# Complexity vs. Optimality: Unraveling Source-Destination Connection in Uncertain Graphs

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University

May 3, 2017

# Outline

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University

# Outline

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University

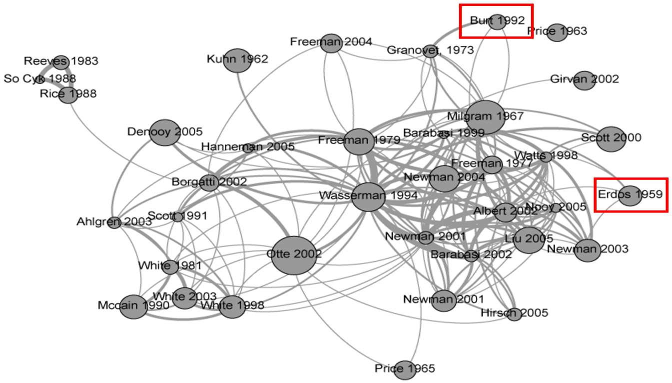
# Motivating Examples



Can the two nodes communicate with each other?

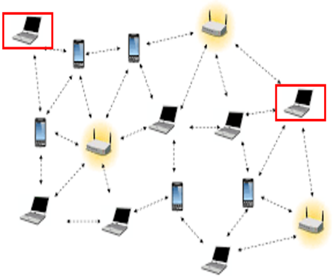
# Motivating Examples



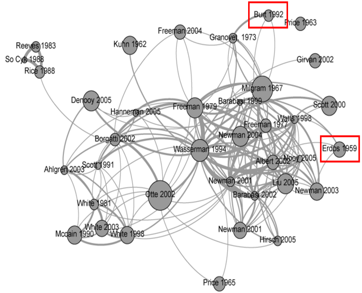



Is the two papers related with each other?

# Motivating Examples



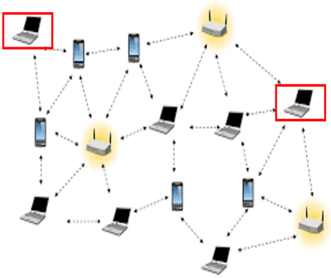
Communication Network



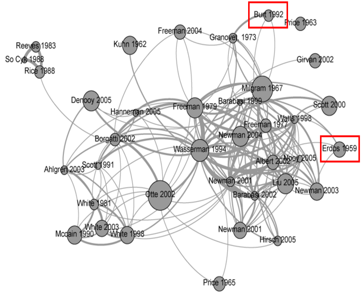

Citation Network

- Link Failure, "Gift Citation"
- **Solution:** Link Probing, Text Mining

# Motivating Examples



Communication Network




Citation Network

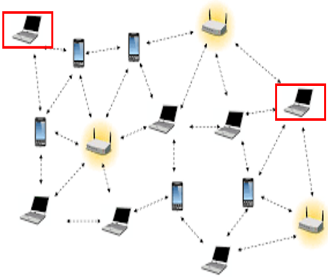
- Link Failure, "Gift Citation"
- **Solution:** Link Probing, Text Mining

# Motivating Examples

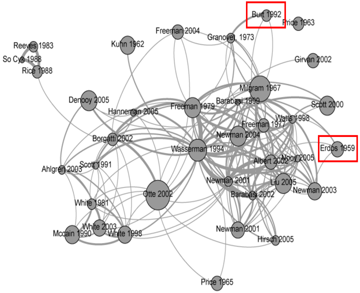
Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University





Communication Network




Citation Network

**How to perform tests cost-effectively?**

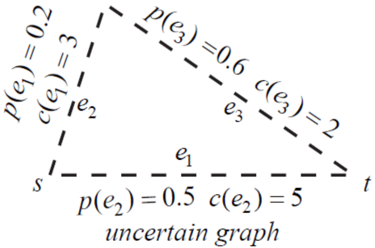
# Outline

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University

# Uncertain Graph

- Network topology (Prior,$(V, E)$)
- Prior existence probability (Edges, $p$)
- Testing cost (Edges, $c$)
- Composition of realizations

