CS630 Graduate Algorithms

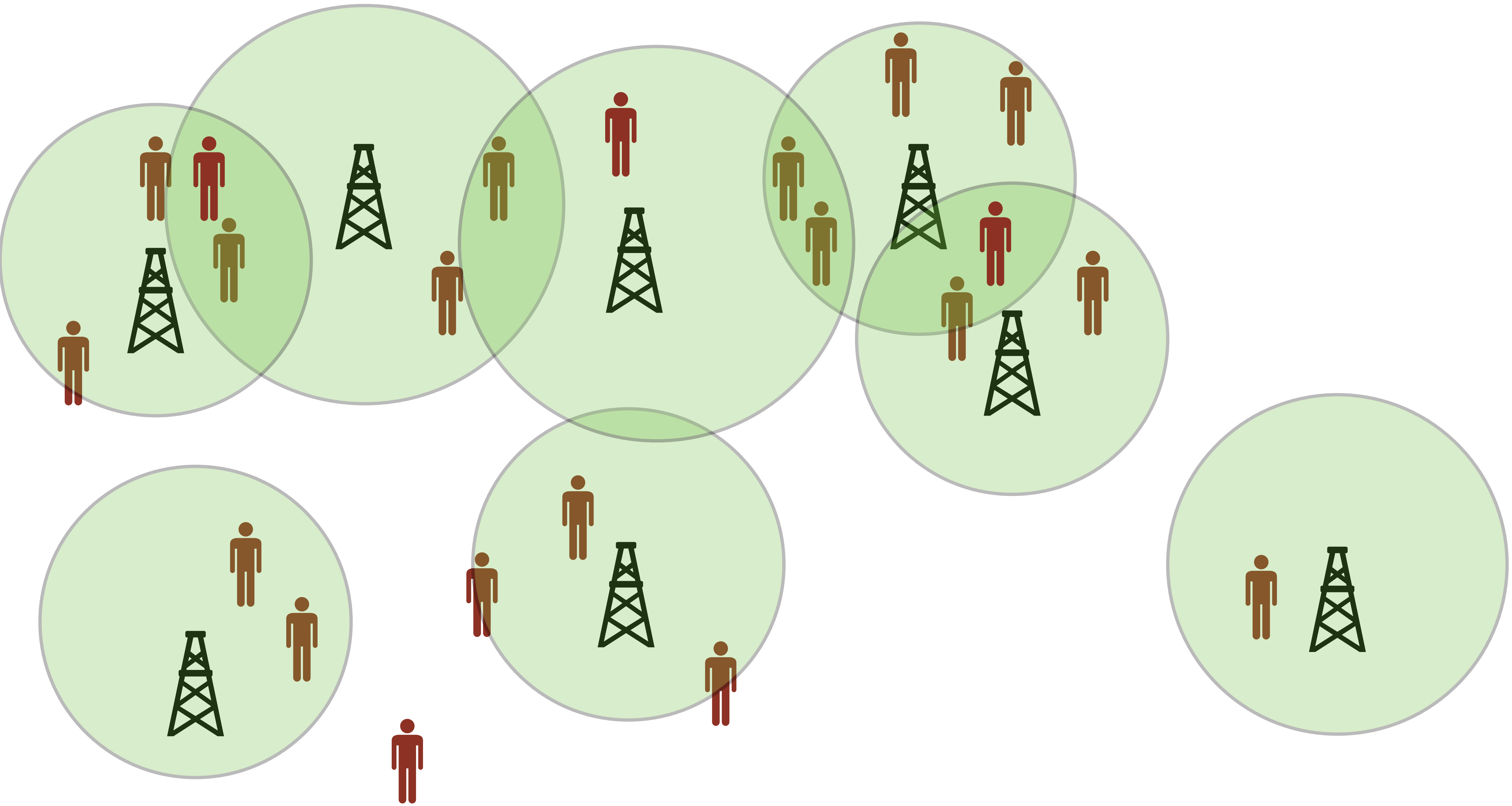
October 1, 2024

by Dora Erdos and Jeffrey Considine

- Monotone submodular functions
  - coverage problems
  - greedy optimization

Radio stations

Each station has a broadcast range. Where to broadcast from to reach the maximum audience, if we only have a budget to broadcast from k stations?



Max k-Coverage Problem

Set Cover: Given a universe $U = \{u_1, u_2, \ldots, u_n\}$ of elements and a collection $S = \{S_1, S_2, \ldots, S_m\}$ of subsets of *U*, find a minimum number of the sets in *S* such that their union contains *every* item in U.

Max k-Coverage Problem: Given a universe $U = \{u_1, u_2, \ldots, u_n\}$ of elements and a collection $S = \{S_1, S_2, \ldots, S_m\}$ of subsets of *U* and an integer *k*, find *k* sets in *S* such that the number of elements covered by their union is *maximized*.

Max k-coverage greedy algorithm

Algorithm:

Algorithm 1: GreedySC($U, S_1, \ldots S_m$)

$X \leftarrow U$ `/* uncovered elements in U */`
$C \leftarrow$ empty set of subsets;
while X *is not empty* **do**
    Select S_i that covers the most items in X;
    $C \leftarrow C \cup S_i$;
    $X \leftarrow X \setminus S_i$;
return C;

Max k-coverage greedy algorithm

Algorithm: for k iterations select the set that covers the most additional elements.

Algorithm 1: GreedySC($U, S_1, \ldots S_m$ k)

$X \leftarrow U$ /* uncovered elements in U */

$C \leftarrow$ empty set of subsets;

~~**while** *X is not empty* **do**~~ **For** j=1…k **do**

  Select S_i that covers the most items in X;

  $C \leftarrow C \cup S_i$;

  $X \leftarrow X \setminus S_i$;

return C;

Max k-coverage approximation

calculus: $\left(1-\frac{1}{t}\right)^t < \frac{1}{e}$

Theorem: The greedy algorithm has approximation factor $1-\left(1-\frac{1}{k}\right)^k > 1-\frac{1}{e} \approx 63\,\%$

meaning:

remember: for the Set Cover problem, if the optimal solution uses L sets, then the approximation factor of the greedy algorithm is $L \cdot \ln n$

What’s the difference?

Max k-coverage approximation

Theorem: The greedy algorithm has approximation factor $1-\left(1-\frac{1}{k}\right)^k > 1-\frac{1}{e} \approx 63\,\%$ meaning:

- the greedy algorithm covers at least ~63% of the items that an optimal cover with k sets would

remember: for the Set Cover problem, if the optimal solution uses L sets, then the approximation factor of the greedy algorithm is $L \cdot \ln n$

What's the difference?

- for the k-coverage problem the approximation has *constant* ratio, for set cover it depends on the *input size* n
- (note that k is not constant, it's part of the input!)
- intuitively the first k sets cover larger ratio of the points, as we select more sets the marginal gain of extra elements covered is diminishing

