Optimal resources distribution to disaster spreading

Guangyu Li, Jiawen Luo, Zhichao Han

Optimal resources distribution to disaster spreading

Our presentation is organized as follows:

- Background: disaster spreading model and research question
- Methods: how to find optimal strategy to distribute resources
  - PDE-constrained optimization & Adjoint method
- Results: comparison of different strategies

Background

Disaster spreading model

Based on original paper[1] , the disaster spreading could be modeled as:

- the system containing interacted unities $\Rightarrow G = (V, E)$
- unit i has influence to unit $j \Rightarrow$ a link from node i to node j
- state of node i at time t: $x_i(t) \in \mathbb{R}^+$
- $0 \leq x_i < \theta_i$: node i is not failed; $\theta_i \leq x_i$: node i is failed

The dynamic of node i is modeled by formula (1):

$$\frac{dx_i}{dt} = -\frac{x_i}{\tau_i} + \Theta_i\left(\sum_{i \neq j} \frac{M_{ji} x_j(t - t_{ji})}{f(O_j)} e^{-\beta t_{ji}}\right) \quad (1)$$

[1] L Buzna, K Peters, H Ammoser, C Kühnert, D Helbing, PRE 75 (5), 056107

Research question

In original paper[1] , the author compared six heuristic strategies to distribute resources.
However there exist lots of other heuristic strategies since the ways to distribute resources are infinite.

[1] L Buzna, K Peters, H Ammoser, C Kühnert, D Helbing, PRE 75 (5), 056107

Research question

In original paper[1] , the author compared six heuristic strategies to distribute resources.
However there exist lots of other heuristic strategies since the ways to distribute resources are infinite.

what is the optimal strategy?

[1] L Buzna, K Peters, H Ammoser, C Kühnert, D Helbing, PRE 75 (5), 056107

Methods

Resource distribution as optimization problem

We find this problem could be modeled as an optimization problem:

- Objective function:
  - $J = \sum_i \Phi_i(x_i)$ minimizes the number of damaged nodes at end time $t = T$
  - $J = \sum_i x_i$ minimizes the averaging status of all nodes
- Constraints:
  - $\frac{\partial \mathbf{x}}{\partial t} - (K - P)\mathbf{x} = 0$
  - $\sum_i^N R_i(t) = R(t)$

Forward Equation

$$\frac{\partial \mathbf{x}}{\partial t} = (K - P)\mathbf{x}$$

in which $P(t)$ is a diagonal matrix

$$P_{ij}(k) = \frac{1}{\tau_i(k)}\delta_{ij} = \frac{1}{(\tau_{start} - \beta_2)e^{-\alpha_2 \sum_{t=1}^{k} \Delta R_i(t)} + \beta_2}\delta_{ij}$$

and K is also a matrix with each element of K as an integration kernel,

$$K_{ij}(t, s) = \frac{M_{ji} e^{-\beta t_{ji}}}{f(O_j)}\delta(s - t + t_{ji})$$

The Lagrangian

Formulate the Lagrangian as:

$$L = \sum_i f(x_T^{(i)}) + \int_0^T \langle \tilde{\mathbf{x}}^\dagger, \frac{\partial \tilde{\mathbf{x}}}{\partial t} - (K - P)\tilde{\mathbf{x}} \rangle dt$$

$$+ \int_{10}^T \langle \tilde{\mathbf{x}}_l^\dagger, \frac{\partial \tilde{\mathbf{x}}_l^\dagger}{\partial t} - (K - P)\tilde{\mathbf{x}}_l^\dagger \rangle dt + \sum_{t=1}^{Nt} \lambda_t (\sum_i \Delta R_i(t) - \Delta R(t))$$

Take variations of the Lagrangian

- Adjoint equation:

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

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

- Compatibility condition: $(x_T^\dagger)^{(i)} = -f'(x_T^{(i)})$
- Gradient:

$$\frac{\partial L}{\partial \Delta R_i(t)} = \lambda_t + \int_0^T \langle \frac{\partial P}{\partial \Delta R_i(t)} \tilde{\mathbf{x}}, \tilde{\mathbf{x}}^\dagger \rangle dt + \int_{10}^T \langle \frac{\partial P}{\partial \Delta R_i(t)} \tilde{\mathbf{x}}_l, \tilde{\mathbf{x}}^\dagger{}_l \rangle dt$$

$$= \lambda_t + \sum_{j=t}^{T} w_j \Delta t [\frac{\partial P_{ii}(j)}{\partial \Delta R_i(t)}] x_i(j) x_i^\dagger(j) + \sum_{j=\max(t,10)}^{T} w_j \Delta t [\frac{\partial P_{ll}(j)}{\partial \Delta R_l(t)}] x_l(j) x_l^\dagger(j).$$

