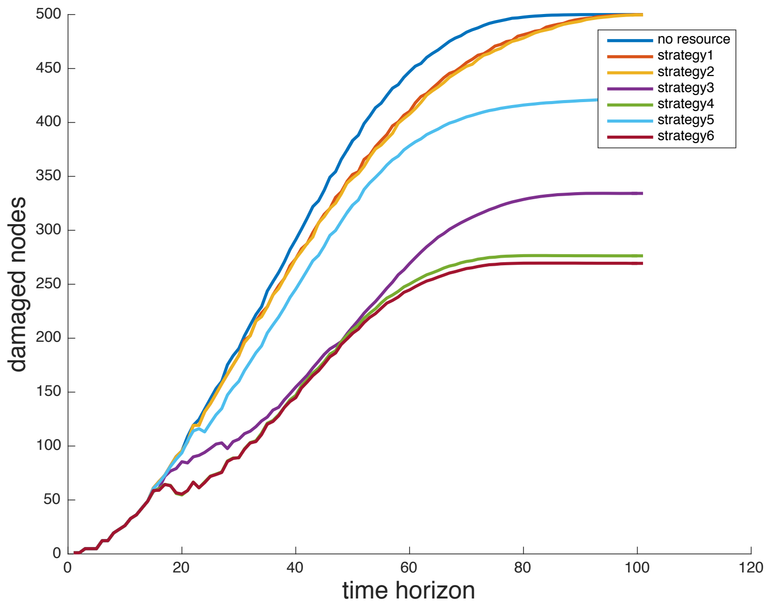
8 Simulation Results and Discussion

For the numerical experiments, we first run the same experiments of [1] and reproduce *figure 2* in original paper[1]. Then we compare the optimal strategy from our optimization with these six heuristic strategies.

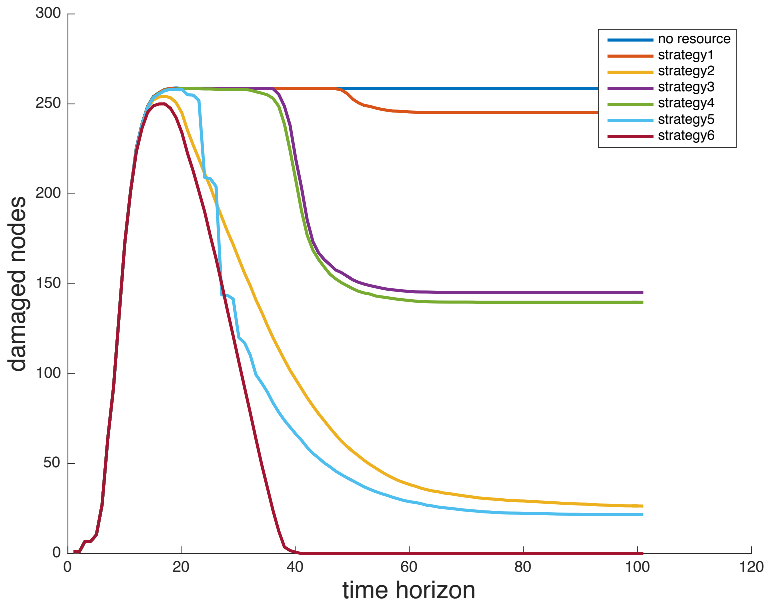
8.1 Reproducing Experiments of Original Paper

In order to benchmark our model and codes, we first perform the same experiments as in the original paper[1] and reproduce *figure 2* in [1].