Max k-coverage approximation

calculus: $t > 0 : \left(1 - \frac{1}{t}\right)^t < \frac{1}{e}$

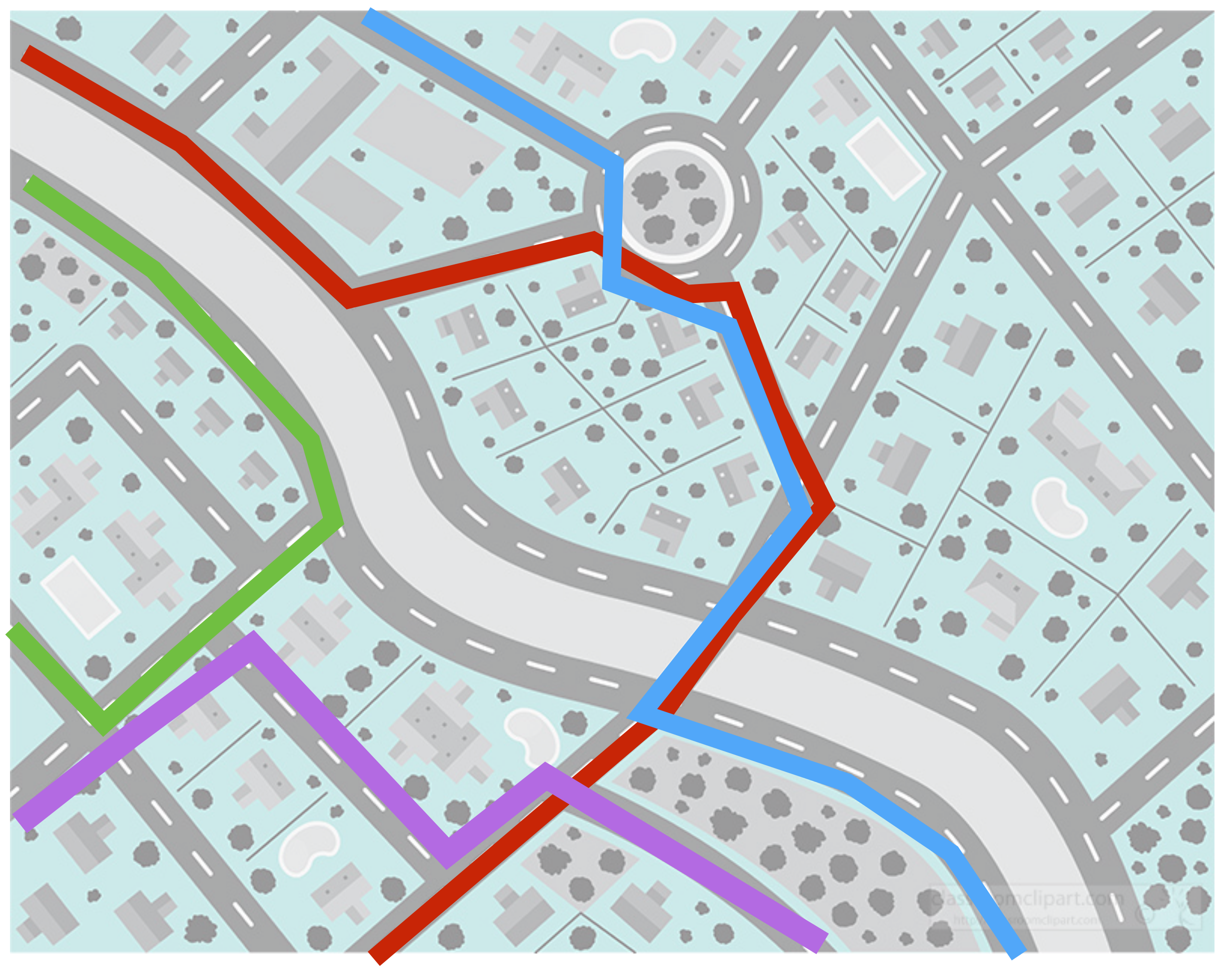
Theorem: the greedy algorithm has approximation factor $1 - \left(1 - \frac{1}{k}\right)^k > 1 - \frac{1}{e} \approx 63\,\%$

proof:

- Z is the set of items covered by the optimal solution
- among the k sets in the optimal solution, there is at least one set that covers1/k fraction of Z
- since Greedy-k-SC selects the largest set, it also covers at least Z/k items
- after the first iteration at most $Z\left(1 - \frac{1}{k}\right)$ remain uncovered
- since greedy selects the largest marginal gain, it covers at least 1/k of the remaining elements in Z in each iteration: $Z\left(1 - \frac{1}{k}\right)^k$
- after k rounds there are at least $Z - Z\left(1 - \frac{1}{k}\right)^k = Z\left(1 - \left(1 - \frac{1}{k}\right)^k\right)$ points covered

Detecting Potholes

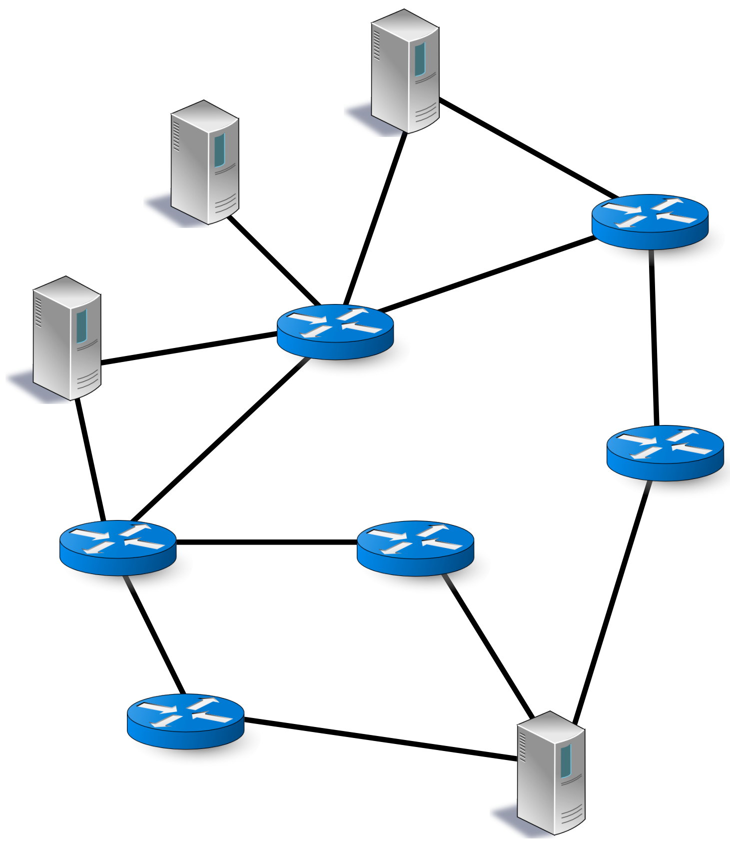
- Mount sensors on busses to detect potholes in the road along their routes
- Bus routes overlap so different routes may cover the same streets
- Given a small budget of sensors, which bus routes should we equip with sensors to detect as many potholes as possible?



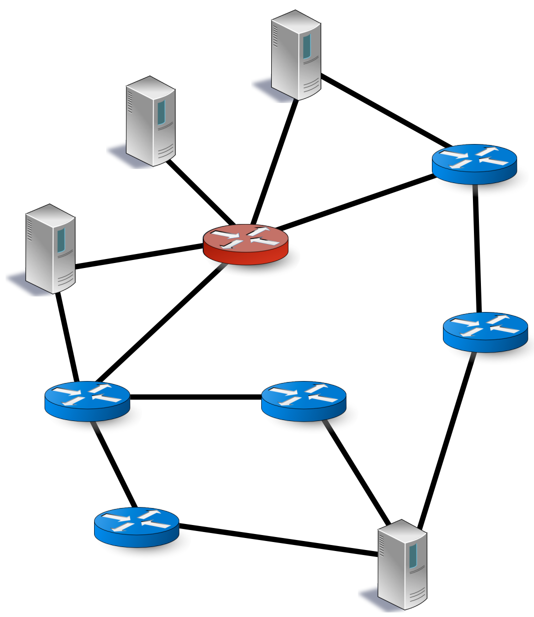
Ali, Dyo [2017]: https://www.scitepress.org/Papers/2017/64698/64698.pdf

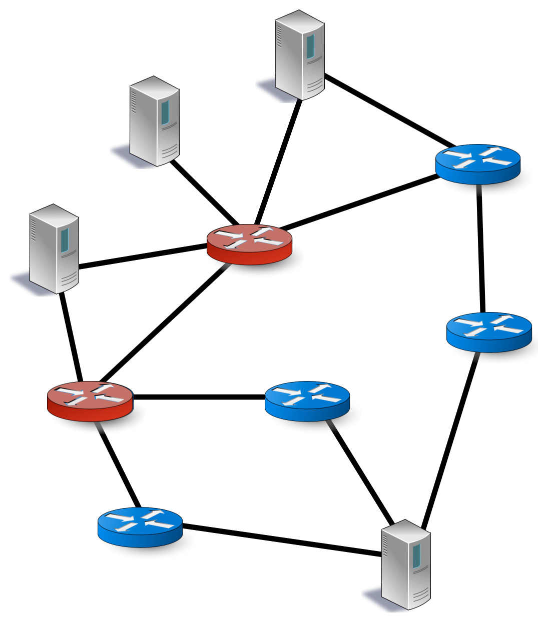
Covering shortest paths

Select two routers that together cover the most shortest paths.