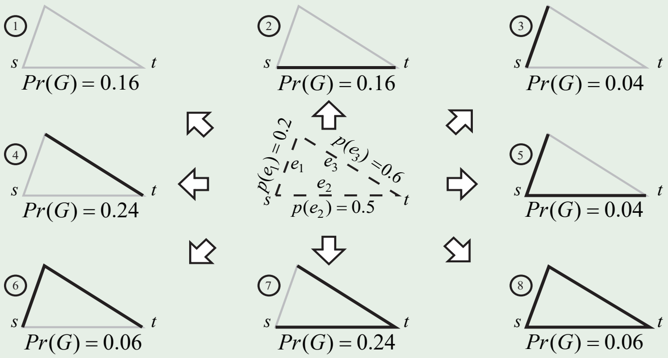


# Uncertain Graph and Underlying Graph



For an uncertain graph $\mathcal{G}$, denote $G$ as its underlying realization. $\mathcal{G}$ can be interpreted as a product distribution over all its possible realizations.

## Example (Uncertain graph and its underlying realizations)



# Problem Definition



**Definition (Connectivity Determination Problem)**

Given an uncertain graph $\mathcal{G}(V, E, p, c)$ and two nodes $s, t \in V$ designated as source and destination, find a testing strategy to determine the $s$-$t$ connectivity while incurring the minimum expected cost.

- The results of tests are dictated by the (priorly unknown) underlying graph.
- The expectation of cost is taken over all possible realizations of $\mathcal{G}$.
- The testing strategy can be adaptive.

# Problem Definition



**Definition (Connectivity Determination Problem)**

Given an uncertain graph $\mathcal{G}(V, E, p, c)$ and two nodes $s, t \in V$ designated as source and destination, find a testing strategy to determine the $s$-$t$ connectivity while incurring the minimum expected cost.

- The results of tests are dictated by the (priorly unknown) underlying graph.
- The expectation of cost is taken over all possible realizations of $\mathcal{G}$.
- The testing strategy can be adaptive.

# Adaptive Testing Strategy

How to properly define the testing strategy?

- A strategy decides the next edge to test based on the previous results.
- A strategy terminates by verifying the existence of an *s*-*t* path or an *s*-*t* cut.

**Definition (Temporary State)**

A temporary state s of an uncertain graph $\mathcal{G}(V, E, p, c)$ is an $|E|$-dimension vector with elements "0", "1" and "*". Define $\mathcal{S} = \{0, 1, *\}^{|E|}$ to be the set of temporary states associated with $\mathcal{G}$.

# Adaptive Testing Strategy



How to properly define the testing strategy?

- A strategy decides the next edge to test based on the previous results.
- A strategy terminates by verifying the existence of an *s*-*t* path or an *s*-*t* cut.

**Definition (Temporary State)**

A temporary state s of an uncertain graph $\mathcal{G}(V, E, p, c)$ is an $|E|$-dimension vector with elements "0", "1" and "*". Define $\mathcal{S} = \{0, 1, *\}^{|E|}$ to be the set of temporary states associated with $\mathcal{G}$.

# Adaptive Testing Strategy



## Definition (Temporary State)

A temporary state **s** of an uncertain graph $\mathcal{G}(V, E, p, c)$ is an $|E|$-dimension vector with elements "0", "1" and "*". Define $\mathcal{S} = \{0, 1, *\}^{|E|}$ to be the set of temporary states associated with $\mathcal{G}$.

## Definition (Adaptive Testing Strategy)

An adaptive testing strategy is a mapping $\pi : \mathcal{S} \mapsto E \cup \{\perp\}$.

- Define $E_\pi(G)$ as the set of edges strategy $\pi$ on the underlying graph $G$.
- The expected cost of $\pi$ is given as $Cost(\pi) = \sum_{G\in\mathcal{G}}[Pr(G)\sum_{e\in E_\pi}(G)c(e)]$.

# Outline

# Two Variants of the Problem



1. Compute the whole strategy (P1)

**Definition (Decision Version of P1)**

Given an uncertain graph $\mathcal{G}(V, E, p, c)$ and two nodes $s, t \in V$ designated as source and destination, is there a testing strategy that determines the $s$-$t$ connectivity with expected cost less than $k$.

2. Compute the strategy sequentially (P2)

**Definition (Decision Version of P2)**

Given an uncertain graph $\mathcal{G}(V, E, p, c)$, two nodes $s, t \in V$ designated as source and destination and the current temporary state, decide the optimal next edge to test.