One can see that our result (FIG. 2) is almost the same with that in original paper[1]. During the process of reproducing these results, we have several observations on this disaster spreading model. The most important one is that the spreading process is highly dependent on the choice of initial breaking point. For example in a grid network, a starting node at the center of network will lead to a much more severely damaged final state compared to a starting node at the corner. Due to this reason, we would argue that the averaging among random starting node is meaningless, although we still use this this kind of methodology for comparison in this report. We would suggest to discuss the effect of initial damaged node's location on the spreading of disasters in future research. We also notice that in both grid and scale-free networks, S6 has the best performance. This is easy to understand because



(a) Heuristic strategies on GRID network



(b) Heuristic strategies on ScaleFree network

Figure 2: Average number of damaged nodes in (a) grid network and (b) scale-free network, with different heuristic strategis. Total external resources is 1000 and $t_D = 8$. The initial damaged node is chosen randomly and these curves is average of 50 experiments.

to conduct S6, we require more information than in other strategies and distribute resources with consideration of each node's status. From this observation, we can infer that more detailed information of the network topology and nodes' status will admit a better strategy. In another word, we can find an optimal strategy is we have full information on the network topology and nodes' properties. And this will be main result of the this report.

8.2 The Optimal Strategy

We have introduced our optimization method in Sec. 6, and we will show our results in this section. Details about setups of our model will be shown in Appendix 10 and 11.

8.2.1 Minimize The Number of Damaged Nodes

As anticipated in section 5, we originally want to minimize the number of damaged nodes after a certain time window T. This is also the base idea used in [1]. This choice is very natural because we want to control the number of damaged nodes, i.e. in our model the number of nodes $x_T^{(i)} > \theta_i$ after spreading of a disaster. Intuitively an objective function for this purpose is the $0-1$ loss function:

$$\begin{aligned} J &= \sum_i h(x_i) \\ \text{where } h(\cdot) &= \begin{cases} 1 & \text{if } x_i \geq \theta_i \\ 0 & \text{otherwise} \end{cases} \end{aligned} \tag{12}$$

However, discontinuity of this function will be a problem when we take variance of the Lagrangian. To solve this we introduce another objective function to approximate the $0-1$ loss function.

$$\begin{aligned} J &= \sum_i \Phi_i(x_i) \\ \text{where } \Phi_i(y) &= \frac{1-\exp(-\alpha y)}{1+\exp(-\alpha(y-\theta_i))} \end{aligned} \tag{13}$$

Sigmoid functions with respect to different values of α are shown in Fig. 3. We find that a sigmoid function with α set to 20 can approximate the $0-1$ loss function very well. So we set $\alpha = 20$ in our model. By this function, we expect that optimized strategy will give a lower number of damage nodes than any other strategies. Fig. 4(a) shows the evolution of number of damaged nodes with respect to different strategies proposed in [1] and the optimized solution on a grid network.