Greedy: select router that covers most additional paths

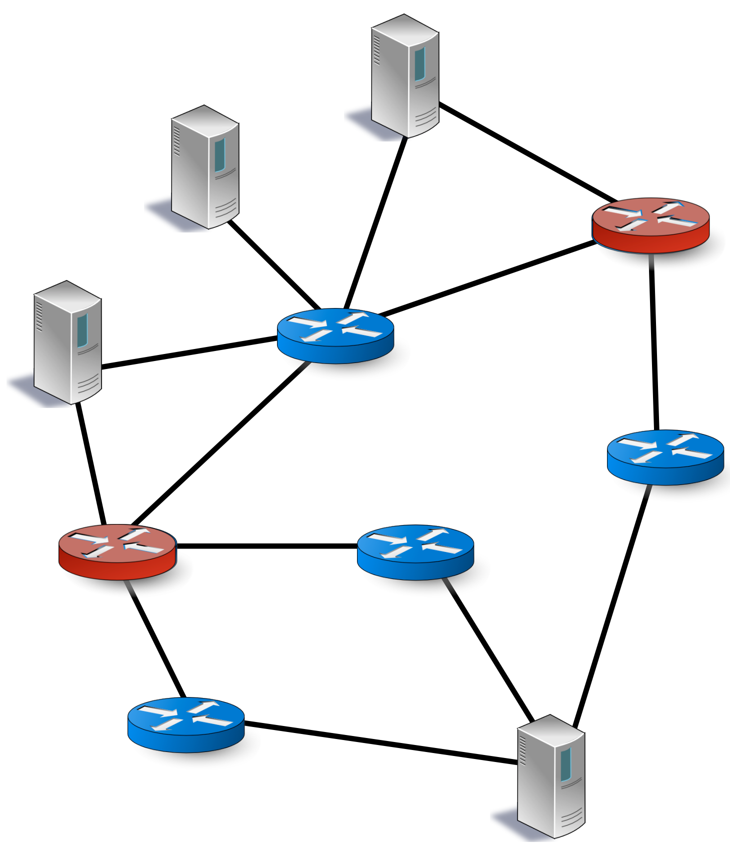




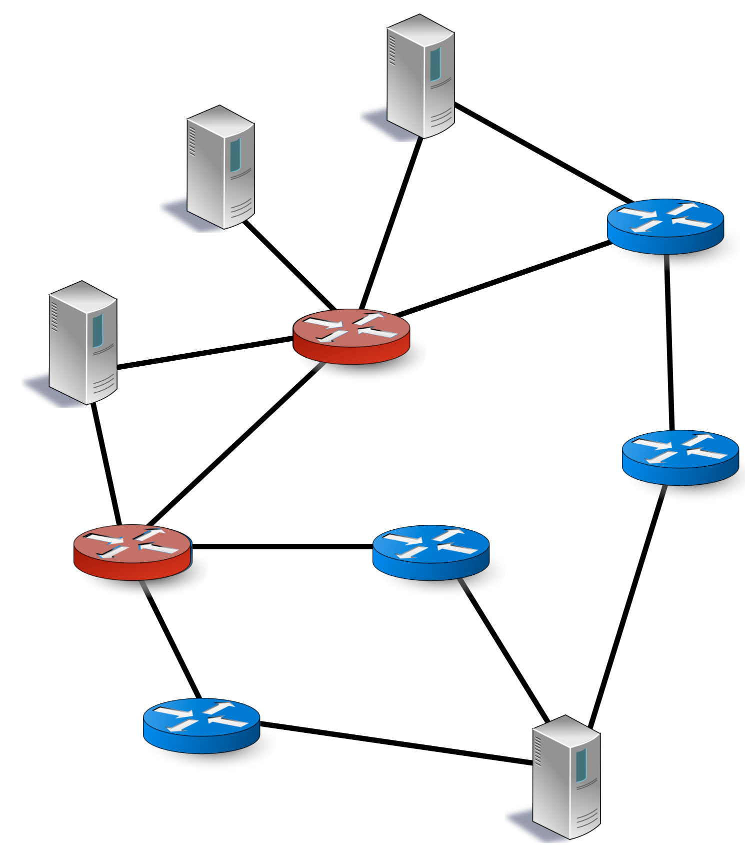
Covering shortest paths

Select two routers that together cover the most shortest paths.

Optimal solution



Greedy: select router that covers most additional paths



Bulk Pricing

BULK PRICE ELIGIBLE $3.85 /piece

Buy **50** or more **$3.27**

Model# 769887219614

2 in. x 4 in. x 96 in. Prime Kiln-Dried Whitewood Stud

Pickup
2,500 in stock at Watertown

Delivery
Unavailable

BULK PRICE ELIGIBLE $3.85

Buy **50** or more **$3.27**

(4734)

Model# 058449

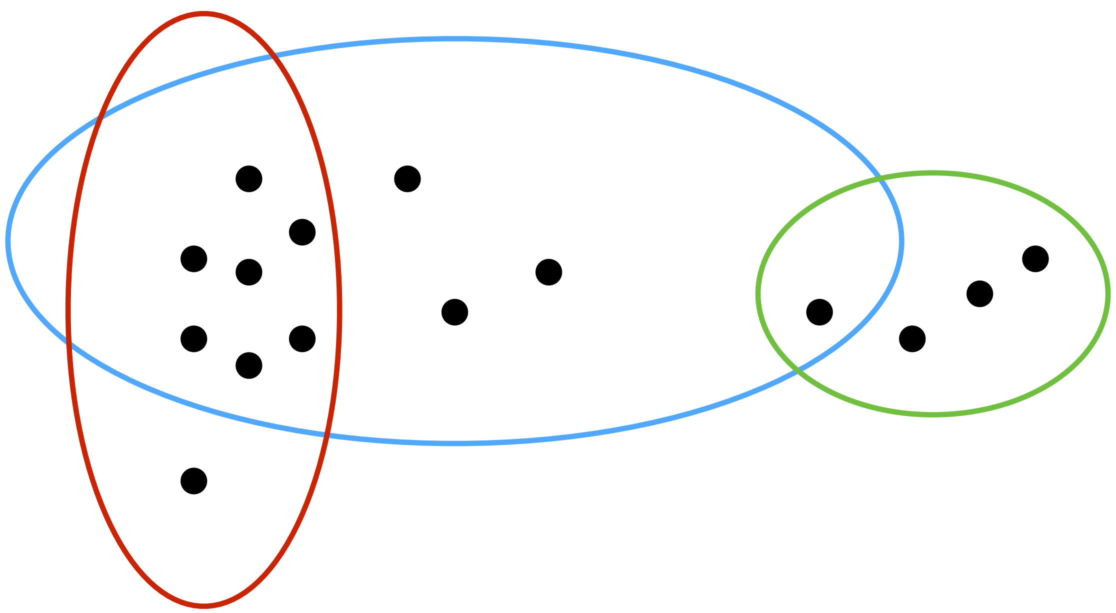
2 in. x 4 in. x 8 ft. Prime Stud

Pickup
2,500 in stock at Watertown

Delivery
Scheduled Delivery

Greedy algorithm for coverage problems

- in each problem we assign some positive value to a set of objects
- diminishing returns
  - the additional benefit of one more set is less as more sets are selected



- natural greedy algorithm:
  - For k iterations repeatedly select the object with largest gain towards our objective function.

