How to Network in Online Social Network

Giovanni Neglia, Xiuhui Ye (Politecnico di Torino), Maksym Gabielkov, Arnaud Legout (Inria)

Maestro Team

27 March 2015

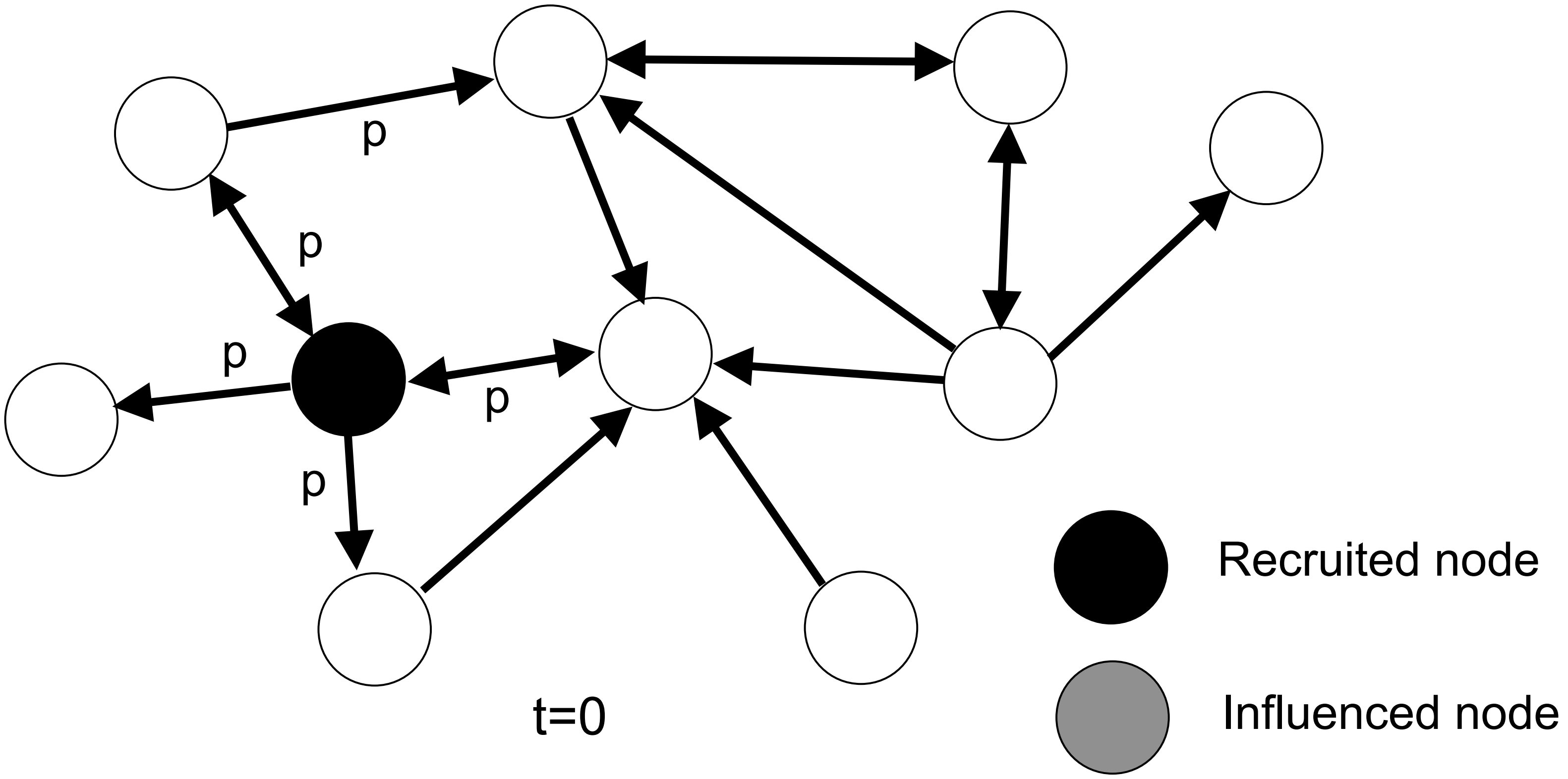