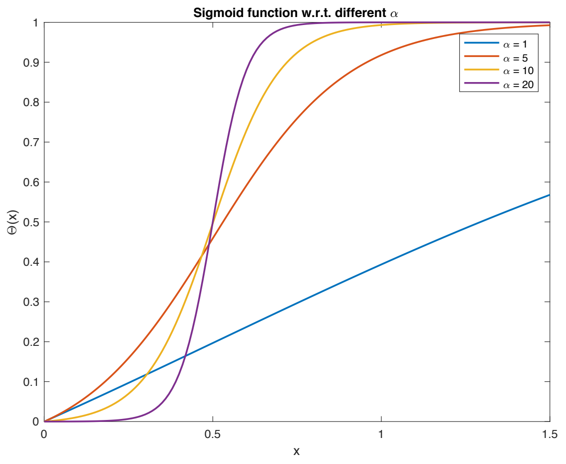


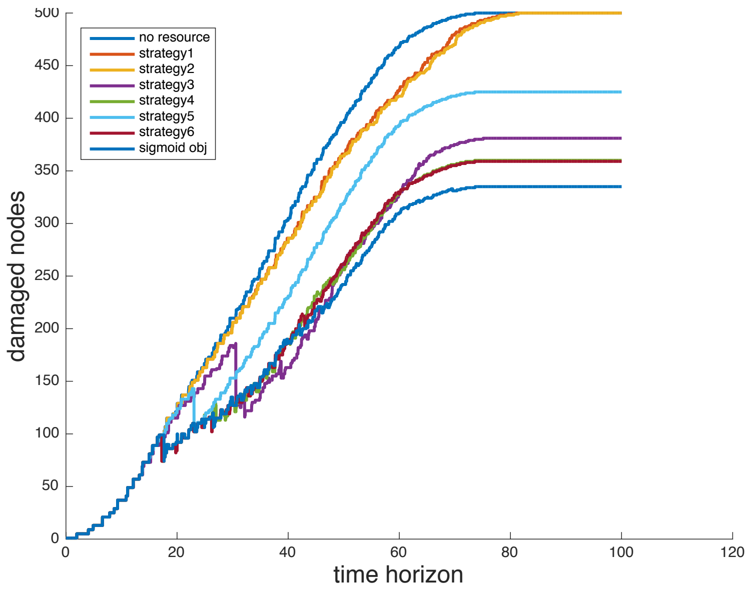
Figure 3: Shape of sigmoid function w.r.t. different α. θ is set to 0.5. When $\alpha = 20$, this function gives a very good approximation of the $0 - 1$ step function.

The same result on a scale-free network is shown in Fig. 7(a). It is clear that the optimized solution is better than any other heuristic strategies. It is worth to mention that, due to nature of the chosen objective function, a random starting resources distribution strategy will not be a good choice. This is because almost all nodes are above θ_i and locate and the right flat part of sigmoid function, so the gradient is almost zero here. However we have a natural choice of starting model, say **S6** in a grid network. Although this choice makes optimization numerically possible, we are still faced with the problem of very small gradient and many local minimums. There is no guarantee that this optimal solution in our plot is a global minimum and we believe that this solution can still be improved by using more suitable numerical techniques.

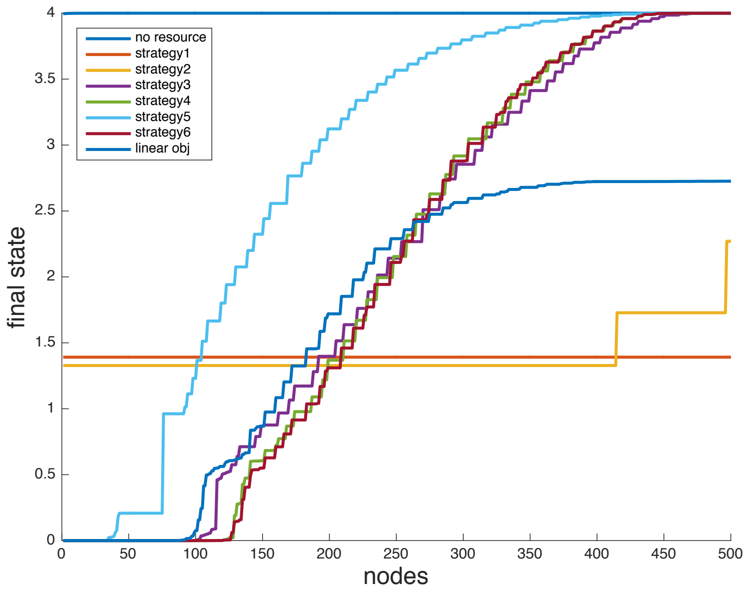
8.2.2 Optimize Final States

In Fig. 4(b) and 7(b) we plot the final state of each nodes in an ascending way. One can see that although the number of damaged nodes is controlled by those strategies such as **S6**, some nodes persist very high status value. In real world, it would imply that those components of the network is badly damaged for a very long time. One can argue that this kind of situation is not acceptable and want to avoid those very badly damaged nodes. This is exactly the starting point of our choice of a second objective function as:

$$J = \frac{1}{n}\sum_{i} x_T^{(i)} \tag{14}$$



(a) Damaged nodes on grid network



(b) Final states on grid network

Figure 4: (a) Number of damaged nodes and (b) final states on grid network. Total external resources is 1000 and $t_D = 8$. The initial damaged node is chosen randomly. The maximal iterations of gradient descent is 100.