# Two Variants of the Problem



1. Compute the whole strategy (P1)

**Definition (Decision Version of P1)**

Given an uncertain graph $\mathcal{G}(V, E, p, c)$ and two nodes $s, t \in V$ designated as source and destination, is there a testing strategy that determines the $s$-$t$ connectivity with expected cost less than $k$.

2. Compute the strategy sequentially (P2)

**Definition (Decision Version of P2)**

Given an uncertain graph $\mathcal{G}(V, E, p, c)$, two nodes $s, t \in V$ designated as source and destination and the current temporary state, decide the optimal next edge to test.

# Complexity-theoretic Results on P1



**Theorem**

*Computing the expected cost of the optimal strategy is #P-hard [1]. (The decision version of a #P-hard problem is NP-hard)*

**Proof**

- By reduction from the network ($s$-$t$) reliability problem.

**Definition (The Network Reliability Problem)**

Given a network $G(V, E)$ with two nodes in $V$ designated as source and destination, assuming that each edge fail with probability $\frac{1}{2}$, compute the probability of $s$ being connected to $t$.

[1] L. G. Valiant, "The complexity of enumeration and reliability problems"

# Complexity-theoretic Results on P2



**Theorem**

*Deciding the optimal first edge to test is NP-hard.*

**Proof.**

The proof is done by reduction from the set cover problem.

**Definition (The Set Cover Problem)**

Given a universe $\mathcal{U}$ of elements, a family $\mathcal{S}$ of subsets of the universe and a predefined integer $k$, does there exist a subfamily $\mathcal{C} \subseteq \mathcal{S}$ such that $\bigcup_{C \in \mathcal{C}} = \mathcal{U}$ and $|\mathcal{C}| \leq k$.

□

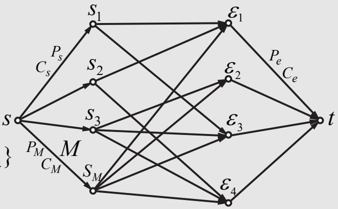
# Complexity-theoretic Results on P2

**Theorem**

*Deciding the optimal first edge to test is NP-hard.*

**Proof.**

$s_1 = \{\varepsilon_1, \varepsilon_3\}$
$s_2 = \{\varepsilon_1, \varepsilon_4\}$
$s_3 = \{\varepsilon_2, \varepsilon_3, \varepsilon_4\}$




Through appropriately assigning the value of $P_s$, $C_s$, $P_M$, $C_M$, $P_e$, $C_e$, we can show that the optimal first edge to test is $M$ if and only if there does not exist a set cover $\mathcal{C}$ with $|\mathcal{C}| \leq k$. $\square$

# Outline

# Markov Decision Process Framework

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University



## Definition (Markov Decision Process)

A mathematical model for modeling decision making under uncertain situations. A Markov Decision Process (MDP) model contains:

- State Space
- Decision Epochs
- Action Sets
- Transition Probabilities and Rewards
- Strategy (with certain expected total reward)

Markov Property: the effects of an action only depends on the current state and the action itself.

# Mapping the Problem into MDP



The correspondence between the key elements:

- State Space: the set of temporary states
  - $\mathcal{S} = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \ldots \mathcal{S}_{|E|}$
- Decision Epochs
- Action Sets: testing edges or terminating
  - $A_{\mathbf{s}}$, $A = \bigcup_{s \in \mathcal{S}} A_{\mathbf{s}} = E \cup \{\perp\}$
- Transition Probabilities and Rewards
  - From $\mathbf{s}$ to $\mathbf{s} \cdot e$
  - From $\mathbf{s}$ to $\mathbf{s} \backslash e$
- Strategy

# Exact Algorithm Based on Dynamic Programming

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University



**Algorithm 1** The MDP-based Exact Algorithm

**Input:** Uncertain graph $\mathcal{G}(V, E, p, c)$, source $s$, destination $t$
**Output:** The optimal testing strategy $\pi$

```
Initialize:  u_π(s) = 0, for all s ∈ S_|E|
for i = |E| to 0 do
    for All s in S_i do
        if s is a terminating state then
            u_π(s) := 0, π(s) := ⊥.
        else
            e* := argmax_{e∈A_s} {−c(e) + p(e)u_π(s·e)
                  +(1 − p(e))u_π(s\e)},
            u_π(s) := −c(e*) + p(e*)u_π(s·e*)
                  +(1 − p(e*))u_π(s\e*),
            π(s) := e*.
return π
```

Backward Induction

# A Simple Greedy Approach



A strategy that tests the edges following the ascending order of costs is an $O(|E|)$-approximation.