Outline

1. Influence maximization problem (Kempe, Kleinberg and Tardös in 2003)

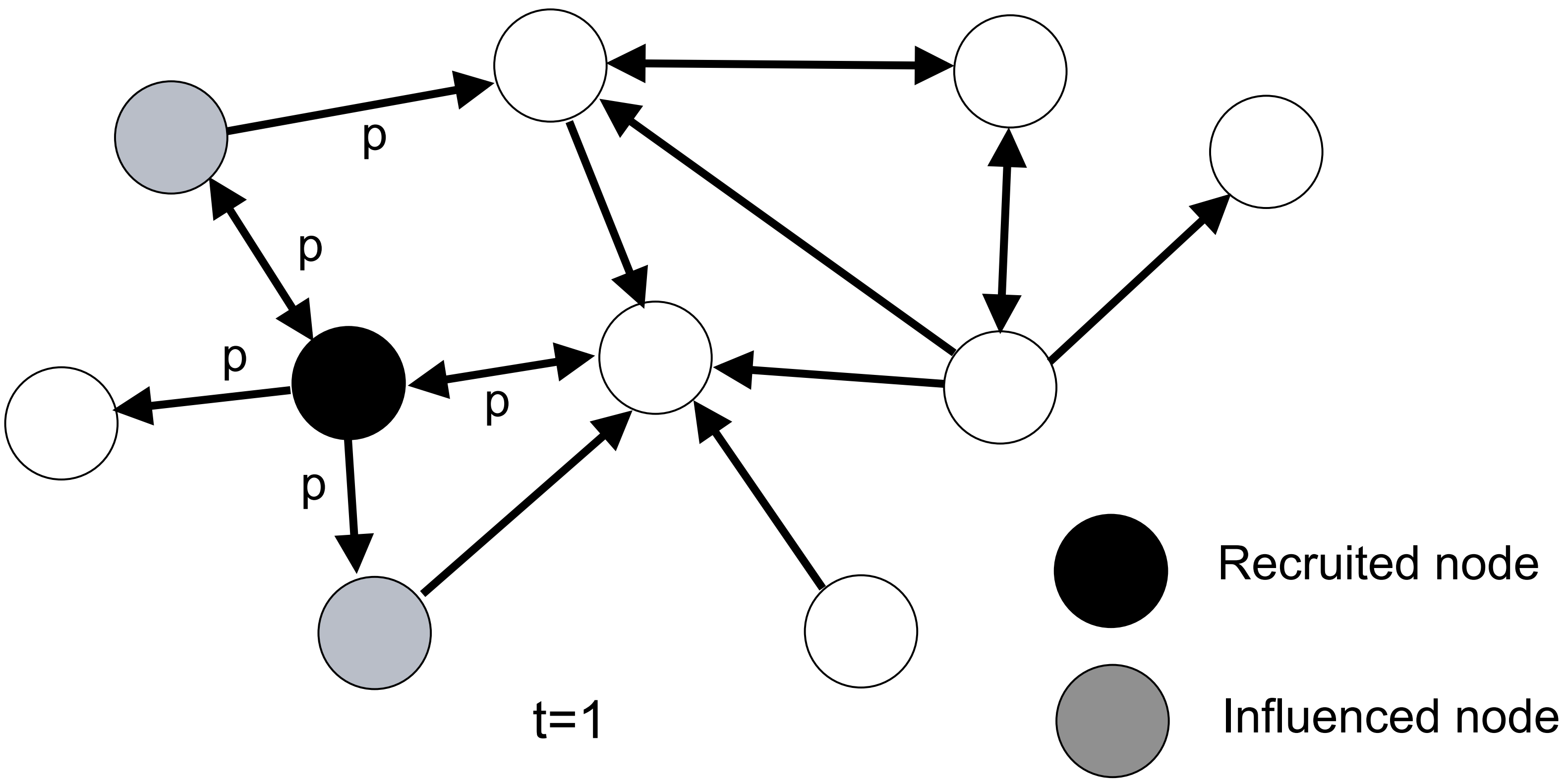
2. How the problem changes for a user in an online social network

