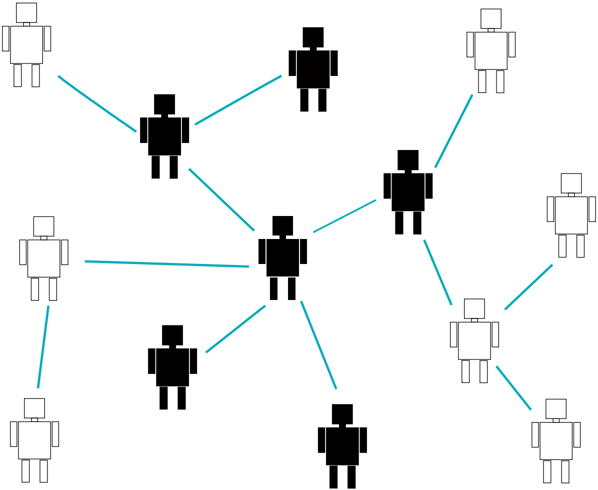
Network Centrality as Statistical Inference in Large Networks

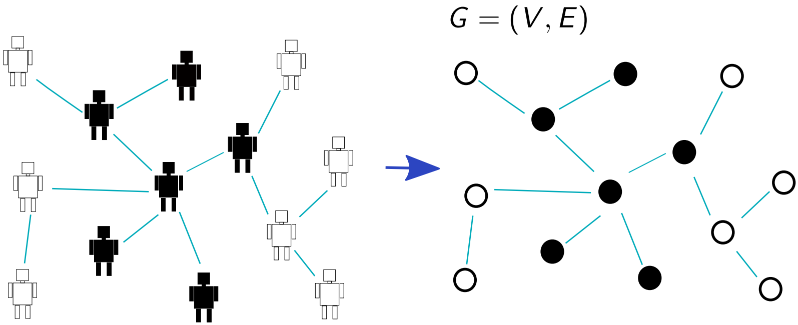
Chee Wei Tan

Institute for Pure and Applied Mathematics
Joint work with Peter Pei-Duo Yu and Hung-Lin Fu

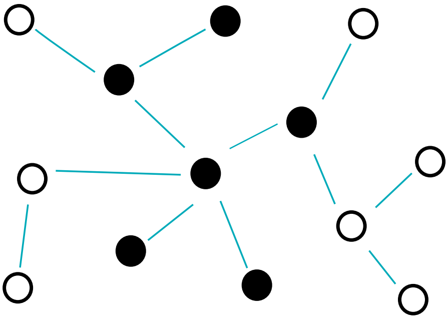
December 4, 2018

Problem Statement



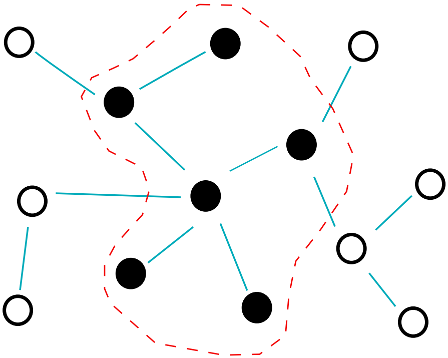


$G = (V, E)$



$G = (V, E)$

Infected subgraph $G_n = G_6$

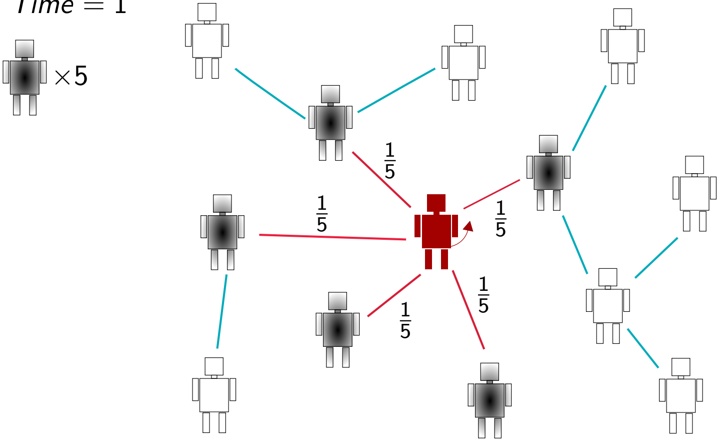


Where is the source in G_6?

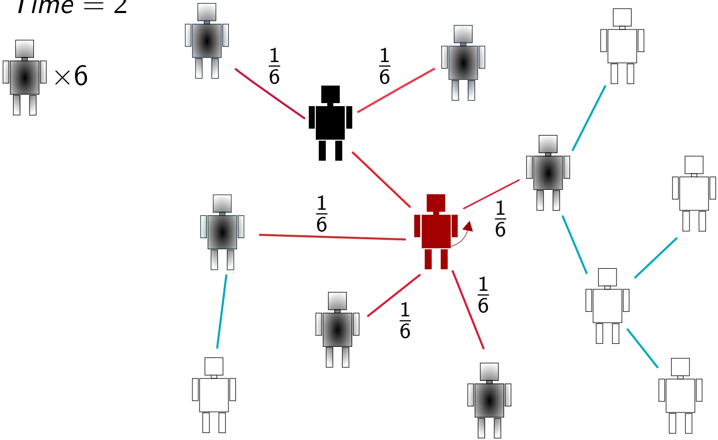
The Model and Assumptions

- The online social network is modeled by an undirected (possibly infinte) graph $G = (V, E)$ where $V = \{v_1, v_2, ...\}$ is the vertex set and $E = \{(i, j) | i, j \in V\}$ is the edge set.
- The users in the online social network are the vertices in G, and the edges model the connection between users.
- Assume the Susceptible-Infectious(SI) spreading model.
- Every vertex is equally like to be the source
- Assume that in each time period, one vertex is uniformly chosen from the neighbors of those infected vertices to be infected.