Objective function for coverage problems

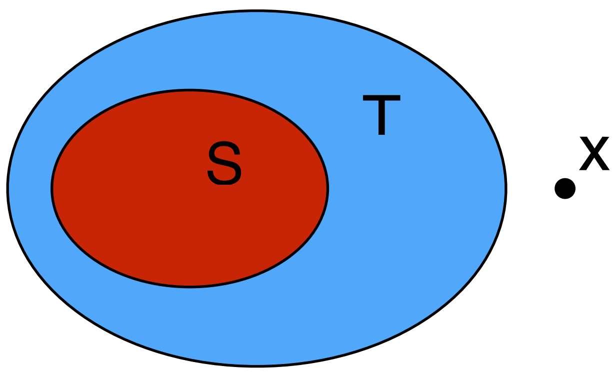
set function: a function $f : 2^X \to \mathbb{R}_+$ that takes sets as input and outputs numbers

- 2^X is the set of all subsets of X, think of the set represented as a bit vector

Submodular functions

The set function $f : 2^X \to \mathbb{R}_+$ is submodular if for every $S \subset T \subset X$ and $x \in X \backslash T$

$$F(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$$

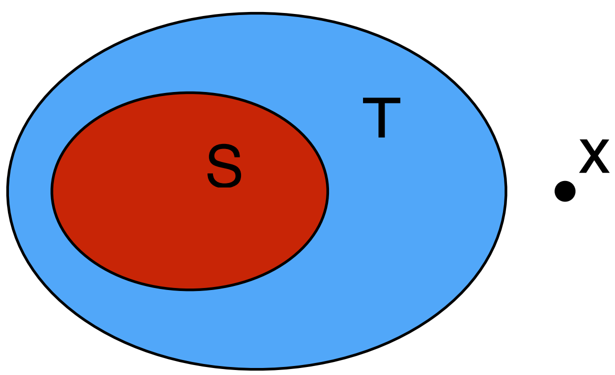


f is monotone increasing if for every $S \subseteq T$ we have $f(S) \leq f(T)$

Submodular functions

The set function $f : 2^X \to \mathbb{R}_+$ is submodular if for every $S \subset T \subset X$ and $x \in X \backslash T$

$$F(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$$



f is monotone increasing if for every $S \subseteq T$ we have $f(S) \leq f(T)$

examples of monotone submodular functions:

$f(S) = c \cdot |S|$

$f(S) = \sum_{i \text{ in } S} w_i$ where $w_i \geq 0$ linear functions

budget-additive $f(S) = \min\{B, \sum_{i \text{ in } S} w_i\}$

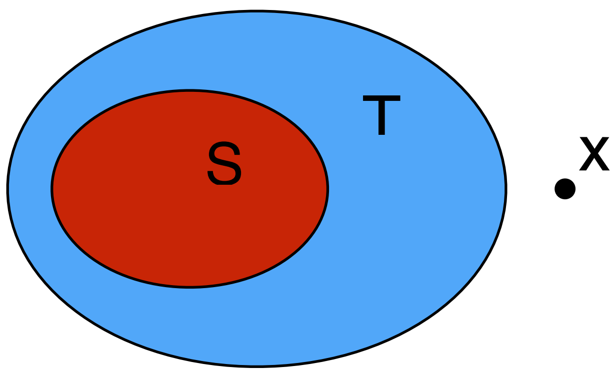
coverage functions - items, paths, sets, …

entropy of random variables, information gain

Submodular functions

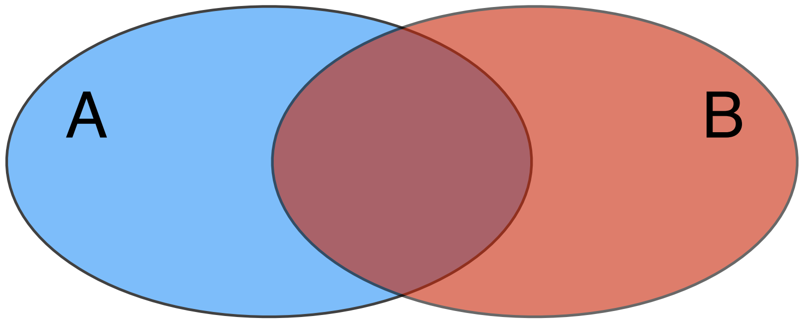
The set function $f : 2^X \to \mathbb{R}_+$ is submodular if for every $S \subset T \subset X$ and $x \in X \backslash T$

$$F(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$$



Equivalent definition: f is submodular if for every $A, B \subset X$

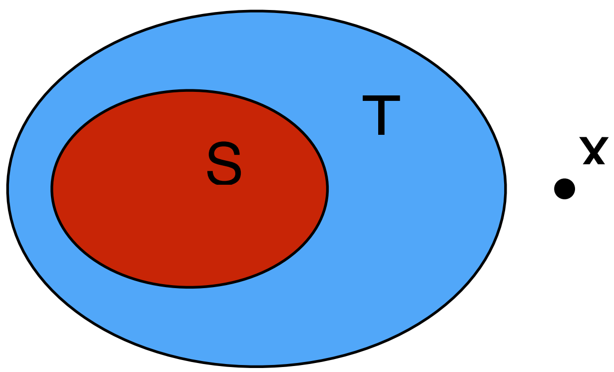
$$f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$$



Submodular functions

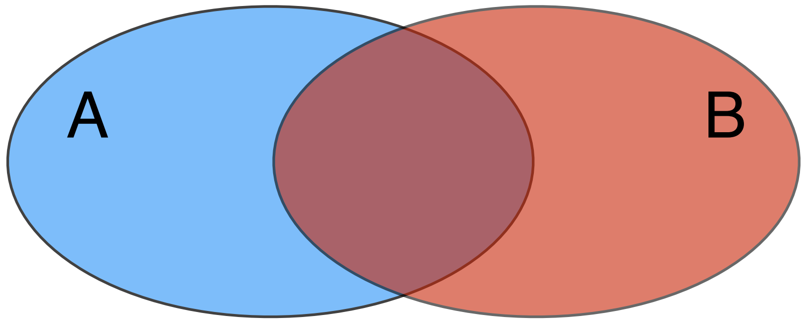
The set function $f : 2^X \to \mathbb{R}_+$ is submodular if for every $S \subset T \subset X$ and $x \in X \backslash T$

$$F(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$$



Equivalent definition: f is submodular if for every $A, B \subset X$

$$f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$$



proof:

$\Rightarrow$setting $A = T, B = S \cup \{x\}$ we get the formula

$\Leftarrow$ use $A \cap B \subseteq B$ inductively apply the inequality to each element in $A \backslash B$ to get

$F(A \cup B) - F(B) \leq F(A) - F(A \cap B)$

Greedy algorithm for monotone submodular functions

Suppose that the objective function f of some maximization problem is *monotone submodular*.