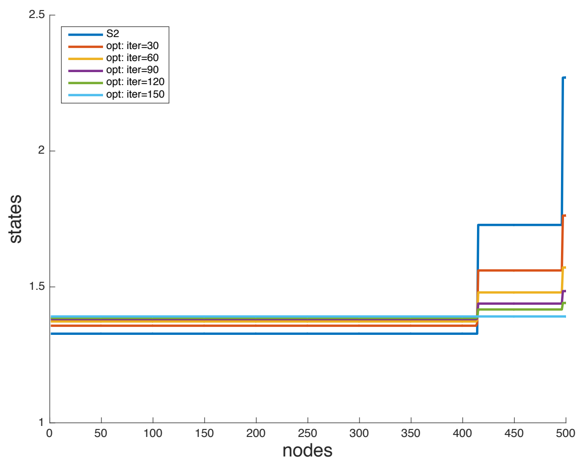


Figure 5: States of optimal strategy at different time, optimized from **S2**. We can find that the optimal solution converges to **S1**

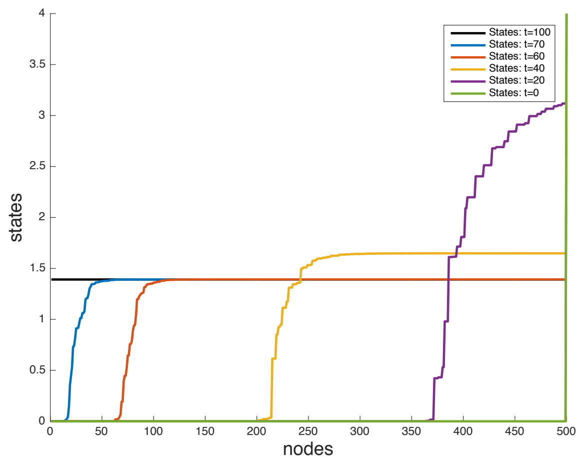


Figure 6: Nodes' states of **S1** at different time steps. We find the whole system comes to a stable state.

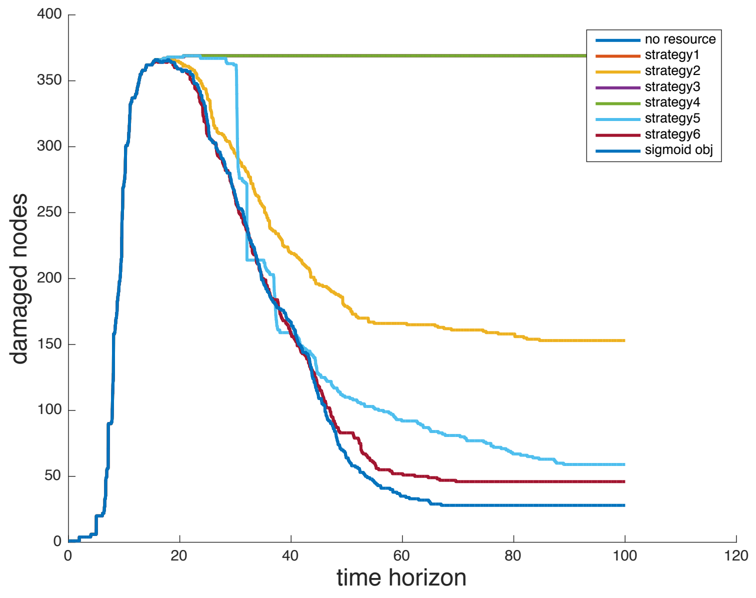
Eq. (14) aims at optimizing the average final state of all nodes. It exclude the situation that some nodes are severely damaged. The optimized strategies on grid network and scale-free network are shown in Fig. 4(b) and Fig. 7(b) respectively. On scale-free network our optimal gives a better strategy compared to all other ones, while on a grid network one may see that the optimal solution is exactly the one comes from S1. We would interpret this as that $T = 100$ is long enough for S1 and S2 to reach a stable state, and in this case S1 is indeed the best one. We can explain this qualitatively. The time window is long enough and the system arrive at the stable state and due to the nature of grid-network, the local topology of each node is almost the same. So we would speculate that when arriving at a stable state, all nodes must have similar status. However if performing some simple analysis on the definition of recover rate, one would find that distribute resources equally would be the best solution under those assumptions above. We also proved this numerically. In Fig. 5, we show that if we start our optimization from S2, we arrive at exactly the result of S1. Also, to show that results from S1 is a stable solution, we show the evolution of nodes' status at several time steps in Fig. 6. One can also see that the optimal solution in this case might also be unsatisfactory. This objective function will more likely lead to averaging state of all nodes. When total resource is very limited, this strategy will result that almost all nodes are damaged, although not in a very bad way.