Algorithm

- Start from random strategy or an existed one
- Do once forward simulation $\rightarrow \mathbf{x}$
- Calculate $\mathbf{x}_T^\dagger$ from compatibility condition
- Do once backward simulation $\rightarrow \mathbf{x}^\dagger$
- Compute λ_t from constraint: $\sum_i \Delta R_i(t) - \Delta R(t) = 0$
- Calculate $\frac{\partial L}{\partial \Delta R_i(t)}$.
- Update strategy as $\Delta R_i(t) \leftarrow \Delta R_i(t) - \Delta \cdot \frac{\partial L}{\partial \Delta R_i(t)}$
- Normalization: $\Delta R_i(t) \longmapsto \frac{\mathbb{I}_{x>0}(\Delta R_i(t))}{\|\Delta R_i(t)\|} \Delta R(t)$
- Return to second step if not yet converged

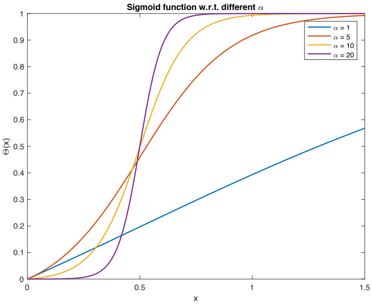
Results

Minimize the number of damaged nodes

0-1 loss funtion:

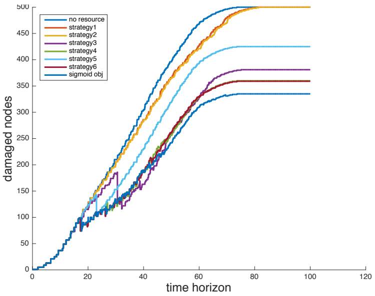
$$J = \sum_i h(x_i) \qquad \text{where } h(\cdot) = \begin{cases} 1 & \text{if } x_i \geq \theta_i \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

But, it is **not differentiable!**

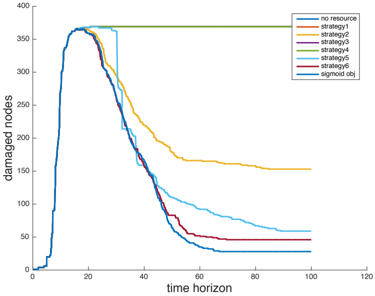


So sigmoid function is used to approximate 0-1 loss.

Minimize number of damaged nodes (cont'd)



Damaged nodes on grid network

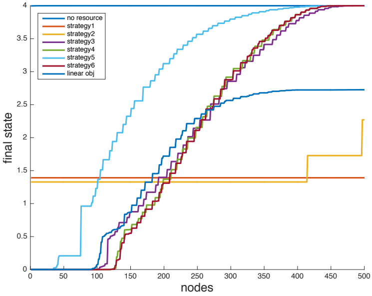


Damaged nodes on scale-free network

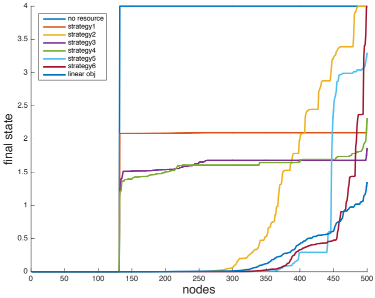
Optimize final states

The objective function is simply:

$$J = \frac{1}{n} \sum_{i} x_T^{(i)} \tag{3}$$



Final state of grid network

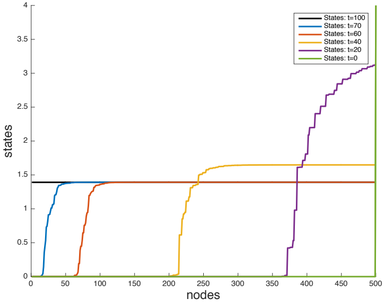


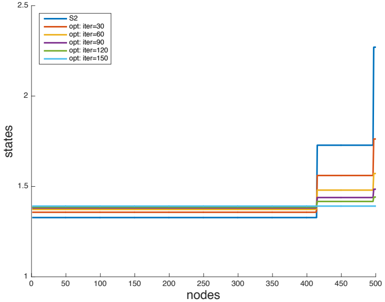
Final state of scale-free network

Optimize Final States (cont'd)

Explanation: the horizontal line of **S1** in grid network:

- it is a stable state, which can be shown mathematically (details in our report)
- it can further show that **S1** is optimal w.r.t optimizing final states (details in our report)





Conclusion

We did these:

- reproduce the results of original paper[1]
- find the optimal strategy by solving the PDE-constrained optimization problem (however, may be a local optimum)
- compare the optimal strategy with these heuristic strategies on grid network and scale-free network

[1] L Buzna, K Peters, H Ammoser, C Kühnert, D Helbing, PRE 75 (5), 056107

Thanks for your attention!