Algorithm 1: GreedySubmodular$(X, S_1, \ldots, S_m, kf(\ )))$

/* X is the universe of elements, S_i are subsets, k is an int, $f(\ )$ is a submodular function */

$C \leftarrow \emptyset$ /* $C \subseteq X$ currently covered items in X */

for $i = 1$ *to* k **do**

  find i to maximize $f(C \cup S_i) - f(C)$;

  $C \leftarrow C \cup \{S_i\}$;

return C

submodular functions and complexity

problem type	maximization	minimization
unconstrained	NP-hard some approximations	polynomial via convex optimization
constrained - select k	NP-hard constant approx ration (1-1/e)	usually NP-hard to approximate

Greedy approximation factor

Theorem [Nemhauser, Wolsey, Fisher, 1978]: For any maximization problem with a monotone submodular objective function the greedy algorithm yields a (1-1/e)-approximation.

Why is this useful?

Greedy approximation factor

Theorem [Nemhauser, Wolsey, Fisher, 1978]: For any maximization problem with a monotone submodular objective function the greedy algorithm yields a (1-1/e)-approximation.

Why is this useful?

- for optimization problems - which are often NP-C - the most simple greedy algorithm is a pretty good optimization.
  - "pretty good" = constant!
- it's enough to prove that the function the problem is maximizing is indeed monotone submodular.

Product adoption via viral marketing

- Example of Viral Marketing: Hotmail.com

- Jul 1996: Hotmail.com started service
- Aug 1996: 20K subscribers
- Dec 1996: 100K
- Jan 1997: 1 million
- Jul 1998: 12 million

Bought by Microsoft for $400 million

Marketing: At the end of each email sent there was
a message to subscribe to Hotmail.com
“Get your free email at Hotmail"

Models of influence in networks

Intuition: fraction of friends that have already adopted the product influence the likelihood of a node becoming an adopter.

Problem:

Select an initial group of k influencers so that - given some propagation model - the expected number of converts is maximized.

Granovetter: Threshold Models for Collective Behavior (1978)

Domingos, Richardson: Mining the Network value of Customers (2001) Mining Knowledge-sharing Sites for Viral Marketing

Kempe, Kleinberg, Tardos: Maximizing the Spread of Influence Through a Social Network (2003)

Models of influence in networks

Intuition: fraction of friends that have already adopted the product influence the likelihood of a node becoming an adopter.

Models:

- Linear Threshold Model
- Independent Cascade Model

Problem:

Select an initial group of k influencers so that - assuming one of the above propagation models - the expected number of converts is maximized.

Linear threshold model

Setup: Given a graph G(V,E)

- there is an initial set of active nodes (called seeds)
- once a node becomes active it will possible activate its neighbors

Linear threshold model

- each node *v* has an *activation threshold* $\theta_v \in [0,1]$

- node *v* is influenced by each neighbor *w* by some weight $b_{v,w}$ such that

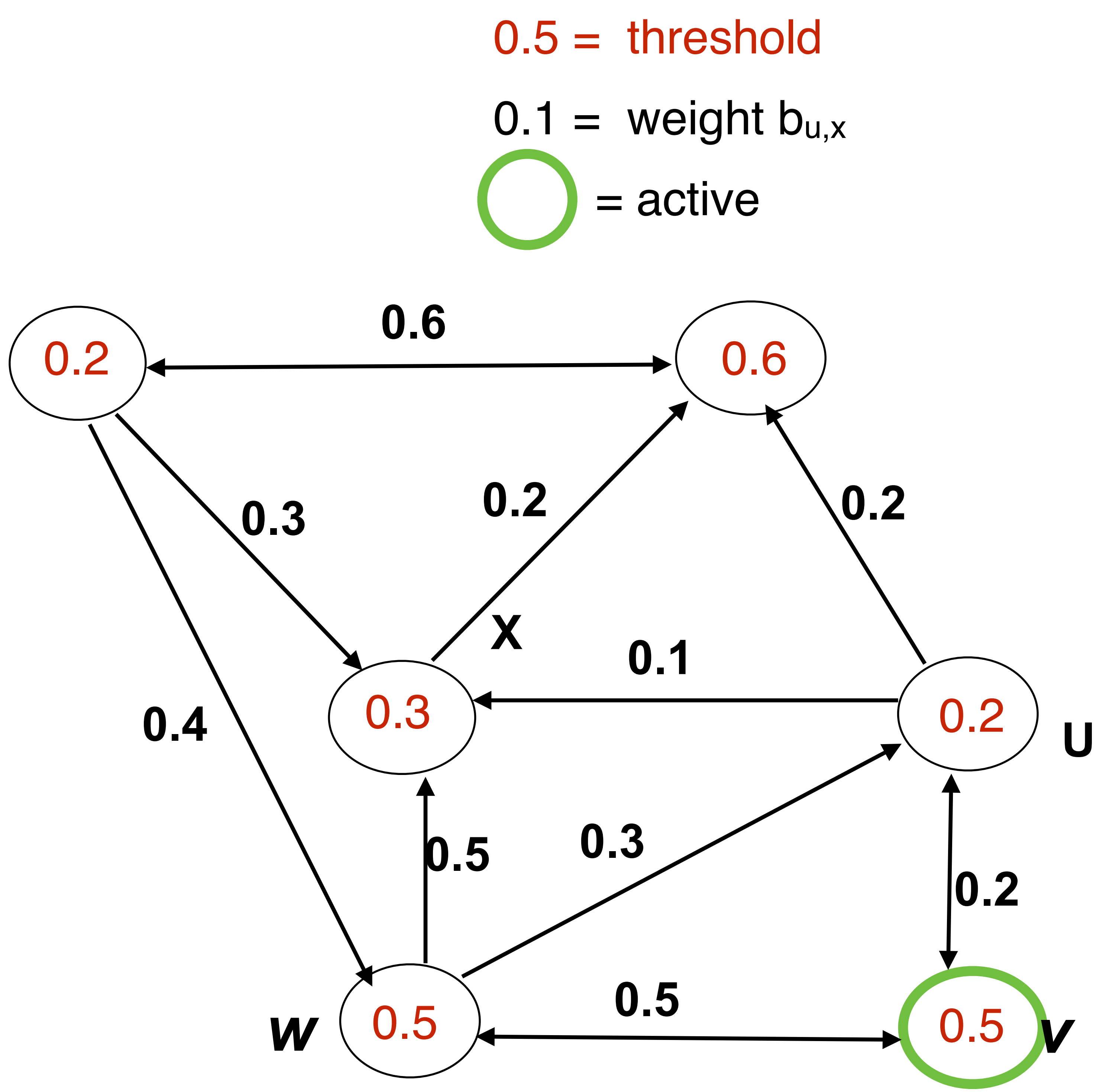
$$\sum_{w \text{ is neighbor of } v} b_{v,w} \leq 1$$

- *v* becomes active iff

$$\sum_{\substack{w \text{ is active} \\ w \text{ is neighbor of } v}} b_{v,w} \geq \theta_v$$

input: G(V,E), θ_v, $b_{v,w}$

Example


0.5 = threshold
0.1 = weight $b_{u,x}$
= active
0.6
0.2
0.6
0.3
0.2
0.2
X
0.1
0.4
0.3
0.2
U
0.5
0.3
0.2
0.5
W
0.5
0.5
V

Independent cascade model

Setup: Given a graph G(V,E)

- there is an initial set of active nodes (called seeds)
- once a node becomes active it will possible activate its neighbors

Independent cascade models

- when a node *v* becomes active at time t it has a *single* chance to activate its neighbor *w*
- the activation succeeds with probability $p_{v,w}$