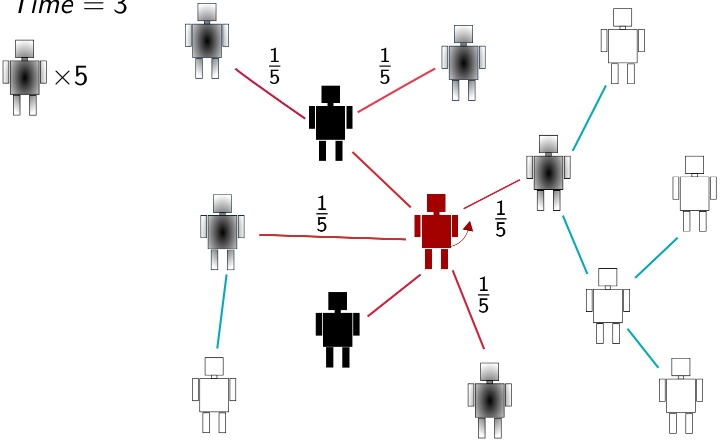
$Time = 1$



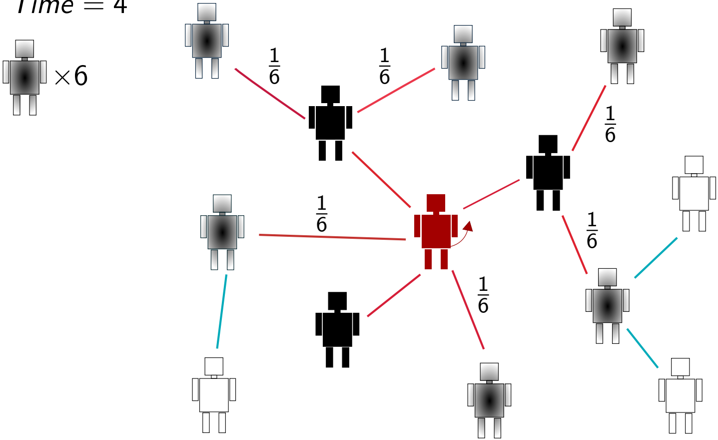
$Time = 2$



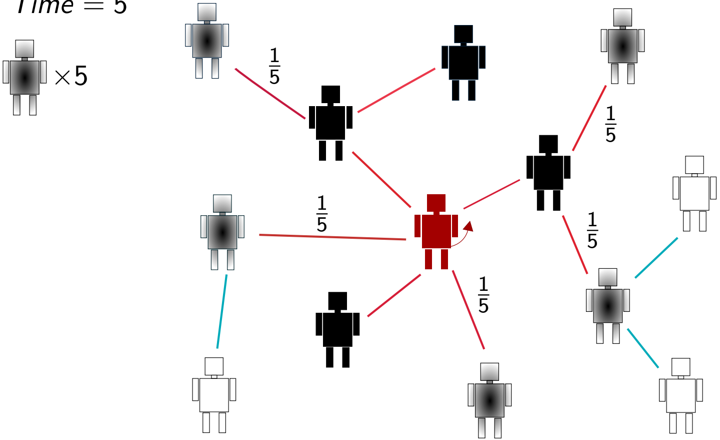
$Time = 3$



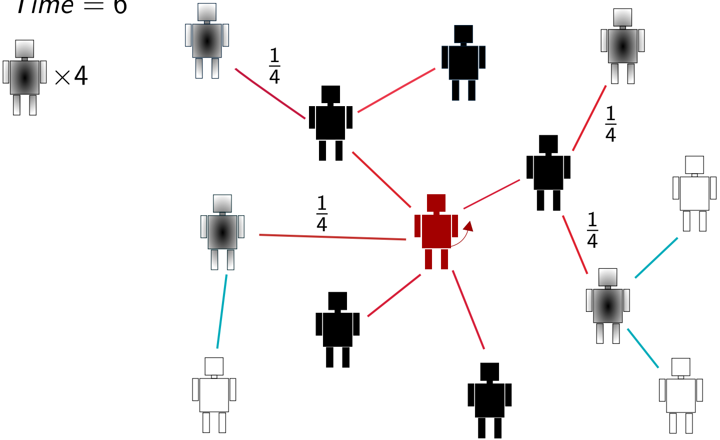
$Time = 4$



$Time = 5$



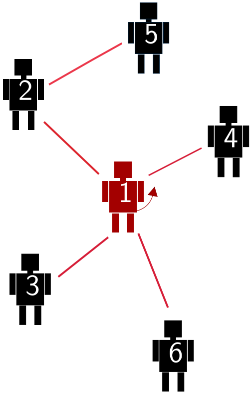
$Time = 6$



$Time = 6$

spreading order:(1,2,3,4,5,6)

$probability = \frac{1}{5} \times \frac{1}{6} \times \frac{1}{5} \times \frac{1}{6} \times \frac{1}{5}$



Maximum Likelihood Estimation Optimization Problem

$$
\begin{aligned}
\text{maximize} \quad & P(v|G_n) \\
\text{subject to} \quad & G \text{ is a degree regular infinite tree} \\
& G_n \subset G
\end{aligned}
$$

Reference: D.Shah and T.Zaman, Rumors in a Network: Who's the Culprit?, IEEE Transaction on Information Theory, 2011.