**Theorem**

*Given an uncertain graph $\mathcal{G}(V, E, p, c)$ and two nodes $s$ and $t$ as source and destination, let $\pi$ be a strategy that tests the edges in $E$ according to their costs sorted in an increasing order. Then, $Cost(\pi) \leq |E| \cdot Cost(\pi^*)$, where $\pi^*$ is the optimal strategy*

# Adaptive Submodular Algorithm – Preliminaries

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University



**Definition (Extension)**

For two temporary states $\mathbf{a}, \mathbf{b} \in \mathcal{S}$, we say $\mathbf{a}$ is an extension of $\mathbf{b}$, written as $\mathbf{a} \sim \mathbf{b}$ if $\mathbf{a}_i = \mathbf{b}_i$ for all $\mathbf{b}_i \neq *$.

**Definition (Function on Temporary States)**

Let $g : \mathcal{S} \mapsto \mathbb{N}$ be a utility function on temporary states.

- $g$ is *monotonically increasing* if $g(\mathbf{s}') - g(\mathbf{s}) \geq 0$ for all $s \in \mathcal{S}, \mathbf{s}' \sim \mathbf{s}$.
- $g$ is *adaptive submodular* if $g(\mathbf{s} \cdot e) - g(\mathbf{s}) \geq g(\mathbf{s}' \cdot e) - g(\mathbf{s}')$ and $g(\mathbf{s} \backslash e) - g(\mathbf{s}) \geq g(\mathbf{s}' \backslash e) - g(\mathbf{s}')$ whenever $\mathbf{s}' \sim \mathbf{s}$ and $\mathbf{s}_e = \mathbf{s}'_e = *$.

# Adaptive Submodular Algorithm – Utility Function



The utility function $g$ should be *assignment feasible* in the sense that:

- $g(*, *, \ldots, *) = 0$.
- $g(\mathsf{s}) = Q$ iff $\mathsf{s}$ is a terminating state, where $Q$ is the target value.

**Lemma (The Adaptive Submodular Framework [2])**

*Each time choosing (testing) the edge with the maximum expected gain:*

$$\frac{p(e)g(\mathsf{s} \cdot e) + (1 - p(e))g(\mathsf{s} \backslash e) - g(\mathsf{s})}{c(e)}$$

*yields an $O(\ln |Q|)$-approximation.*

[2] D. Golovin and A. Krause, "Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization"

# Adaptive Submodular Algorithm – Utility Function

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University



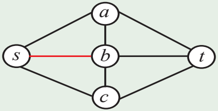
## Definition (The Design of Utility Function $g$)

Define $\mathcal{P}$ and $\mathcal{C}$ as the collection of $s$-$t$ paths and $s$-$t$ cuts in $\mathcal{G}$ respectively. Define $\mathcal{P}_e$ and $\mathcal{C}_e$ as the collection of $s$-$t$ paths and $s$-$t$ cuts that edge $e$ lies on in $\mathcal{G}$ respectively. We have,

$$g_p(\mathbf{s}) = |\bigcup_{e:\mathbf{s}_e=0} \mathcal{P}_e|, \quad g_c(\mathbf{s}) = |\bigcup_{e:\mathbf{s}_e=1} \mathcal{C}_e|,$$

$$g(\mathbf{s}) = |\mathcal{P}||\mathcal{C}| - (|\mathcal{P}| - g_p(\mathbf{s}))(|\mathcal{C}| - g_c(\mathbf{s})).$$

## Example




- Set of paths
- Set of cuts

# Adaptive Submodular Algorithm

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University



Determine the strategy sequentially

**Algorithm 2** The Adaptive Submodular Algorithm

**Input:** Uncertain graph $\mathcal{G}(V, E, p, c)$, source and destination.

**Output:** Testing strategy $\pi$

**Initialize:** Current state $\mathbf{s} := (*, *, \ldots, *)$, The set of tested edges $E_\pi$ as an empty set.

**Repeat** until $s$ becomes a terminating state.

$e^* := \arg\max_{e \in E \setminus E_\pi} \{ \frac{p(e)g(\mathbf{s} \cdot e) + (1-p(e))g(\mathbf{s} \setminus e) - g(\mathbf{s})}{c(e)} \}$.