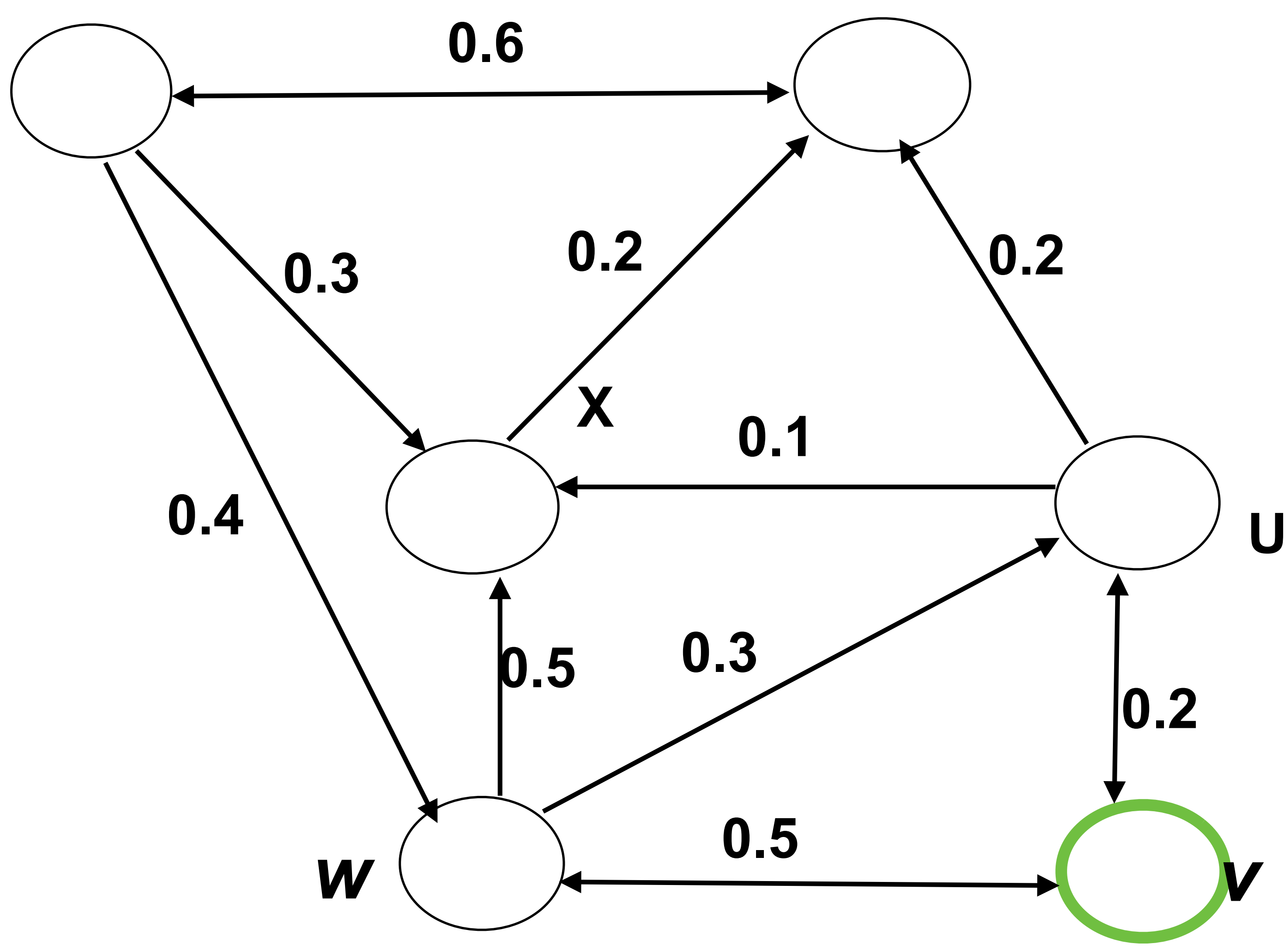
- note: if *w* has multiple active neighbors, each attempts to activate *w* independent of each other

input: G(V,E), $p_{v,w}$

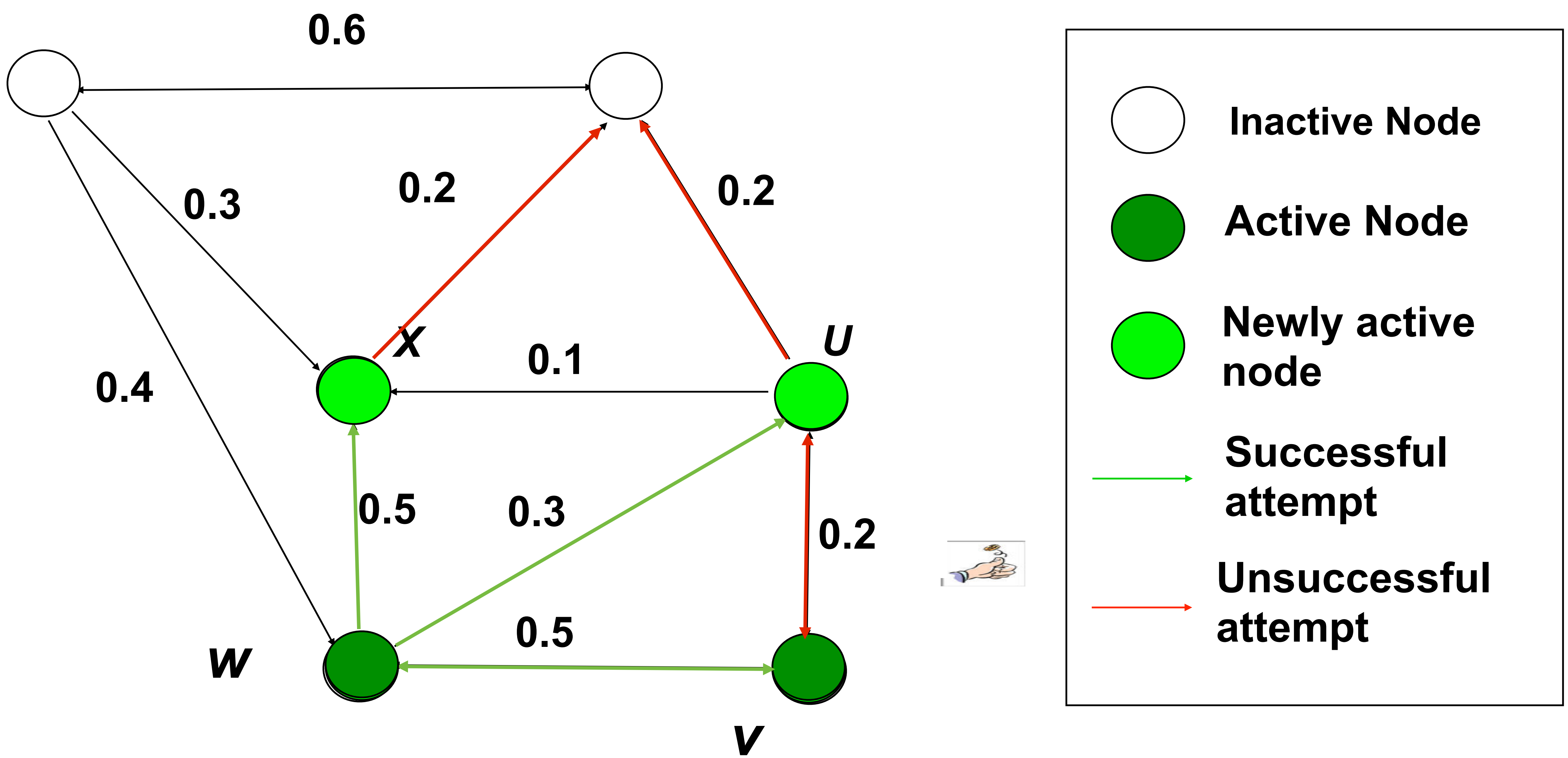
Example

0.1 = probabilitu $p_{u,x}$

= active



Example



Stop!

Influence maximization problem

Let G(V,E) be a graph

the influence f(S) of node set S is the *expected* number of active nodes, given one of the two models, if S is the initially active set.

Influence maximization problem: Given as input G(V,E), one of the models with parameters and budget k, find a set S of k nodes with maximum influence f(S)

What can we say about the objective function f(S)?