Preliminary and Problem formulation

Let $\hat{v}$ be the maximum likelihood estimation for the source vertex, then given an observation G_n, we have

$$\hat{v} = \operatorname*{argmax}_{v \in Gn} P(v|G_n) \tag{1}$$

By Bayes' Theorem and the 4_{th} assumption in our model, we have

$$P(v|G_n) \propto P(G_n|v). \tag{2}$$

If G is a regular tree with infinite size, then the probability $P(\sigma_i|v)$ is a constant $\forall \sigma_i \in M(v, G_n)$. We can conclude that, for any $v \in G_n$

$$
\begin{aligned}
P(G_n|v) &= \sum_{\sigma_i \in M(v,G_n)} P(\sigma_i|v) \\
&= |M(v, G_n)| \cdot P(\sigma_i|v) \\
&\propto |M(v, G_n)|
\end{aligned}
$$

In summary, when G is a degree regular tree with infinite size, then

$$P(v_i|G_n) \propto |M(v_i, G_n)|.$$

The value of $|M(v_i, G_n)|$ is called the **rumor centrality** of v_i, and **rumor center** of G_n is the vertex with the maximum rumor centrality.

Thus, the maximum likelihood estimation for the source is the rumor center of G_n. The rumor center can be computed with polynomial time complexity by a message passing algorithm.

Example: 3-Regular Tree without Boundary Effect

Suppose at time $t = 4$, we observe a network G and n infected vertices, which collectively constitutes a spread graph that we denote by G_n. Note that n represents the number of the infected vertices in G. In this example, G_4 is the induced subgraph with vertices $\{1, 2, 3, 4\}$. For the spreading order (1,2,3,4), the probability corresponds is $\frac{1}{3} \times \frac{1}{4} \times \frac{1}{5}$.

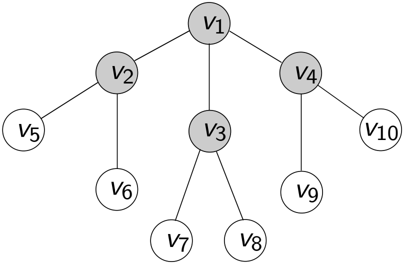


Figure: An example of calculating $P(G_4|v_1)$ when the rumor spread has not reached the end vertices.

There are several ways to spread the rumor from v_1 to other three vertices, for example: (1,2,3,4),(1,3,4,2),(1,2,4,3),(1,3,2,4),(1,4,3,2),(1,4,2,3). For each $\sigma_i \in M(v_1, G_4)$,

$$P(\sigma_i|v_1) = \tfrac{1}{3} \times \tfrac{1}{4} \times \tfrac{1}{5}.$$

Thus,

$$\begin{aligned} P(G_4|v_1) &= \sum_{\sigma_i \in M(v_1, G_n)} P(\sigma_i|v_1) \\ &= |M(v_1, G_n)| \cdot \frac{1}{60} \\ &= \frac{6}{60} \end{aligned}$$

Moreover, we have $P(G_4|v_4) = P(G_4|v_3) = P(G_4|v_2) = \frac{2}{60}$.
v_1 is the maximum likelihood estimation for the source !

Centroid of a Graph

Let us denote the branch weight of a local sub-tree of a vertex v in G_n by

$$\text{weight}(v) = \max_{c \in \text{child}(v)} t_c^v.$$

The vertex of G_n with the *minimum weight* is called the *centroid* of G_n [1].

Theorem

Let G_n be a general tree graph and v is a vertex in G_n. Then, the following statements are equivalent:

1. *The vertex v is a rumor center of G_n and also a distance center of G_n (proved in Shah 2011).*
2. *The vertex v is a centroid of G_n (our CISS 2016 paper).*

Example of G is a finite graph

Let us see what happen if G_n contains an **end vertex** v_5, where v_5 can only receive the rumor.

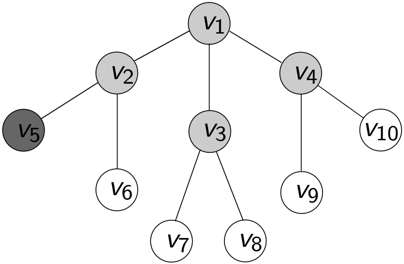


Figure: G is a finite 3-regular tree, G_n is a infected subtree with 1 end node v_5

Example of G is a finite graph (continued)

σ_i	Spreading Order	$P(\sigma_i \| G_5)$	σ_i	Spreading Order	$P(\sigma_i \| G_5)$
σ_1	v_1, v_2, v_5, v_3, v_4	$\frac{1}{144}$	σ_7	v_1, v_2, v_3, v_4, v_5	$\frac{1}{360}$
σ_2	v_1, v_2, v_5, v_4, v_3	$\frac{1}{144}$	σ_8	v_1, v_2, v_4, v_3, v_5	$\frac{1}{360}$
σ_3	v_1, v_3, v_2, v_5, v_4	$\frac{1}{240}$	σ_9	v_1, v_3, v_2, v_4, v_5	$\frac{1}{360}$
σ_4	v_1, v_4, v_2, v_5, v_3	$\frac{1}{240}$	σ_{10}	v_1, v_3, v_4, v_2, v_5	$\frac{1}{360}$
σ_5	v_1, v_2, v_3, v_5, v_4	$\frac{1}{240}$	σ_{11}	v_1, v_4, v_2, v_3, v_5	$\frac{1}{360}$
σ_6	v_1, v_2, v_4, v_5, v_3	$\frac{1}{240}$	σ_{12}	v_1, v_4, v_3, v_2, v_5	$\frac{1}{360}$

This example reveals some interesting properties of boundary effects due to even a single end vertex:

- $P(\sigma_i|v_1)$ increases with how soon the end vertex $_5$ appears in σ_i (as ordered from left to right of σ_i).
- When there is at least one end vertex in G_n, then $P(G_n|v)$ is no longer proportional to $|M(v_1, G_n)|$.