So we can conclude from our optimization that the more we know about the network system, the better we can distribute the resources. In general, the best strategy depends on the objective we want to achieve, especially when resource is limited. Although objective functions may differ, our observation is that decision makers need to identify those highly risky nodes and distribute suitable amount of resources in advance to prevent the spreading of disasters, rather than after a certain node is damaged already.

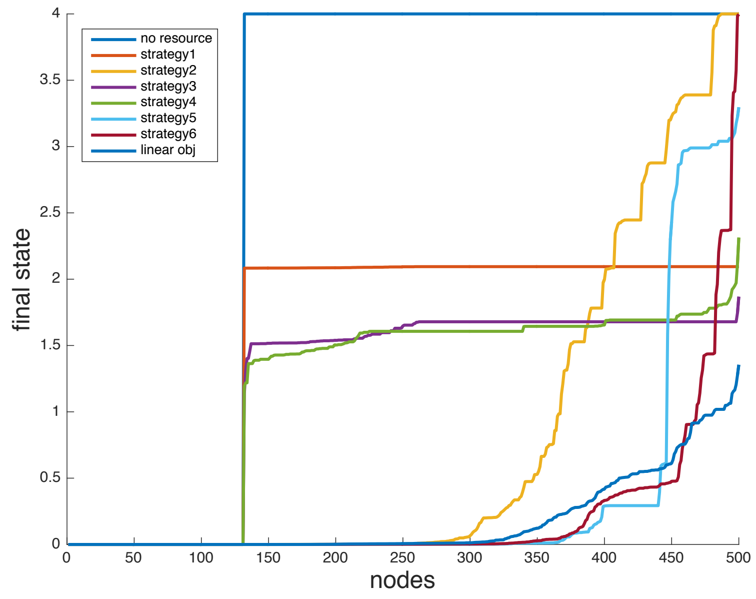
9 Summary and Outlook

To summary, we mainly have done these things in our work:

- Reproduce the results in [1]
- Model disaster spreading as an PDE-constrained optimization problem and come up with two objective functions corresponding to different optimal criteria
- Find the optimal strategy to distribute external resources given the network topology, by solving this optimization problem using adjoint method



(a) Damaged nodes on scale-free network



(b) Final states on scale-free network

Figure 7: (a) Number of damaged nodes and (b) final states on grid network. Parameter setting is the same with that in grid network except for total external resources, i.e. $t_D = 8$ and maximal iterations of gradient descent is 100. Because **S6** has achieve best result with 1000 external resources, in order to discriminate these strategies we decrease external resources to 600.

- Compare the optimal strategies with six heuristic strategies in [1] on grid and scale-free networks.

We would suggest that further work could be done to study more representative setting of objective function, e.g. a weighted sum of the two functions proposed in our work. Also one could study the influence of different network topologies and initial breaking nodes location on the optimized strategy. This might give us more insights into how to prevent and stop the spreading of disasters. Moreover, the amount of available resources is also an important parameter which will influence our solution.

References

[1] Lubos Buzna, Karsten Peters, Hendrik Ammoser, Christian Kühnert, and Dirk Helbing. Efficient response to cascading disaster spreading. *Physical Review E*, 75(5):056107, 2007.