Influence maximization problem

Let G(V,E) be a graph

the influence f(S) of node set S is the *expected* number of active nodes, given one of the two models, if S is the initially active set.

Influence maximization problem: Given as input G(V,E), one of the models with parameters and budget k, find a set S of k nodes with maximum influence f(S)

What can we say about the objective function f(S)?

- monotone increasing — adding one more node to S can only increase the influence
- submodular — adding an additional node to a smaller set S has larger impact on the spread
    - how can we prove this given that f(S) is a probabilistic function? → use expected value?

$$f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T) \; for \; S \subset T \subset V$$

Greedy optimization

Suppose we can prove that the probabilistic function f(S) is submodular

Greedy algorithm:

1. start $S = \emptyset$
2. for k iteration ad v such that in expectation $f(S \cup \{v\}) - f(S)\ is\ max$

Theorem: this greedy algorithm yields a (1-1/e)-approximation

The expected number of activate nodes, when the seeds are selected with the greedy algorithm are ~63% of the expected number for the best seed set.

Proof of submodularity for random independent cascade model

cascade process: if a node v is activated, then flip a coin for each adjacent edge (v,w) to activate w with probability $p_{v,w}$

instead, generate "possible world" G_r

- iterate over each edge of G first
- for each (v,w) flip a coin and keep the edge with probability $p_{v,w}$
- now we have a deterministic graph - an instance of the random graph

active nodes at the end of the diffusion are the ones *reachable* from the seeds in this generated graph

- reachability is submodular - the seeds are nodes that "cover" the paths

conclusion: for any one specific instance of the random model, the influence function f(S) is submodular.

Proof of submodularity for random independent cascade model

cascade process: if a node v is activated, then flip a coin for each adjacent edge (v,w) to activate w with probability $p_{v,w}$

method to simulate the process for experiments!

instead, generate "possible world" G_r

- iterate over each edge of G first
- for each (v,w) flip a coin and keep the edge with probability $p_{v,w}$
- now we have a deterministic graph - an instance of the random graph

active nodes at the end of the diffusion are the ones *reachable* from the seeds in this generated graph

- reachability is submodular - the seeds are nodes that "cover" the paths

conclusion: for any one specific instance of the random model, the influence function f(S) is submodular.

Proof of submodularity for random independent cascade model

conclusion: for any one specific instance of the random model, the influence function f(S) is submodular.

fact: non-negative linear combination of submodular functions is also submodular

expected influence in G:

S = set of seed nodes

G_r = random graph instance

$A(G_r)$ = set of active nodes in G_r given S as seed set

$$f(S) = \sum_{G_r} Pr(G_r) \cdot |A(G_r)|$$

Similar proof can be done for the linear threshold model

Implementing greedy

In practice, how can we implement an optimization algorithm with a random objective function?

Still an open question how to compute efficiently

- Kempe et al.: neat trick called “lazy greedy updates” → only update coverage computation for top few candidate

We get very good estimations by simulation

- repeat many times:
  - generate G_r
  - find the optimal set S on G_r using the deterministic (set cover-style) greedy algorithm
- influence can be computed as the average activation over the many runs

Experimental results - Kempe, Kleinberg, Tardos [2003]

Data:

co-authorship graph in papers on arXiv in the high-energy physics theory section

graph G(V,E)

V: authors

E: there is an edge (v,w) if persons v and w have written a paper together

|V| = 10748, |E| = 53000

model parameters

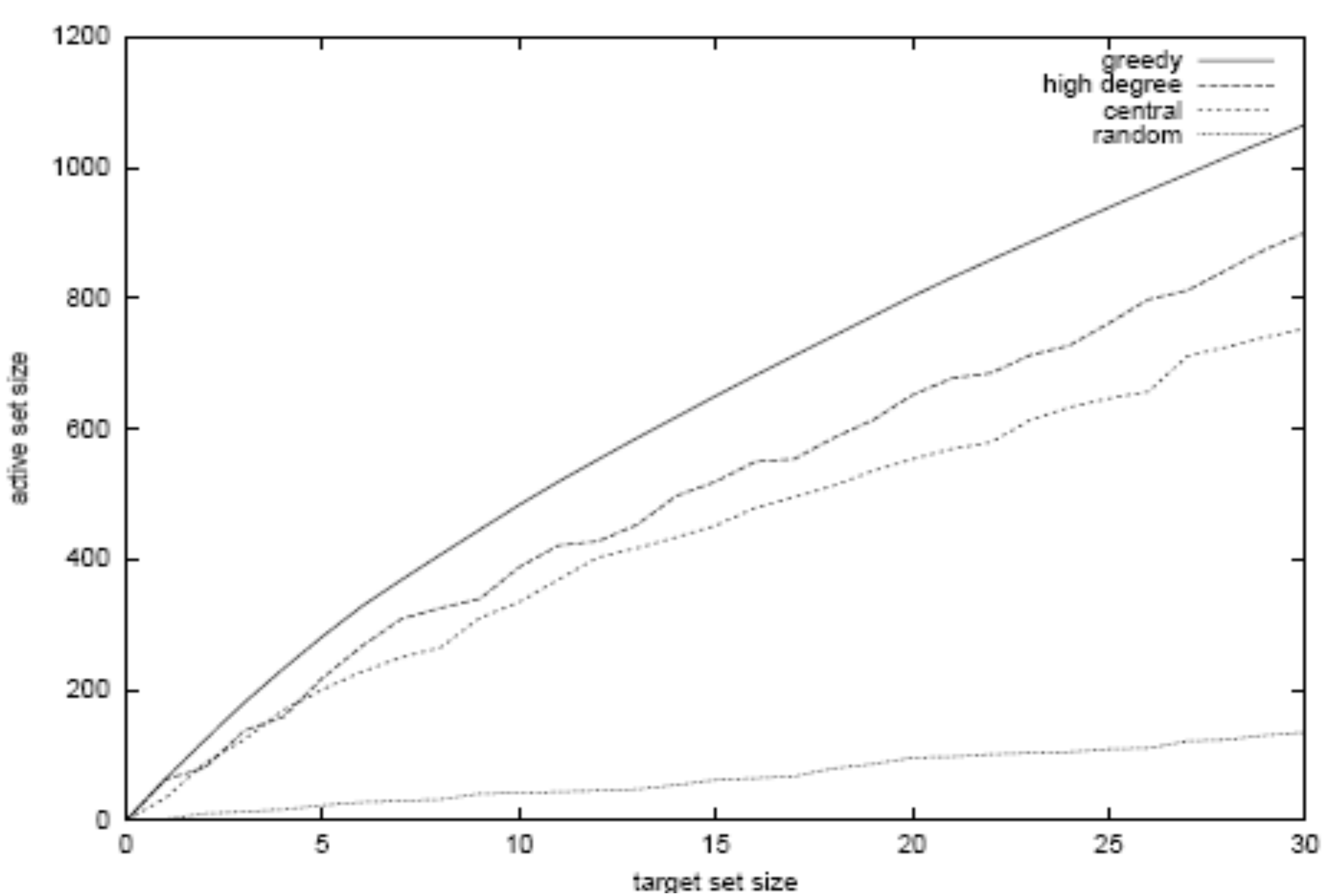
- *linear threshold:* based on multiplicity of edges
  - fraction of papers co-authored $c_{v,w}$ divided by all papers by this person d_v

$$b_{v,w} = \frac{c_{v,w}}{d_v}$$

- *independent cascade:* activation probabilities chosen uniform at random

Experimental results - Kempe, Kleinberg, Tardos [2003]