Maximum Likelihood Estimation Optimization Problem

$$\begin{aligned} \text{maximize} \quad & P(v|G_n) \\ \text{subject to} \quad & G \text{ is a degree regular finite tree} \\ & G_n \text{ contains at least one end vertex} \end{aligned}$$

Reference: P. Yu, C. W. Tan and H. Fu, Rumor Source Detection in Finite Graphs with Boundary Effects by Message Passing Algorithms, IEEE/ACM International Conference on Social Networks Analysis and Mining, 2017.

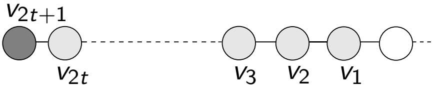


Figure: G_n as a line graph with a single end vertex $v_e = v_{2t+1}$.

The following is the analytical formula for $P(G_n|v_i)$ when $i \neq 2t+1$:

$$P(G_n|v_i) = \begin{cases} \displaystyle\prod_{l=1}^{n-1} \frac{1}{z_d(l)+1}, & i = n; \\ \displaystyle\sum_{k=n-i+1}^{n} \binom{k-2}{k-n+i-1} \cdot P_{v_i}^{v_e}(G_n, k), & \text{otherwise}, \end{cases} \quad (3)$$

where $P_{v_i}^{v_e}(G_n, k)$ is given in the previous slide.

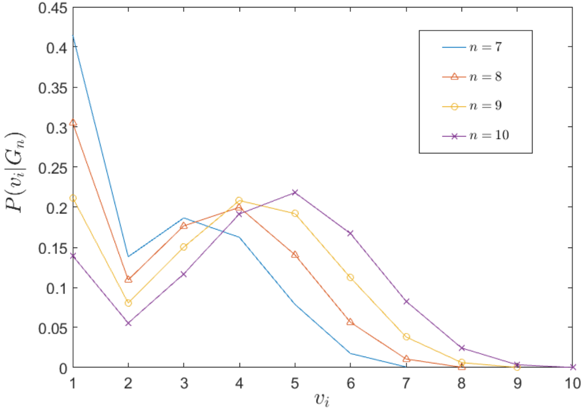


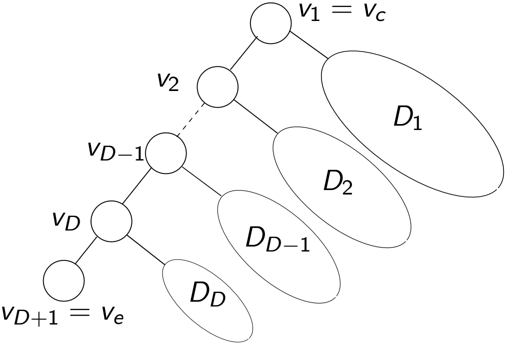
Figure: $P(v|G_n)$, where G_n is a line graph with a single end vertex v_1 over an underlying 4-regular finite graph. Note that v_1 in this figure corresponds to v_{2t+1} in Figure 3.

Theoretical Results for Finite Graph G

Without the property $P(G_n|v) \propto |M(v, G_n)|$, it is hard to find the ML-estimator. But the following theorem narrows down the range of the search when G_n has a single end vertex v_e.

Theorem

Let G be a tree with finite order and $G_n \subseteq G$ is a subtree of G with a single end vertex $v_e \in G_n$, then the maximum likelihood estimator $\hat{v}$ that maximizes $P(v|G_n)$ is located on the path from the centroid v_c to v_e.


$v_1 = v_c$
v_2
v_{D-1}
v_D
$v_{D+1} = v_e$
D_1
D_2
D_{D-1}
D_D

1. **input:** G_n, $\kappa = \{\}$
2. Compute the centroid v_c of G_n.
3. Choose v_c as the root of a tree and use a message-passing algorithm to count the number of end vertices on each branch of this rooted tree.
4. Starting from v_c, and at each hop choose the child with the maximum number of end vertices (if there were more children with the same maximal number of end vertices, then choose all of them). This tree traversal yields a subtree t_{ML} rooted at v_c.
5. **Output:** $\kappa = \{$parent vertices of leaves of $t_{ML}, v_c\}$

Algorithm for ML-estimator on G_n with multiple End Vertices

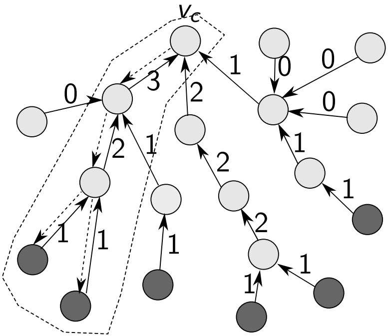
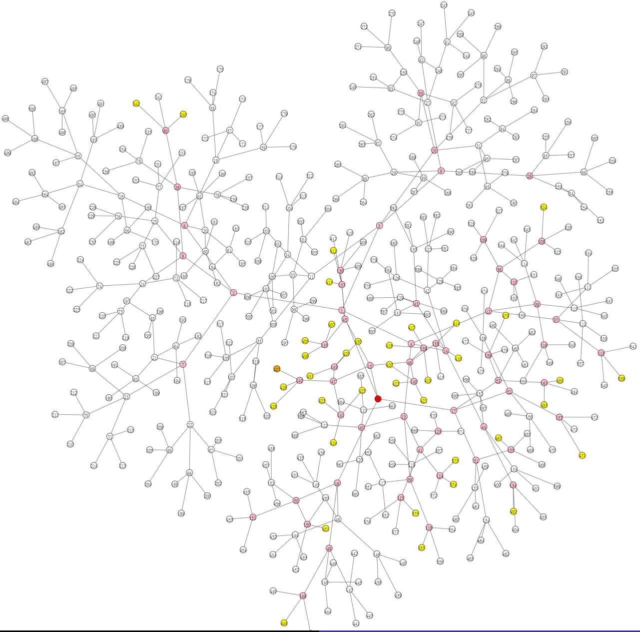
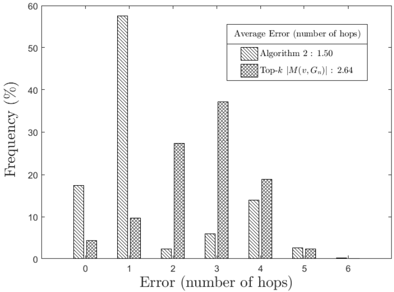