$E_\pi := E_\pi \cup \{e^*\}$, *test* $e^*$ and observe the outcome.

**if** edge $e^*$ exists **then**

$\quad \mathbf{s}_{e^*} := 1$

**else**

$\quad \mathbf{s}_{e^*} := 0$

Test the edge with the maximum marginal gain

# Adaptive Submodular Framework – Performance



**Theorem**

*$g$ is monotonically increasing, adaptive submodular, and assignment feasible.*

Proof:

By the definition of $g$.

**Theorem**

*The Adaptive Submodular Algorithm yields an $O(\ln |Q|) = O(\ln |P||C|)$-approximation.*

Proof:

By the Adaptive Submodular framework proposed by Golovin and Krause [3].

---

[3] D. Golovin and A. Krause, "Adaptive Submodularity: Theory and Applications in Active Learning and Stochastic Optimization"

# Outline

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University

# Experiment Settings



**Experiment Datasets:**

- Citation Networks (273751 nodes, 993025 edges)
- Internet Peer to Peer Networks (5000 nodes, 16469 edges)
- Twitter Ego Networks (213 nodes, 17930 edges)

**Parameters Assignments:**

- Costs: drawn from $\mathcal{N}(50, 100)$
- Probabilities: $p(e = (x, y)) = |\Gamma(x) \cap \Gamma(y)|/|\Gamma(x) \cup \Gamma(y)|$[4]

**Performance Metric:**

- The expected costs are approximated by the averaged costs on 1000 underlying graphs.

[4] $\Gamma(x)$ denotes the set of neighbors of $x$.

# Algorithms Involved in Comparisons

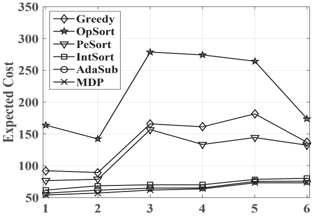
- Greedy Algorithm (Greedy)
- Adaptive Submodular Algorithm (AdaSub)
- Optimistic Sort Algorithm (OpSort): follow an increasing order of $c/p$
- Pessimistic Sort Algorithm (PeSort): follow an increasing order of $c/(1-p)$
- Intersection Sort Algorithm (IntSort): test the edge with the minimum cost that lies on the intersection of a shortest $s$-$t$ path and a minimum $s$-$t$ cut.
- MDP-based Algorithm (MDP): only on a sequence of small subnetworks.

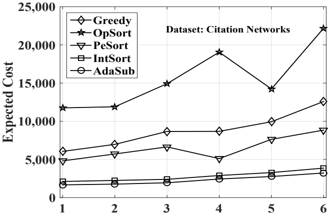
# Experiment Results

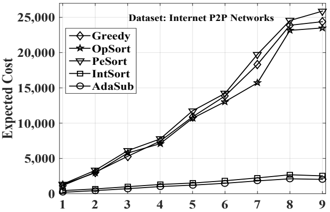
Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

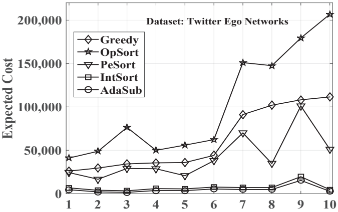
Shanghai Jiao Tong University





# Outline

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University

# Conclusion



- Considering the prevalent uncertainties in networks, we formally define the source-destination connectivity determination problem in uncertain networks.
- By proving the related hardness results, we demonstrate the computational complexity of the problem.
- We propose exact algorithm and efficient approximation algorithms for solving the problem.
- Extensive experiment results demonstrate the effectiveness of our proposed algorithms and the superiority over other heuristics.

Xinzhe Fu, Zhiying Xu, Qianyang Peng, Luoyi Fu and Xinbing Wang

Shanghai Jiao Tong University



# Thank You!