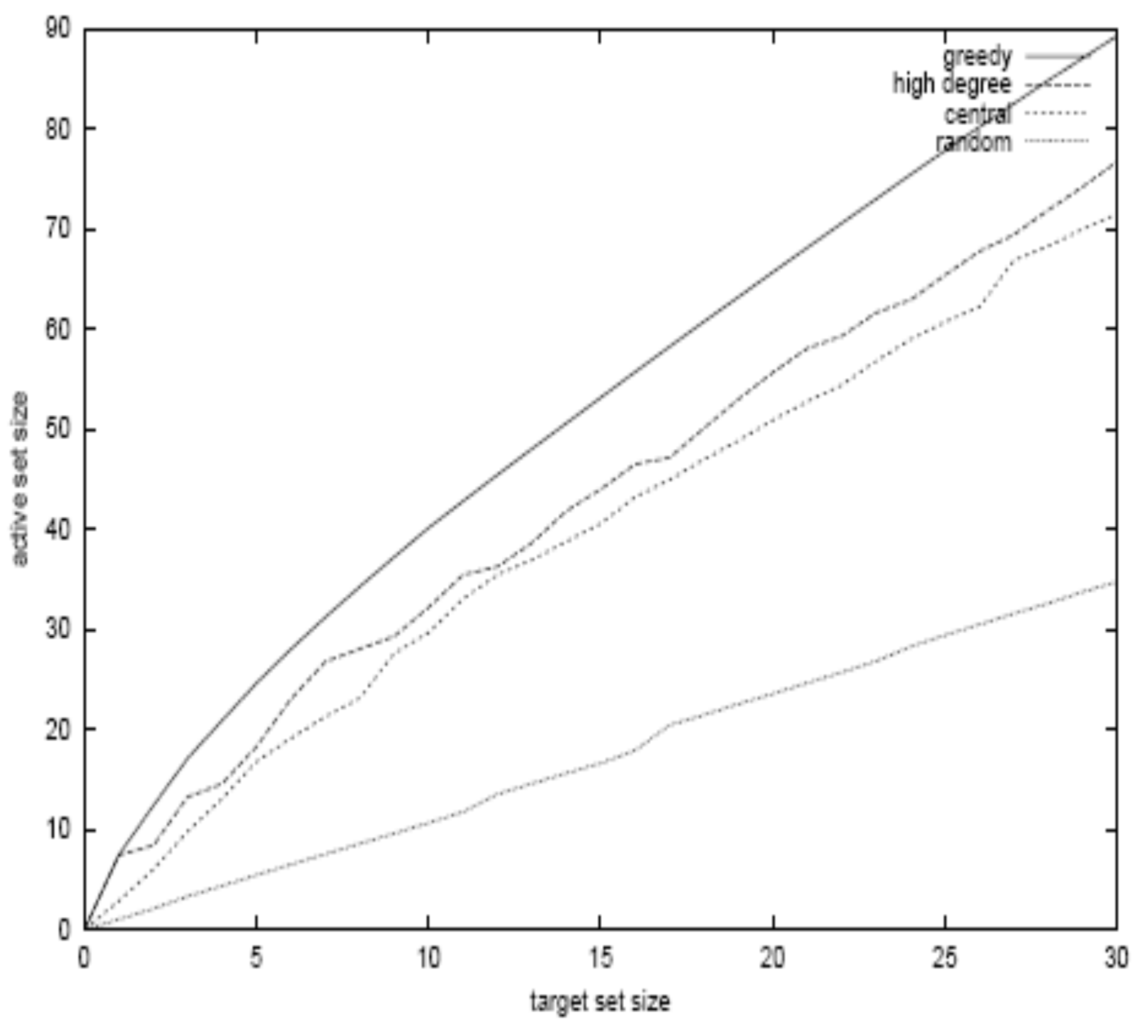
- Simulate process 1000 times, re-select probabilities and thresholds each time
- compare to 3 common heuristics



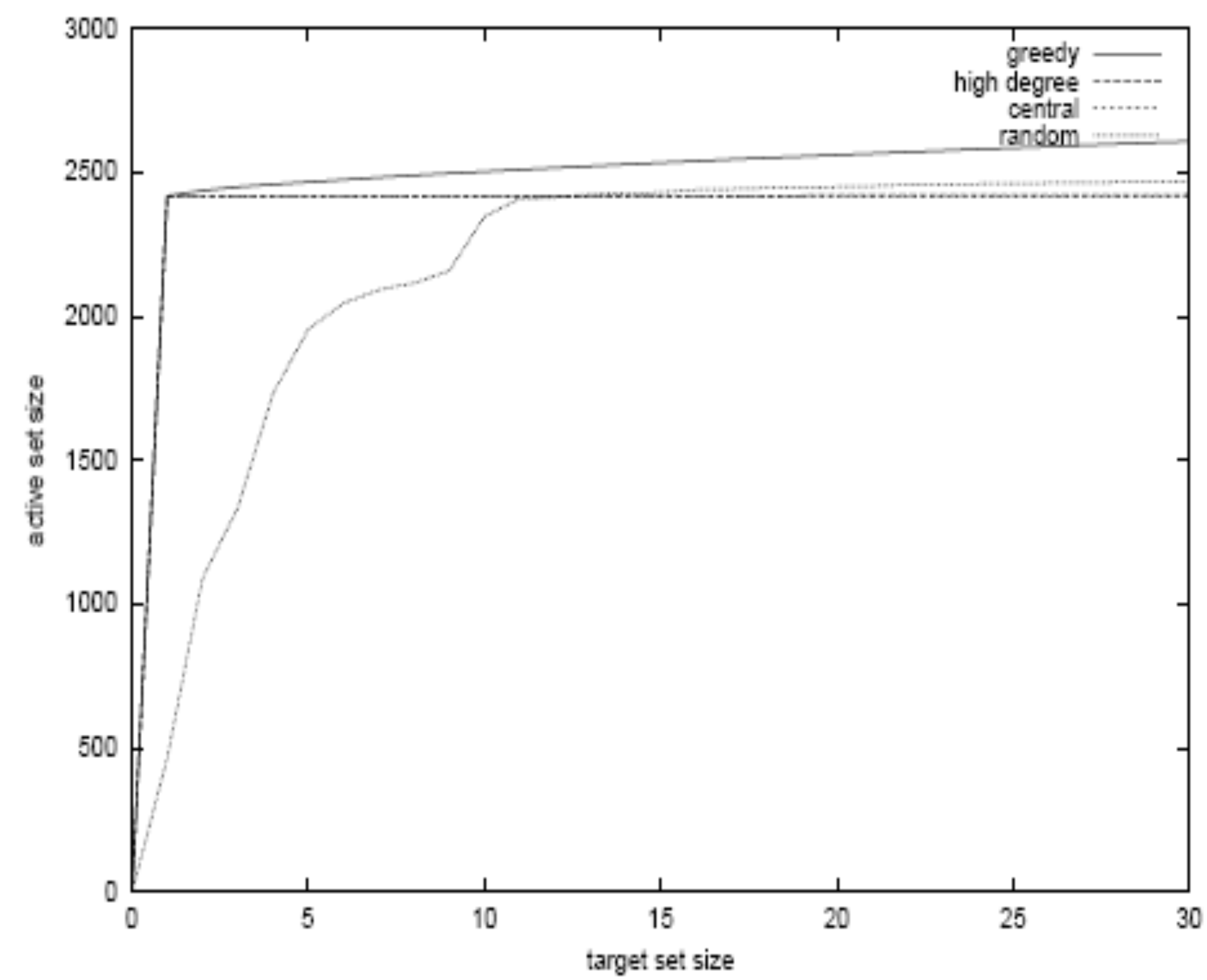
Results for linear threshold model

Experimental results - Kempe, Kleinberg, Tardos [2003]

- Simulate process 1000 times, re-select probabilities and thresholds each time
- compare to 3 common heuristics



p = 0.01



p = 0.1

Results for independent cascade model