Figure: This figure illustrate how the algorithm works on a tree.

Result N=500, n=100, end vertices:37



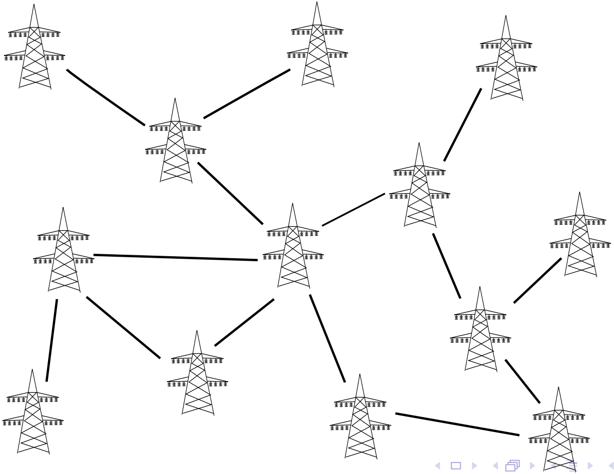
Numerical Result N=500, n=100,d=3,4,5,6



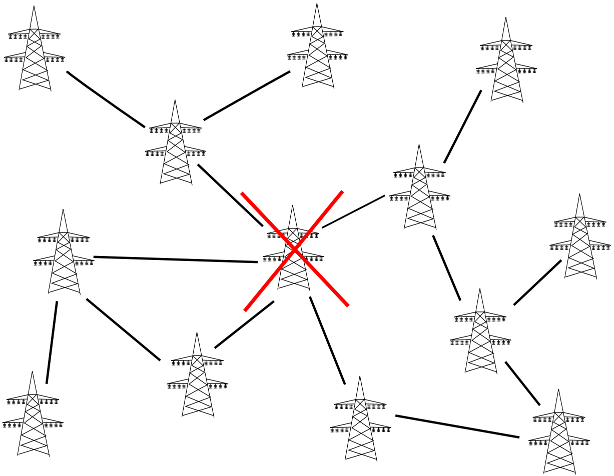
d	$\lvert EndVertices \rvert$	New	Old
3	29.82	1.6	3.2
4	45.23	1.52	2.67
5	55.4	1.81	2.55
6	62.9	1.59	2.3

Problem Statement

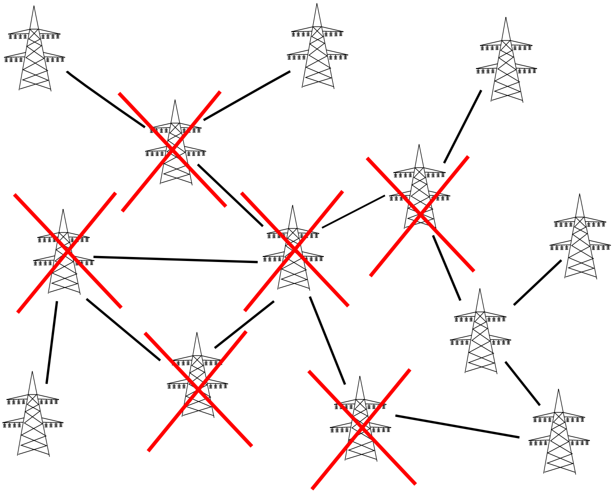
P. Yu, C. W. Tan and H. Fu, Averting Cascading Failures in Networked Infrastructures: Poset-constrained Graph Algorithms, IEEE Journal of Selected Topics in Signal Processing, 2018 [2]