[2] L Chen, W Herreman, and A Jackson. Optimal dynamo action by steady flows confined to a cube. *Journal of Fluid Mechanics*, 783:23–45, 2015.

[3] Kuan Li, Andrew Jackson, and Philip W Livermore. Variational data assimilation for the initial-value dynamo problem. *Physical Review E*, 84(5):056321, 2011.

[4] Jeroen Tromp, Dimitri Komattisch, and Qinya Liu. Spectral-element and adjoint methods in seismology. *Communications in Computational Physics*, 3(1):1–32, 2008.

10 Appendix I – Details about the adjoint method

In this section we will present the adjoint system of our disaster spreading model. For numerical implementation reason, we always need to transform the equation into a discrete version. One way to do this is to find a suitable basis to expand $r(t)$, but the choice here is not so clear in this case and we decide to discretize the resource strategy. $R_i(t)$ will be piecewise constant function, i.e. resource will only arrive at certain time, rather than a continuous function. Let us rewrite the controlling equation as

$$\frac{\partial \mathbf{x}}{\partial t} = -\mathscr{P}(t)\mathbf{x} + \mathscr{K}\mathbf{x}, \tag{15}$$

in which $\mathscr{P}(t)$ is a diagonal matrix

$$\mathscr{P}_{ij}(k) = \frac{1}{\tau_i(k)}\delta_{ij} = \frac{1}{(\tau_{start} - \beta_2)e^{-\alpha_2 \sum_{nt=1}^{k} \Delta R_i(nt)} + \beta_2}\delta_{ij} \tag{16}$$

and $\mathscr{K}$ is also a matrix with each element of $\mathscr{K}$ as a kernel function,

$$\mathscr{K}_{ij}(t,s) = \frac{M_{ji}e^{-\beta t_{ji}}}{f(O_j)}\delta(s-t+t_{ji}), \tag{17}$$

correspond to the two terms on the r.h.s of Eq. (2). Clearly $\mathscr{P}(t)$ is a self-adjoint operator and the adjoint of $\mathscr{K}$ can be easily obtained by using Eq. (6),

$$\mathscr{K}^{\dagger}{}_{ij}(t,s) = \frac{M_{ji}e^{-\beta t_{ji}}}{f(O_j)}\delta(t+t_{ji}-s). \tag{18}$$

Now we can formulate our Lagrange as

$$\begin{aligned}\mathscr{L} =& \sum_i f(x_T^{(i)}) + \int_0^T \langle \tilde{\mathbf{x}^{\dagger}}, \frac{\partial \tilde{\mathbf{x}}}{\partial t} - (\mathscr{K}-\mathscr{P})\tilde{\mathbf{x}}\rangle dt \\ &+ \int_{10}^T \langle \tilde{\mathbf{x}_l^{\dagger}}, \frac{\partial \tilde{\mathbf{x}_l^{\dagger}}}{\partial t} - (\mathscr{K}-\mathscr{P})\tilde{\mathbf{x}_l^{\dagger}}\rangle dt + \sum_{t=1}^{Nt}\lambda_t(\sum_i \Delta R_i(t) - \Delta R(t)).\end{aligned} \tag{19}$$

l is index for the starting node of disaster spreading. $\tilde{\mathbf{x}^{\dagger}}$ stands for $\mathbf{x}^{\dagger}$ with its l th element be 0, while $\tilde{\mathbf{x}^{\dagger}}_l$ represents a vector that all elements of $\mathbf{x}^{\dagger}$ is set to zero except for the l th one. $\Delta R_i(t)$ stands for resource increment at time t and $\Delta R(t)$ is the sum of increments at all nodes in the network. We assume our objective function has the form $\sum_i f(x_T^{(i)})$.