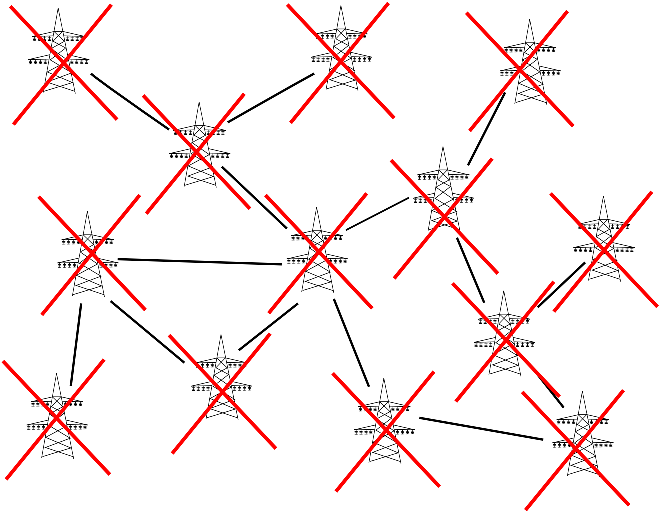
Problem Statement



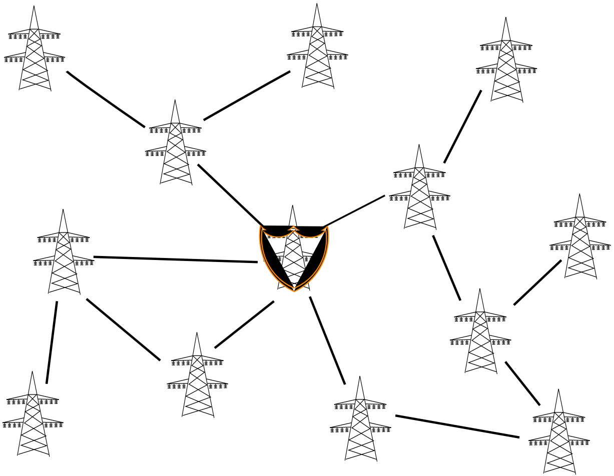
Problem Statement



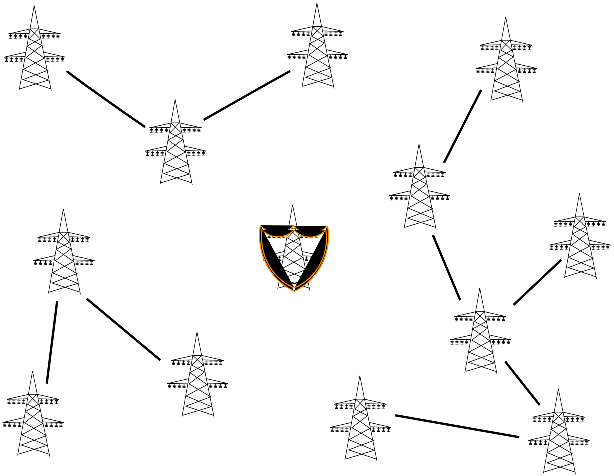
Problem Statement



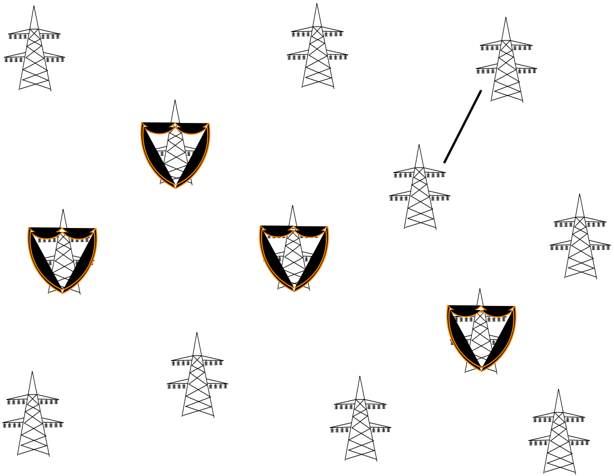
Problem Statement



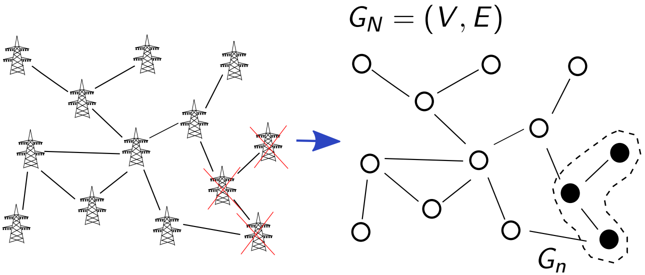
Problem Statement



Problem Statement



The Model and Assumptions



The Model and Assumptions

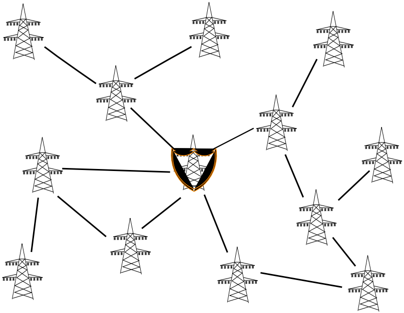
In the extended SI model, we have three types of nodes described as following:

- Susceptible node: Nodes that are susceptible to failure.
- Infected node: Nodes that are under the effect of failure.
- Protected node: Nodes that are protected and can not spread the failure further.
- Every vertex is equally likely to be the source.
- Assume that in each time period, one vertex is uniformly chosen from the neighbors of those infected vertices to be infected.

The Protection Node Placement Problem

$$\begin{aligned} \underset{v \in V_P \subseteq G_n}{\text{minimize}} \quad & \mathbf{E}(|G_n|) \\ \text{subject to} \quad & |V_P| = k, \end{aligned} \tag{4}$$

where $\mathbf{E}(|G_n|)$ is the expectation of the number of failed nodes (i.e., the spread of the cascading failure should it happen)



$$\mathbf{E}(|G_n|) = \frac{1}{13} \cdot [(3+3+3)+(3+3+3)+(6+6+6+6+6+6)]$$
$$= \frac{1}{13} \cdot [3^2 + 3^2 + 6^2]$$

The Protection Node Placement Problem

$$\begin{aligned} \underset{V_P \subseteq V(G_n)}{\text{minimize}} \quad & (C_1^{\{V_P\}})^2 + (C_2^{\{V_P\}})^2 + \ldots + (C_m^{\{V_P\}})^2 \\ \text{subject to} \quad & |V_P| = k, \end{aligned} \tag{5}$$

where $C_1^{\{V_P\}}, C_2^{\{V_P\}}, \ldots,$ and $C_m^{\{V_P\}}$ are the connected components after removing vertices in V_P from G_N.

Posets and Linear Extensions

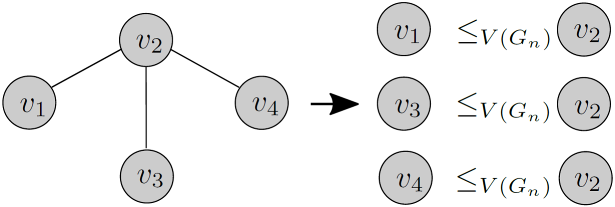
Definition

A non-strict partial order is a relation $\leq_S$ over a set S satisfying the following rules, for all $v_1, v_2, v_3 \in S$:

- $v_1 \leq_S v_1$ (reflexivity)
- if $v_1 \leq_S v_2$ and $v_2 \leq_S v_1$, then $v_1 = v_2$ (antisymmetry)
- if $v_1 \leq_S v_2$ and $v_2 \leq_S v_3$, then $v_1 \leq_S v_3$ (transitivity)

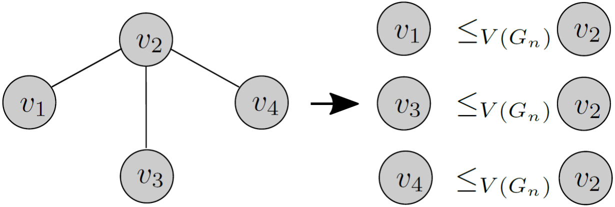
A **total order** has one more rule that every two elements in the set must be assigned a relation.
A **linear extension** $\leq_S^*$ of a partial order $\leq_S$ is a total order which preserve the relation in $\leq_S$, i.e., for all $v_1 \leq_S^* v_2$ whenever $v_1 \leq_S v_2$.

Posets and Rooted Trees



There is no relation between v_1, v_3 and v_4, hence this order is a **partial** order.

Linear Extensions and Cascading Failure



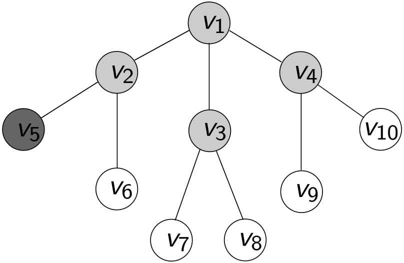
Consider a cascading failure on this graph with a specific order, for example $v_2 \to v_1 \to v_3 \to v_4$, then there is relation between any two vertices in this set, i.e., this specific order is a linear extensions on this posets (rooted tree). Intuitively, choosing the vertex with the maximum number of linear extensions to be protected is a good choice! [3]

Network Centrality to Determine Maximum Number of Linear Extensions of a Poset

Definition

Let G_n be a tree with n vertices, for any $u, v \in G_n$, let t_v^u be the subtree rooted at v by removing the edge (u, v) from G_n and slightly abusing the notation of the subtree size t_v^u as t_v^u.

For example, $t_{v_1}^{v_2} = 7$ and $t_{v_2}^{v_1} = 3$.



Definition

Define the branch weight of a vertex v in G_n by

$$\text{weight}(v) = \max_{c \in \text{child}(v)} t_c^v.$$

The vertex of G_n with the *minimum weight* is called the *centroid* of G_n [1]. For example, v_1 has the minimum weight, hence v_1 is the centroid.

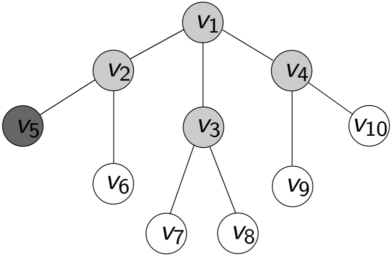


Figure: G is a finite 3-regular tree, G_n is a infected subtree with 1 end node v_5.

Theorem

Let G_N be a general tree graph. Then, the rooted tree with the maximum number of linear extensions is rooted at v^* if and only if v^* is a centroid of G_N(proved in [4]).

Message Passing Algorithm to compute the Centroid of a Graph

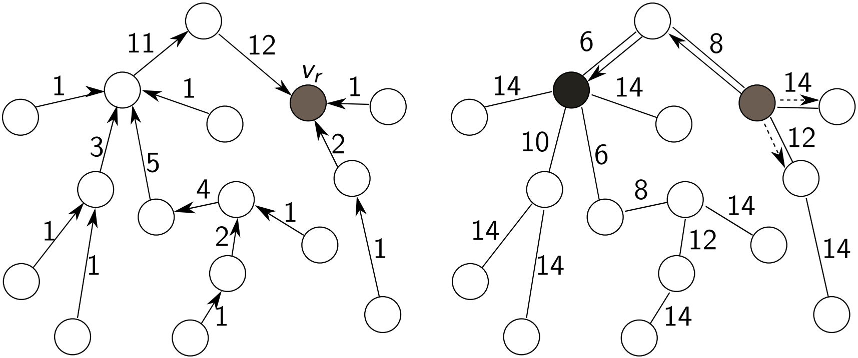
Let $M^{i \to j}$ denote the message from vertex i to vertex j. Let $\text{Diff}(i,j)$ be defined by $\text{Diff}(i,j) = |M^{i\to j} - M^{j \to i}|$.

Theorem

Given a tree G_n with n vertices.
$v_c \in G_n$ is the centroid if and only if $\forall v$ adjacent to v_c and $v_i, v_j \in V(G_n)$, $min_{(v,v_c)\in E(G_n)}\{\text{Diff}(v_c, v)\} \leq \{\text{Diff}(v_i, v_j)\}$. Moreover, for any $u \in G_n$, on the path from v_c to u say $(v_1, v_2, ..., v_D)$, where $v_1 = v_c$ and $v_D = u$. The sequence of $\text{Diff}(v_i, v_{i+1})$ for $i = 1, 2...D$ is increasing.