Following the same steps as in Section 6, one can obtain the adjoint equations:

$$-\frac{\partial x_i^{\dagger}}{\partial t} = \sum_j (\mathscr{K}^{\dagger}{}_{ij} - \mathscr{P}(t)_{ij})x_j^{\dagger}, \forall i \neq l, t \in [0,T] \tag{20}$$

and

$$-\frac{\partial x_l^{\dagger}}{\partial t} = \sum_j (\mathscr{K}^{\dagger}{}_{lj} - \mathscr{P}(t)_{lj})x_j^{\dagger}, t \in [10,T]. \tag{21}$$

The compatibility condition is given by

$$(x_T^{\dagger})^{(i)} = -f'(x_T^{(i)}), \tag{22}$$

and the gradient of $\mathscr{L}$ w.r.t. $\Delta R_i(t)$ is

$$\begin{aligned}\frac{\partial \mathscr{L}}{\partial \Delta R_i(t)} &= \lambda_t + \int_0^T \langle \frac{\partial \mathscr{P}}{\partial \Delta R_i(t)}\tilde{\mathbf{x}}, \tilde{\mathbf{x}^{\dagger}}\rangle dt + \int_{10}^T \langle \frac{\partial \mathscr{P}}{\partial \Delta R_i(t)}\tilde{\mathbf{x}}_l, \tilde{\mathbf{x}^{\dagger}}_l\rangle dt \\ &= \lambda_t + \sum_{j=t}^{T} w_j \Delta t[\frac{\partial \mathscr{P}_{ii}(j)}{\partial \Delta R_i(t)}]x_i(j)x_i^{\dagger}(j) + \sum_{j=\max(t,10)}^{T} w_j \Delta t[\frac{\partial \mathscr{P}_{ll}(j)}{\partial \Delta R_l(t)}]x_l(j)x_l^{\dagger}(j).\end{aligned} \tag{23}$$

w_j is integration weights and here we assume $t_D = 0$ for simplicity. To evaluate this expression, we also need the value of λ_t, which is calculated from the constraint

$$\sum_i \Delta R_i(t) - \Delta R(t) = 0.$$

The strategy is updated as

$$\Delta R_i(t) \leftarrow \Delta R_i(t) - \Delta \cdot \frac{\partial \mathscr{L}}{\partial \Delta R_i(t)}. \tag{24}$$

As one may notice, the updated $\Delta R_i(t)$ might be negative. To solve this, we project $\Delta R_i(t)$ onto its domain and then normalize it to satisfy the constraint:

$$\Delta R_i(t) \longmapsto \frac{\mathbb{I}_{x>0}(\Delta R_i(t))}{\|\Delta R_i(t)\|} \Delta R(t). \tag{25}$$

11 Appendix II – Explanation about our code structure

11.1 Structure of Our Codes

The source codes used in experiments are in *src* folder, which simulate disaster spreading and adjoint-method algorithm. *src_otherVersion* also contains some source code, which also implements the disaster spreading model. These two version of codes are developed individually. The purpose to implement the same model twice is that we can make sure our results are correct. Besides, the code to generate different networks are stored in *codes_generate_networks* folder.

Results of experiments are stored in *results* folder. More specifically,

- *results/Grid_result_original_paper* corresponds to FIG. 2 (a) and *results/SF_result_original_paper* corresponds to FIG. 2 (b)
- *results/grid_opt_results* corresponds to FIG. 4 and *results/sf_opt_results* corresponds to FIG. 7
- *figs* folder contains the figures used in this report
- *report* folder contains latex template of this report

11.2 How to Use Our Codes

We briefly explain how to run our codes.

1. Change the parameter settings in *initialise_param_and_X.m* file in *src* folder
2. Run the model using *main.m* file in *src* folder with MATLAB
   (a) if you want to simulate one of these six heuristic strategies, change `adj` (line 4) to 0 and `max_iter` (line 5) to 1

   (b) if you want to find the optimal solution, change: (1) `adj` (line 4) to 1, (2) textttmax_iter (line 5) to the maximal iterations of gradient descent, (3) `model.delta` (line 14) to the step size in each iteration of gradient descent, (4) `eps` (line 10) to your stop criterion, (5) `obj` (line 42/43) the objective function and (6) `X_adj` (line 54/55) the initial condition for adjoint simulation.