Message Passing Algorithm to compute the Centroid of a Graph

$n = 12 + 1 + 2 + 1 = 16$



Assume G_N is a tree:

- When $|V_p| = 1$, we choose the centroid to be the solution.
- When $|V_p| > 1$, we use the centroid decomposition to select the protection set.

This may not be the optimal solution, but the performance can be bounded above.

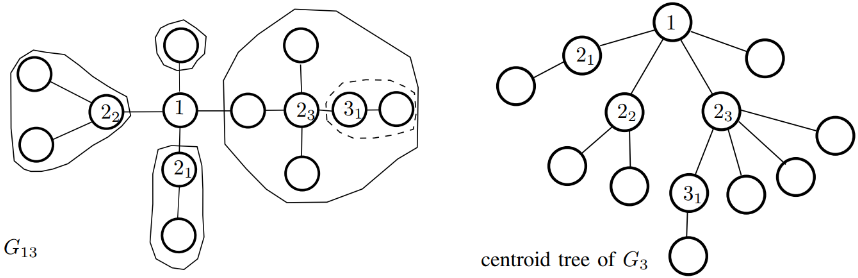
Theorem

Let $f(\{V_p\})$ denote the objective function in (5) and let V_p^* denote the optimal solution of (5). The centroid decomposition approach guarantees that

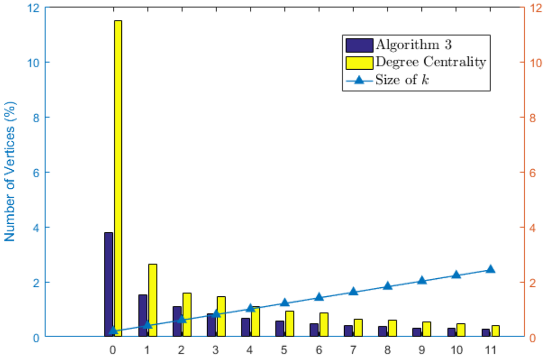
$$1 \leq \frac{f(\{V_p\})}{f(\{V_p^*\})} \leq c\frac{N}{k+1},$$

where k is the size of the protection set V_p and c is a small constant.

Centroid Decomposition

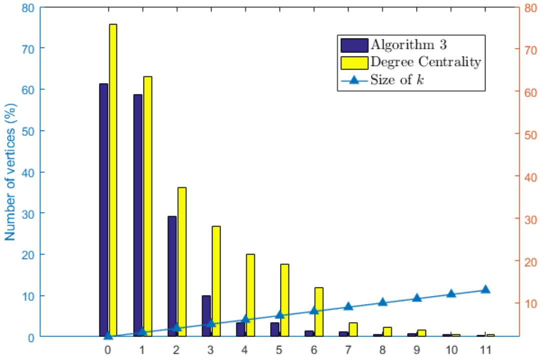


Experimental Results: $N = 4941$



A simulation result when G_N is a random tree. The y-axis represents the number of vertices in percentage and the x-axis represents each trial with different size of k.

Experimental Results: $N = 4941$



A simulation result when G_N is a real world network: Western United State Power Grid Network. The y-axis represents the number of vertices and the x-axis represents each trial with different size of k.

Conceptual Framework for Optimality and Algorithm Design:

- An appropriate network centrality induces a metric on each graph node, and brings graph algorithm machinery to bear on solving the stochastic program
- Explore a variety of useful compact measures of the importance of nodes in the network imbued with optimality basis
- Reverse-engineer or forward-engineer the optimal solution of an optimization problem

In the reverse engineering perspective, we ask:

- Given a network centrality, what are the statistical inference optimization problems that it implicitly solves?
- Distance centrality and branch weight centrality solve the rumor source detection problem for degree-regular tree graphs.
- Betweenness centrality solves the protection node placement problem for a single node special case.
- Network centrality provides guiding principle on algorithm design and can compute exact or approximate solutions.

Network Centrality as Statistical Inference

In the forward engineering perspective, we ask:

- Given a stochastic optimization formulation over a network, how to transform it or to decompose it to one whose subproblems are graph-theoretic and can utilize network centrality, then solve or approximate the whole problem?
- Rumor source detection as a maximum-likelihood estimation problem solved by rumor centrality.
- Expected cascade size minimization problem solved by vaccine centrality.
- New algorithms can be designed based on message-passing (belief propagation) graph analysis
- Deep connections between network centrality on induced abstract data types with probability on trees and graphs.

Thank You!

http://www.cs.cityu.edu.hk/∼cheewtan
Email: cheewtan@cityu.edu.hk

B. Zelinka, "Medians and peripherians of trees," *Arch. Math.*, vol. 4, no. 2, pp. 87–95, 1968.

P. D. Yu, C. W. Tan, and H. L. Fu, "Averting cascading failures in networked infrastructures: Poset-constrained graph algorithms," , *IEEE Journal of Selected Topics in Signal Processing*, p. forthcoming, 2018.

D. Shah and T. Zaman, "Rumors in a network: Whos's the culprit?" *IEEE Trans. Information Theory*, vol. 57, no. 8, pp. 5163–5181, 2011.

C. W. Tan, P. D. Yu, C. K. Lai, W. Zhang, and H. L. Fu, "Optimal detection of influential spreaders in online social networks," *Proc. of Conference on Information Systems and Sciences*, pp. 145–150, 2016.