3. Simulation results on Twitter's complete graph (2012)

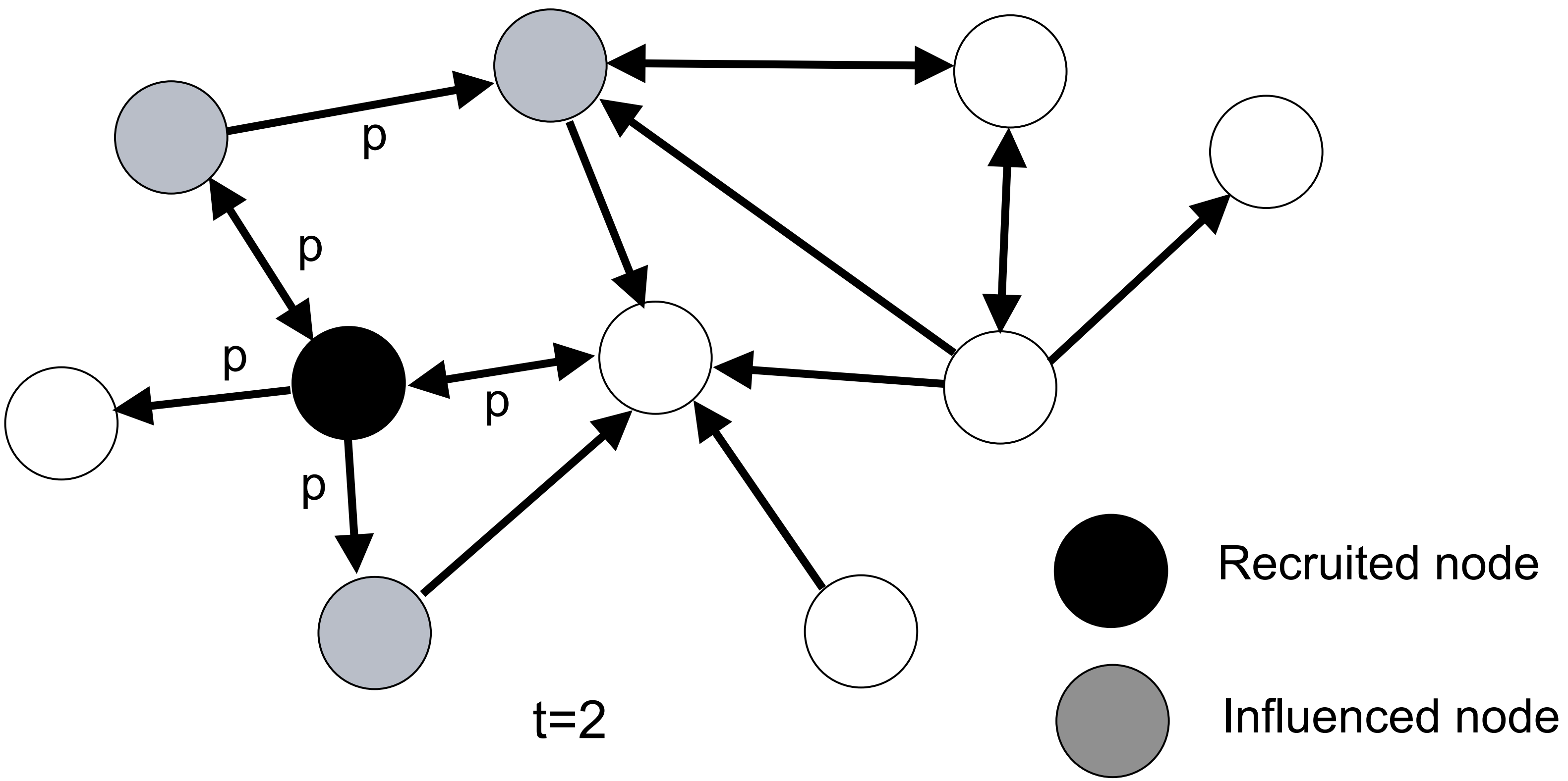
Influence propagation



Influence propagation

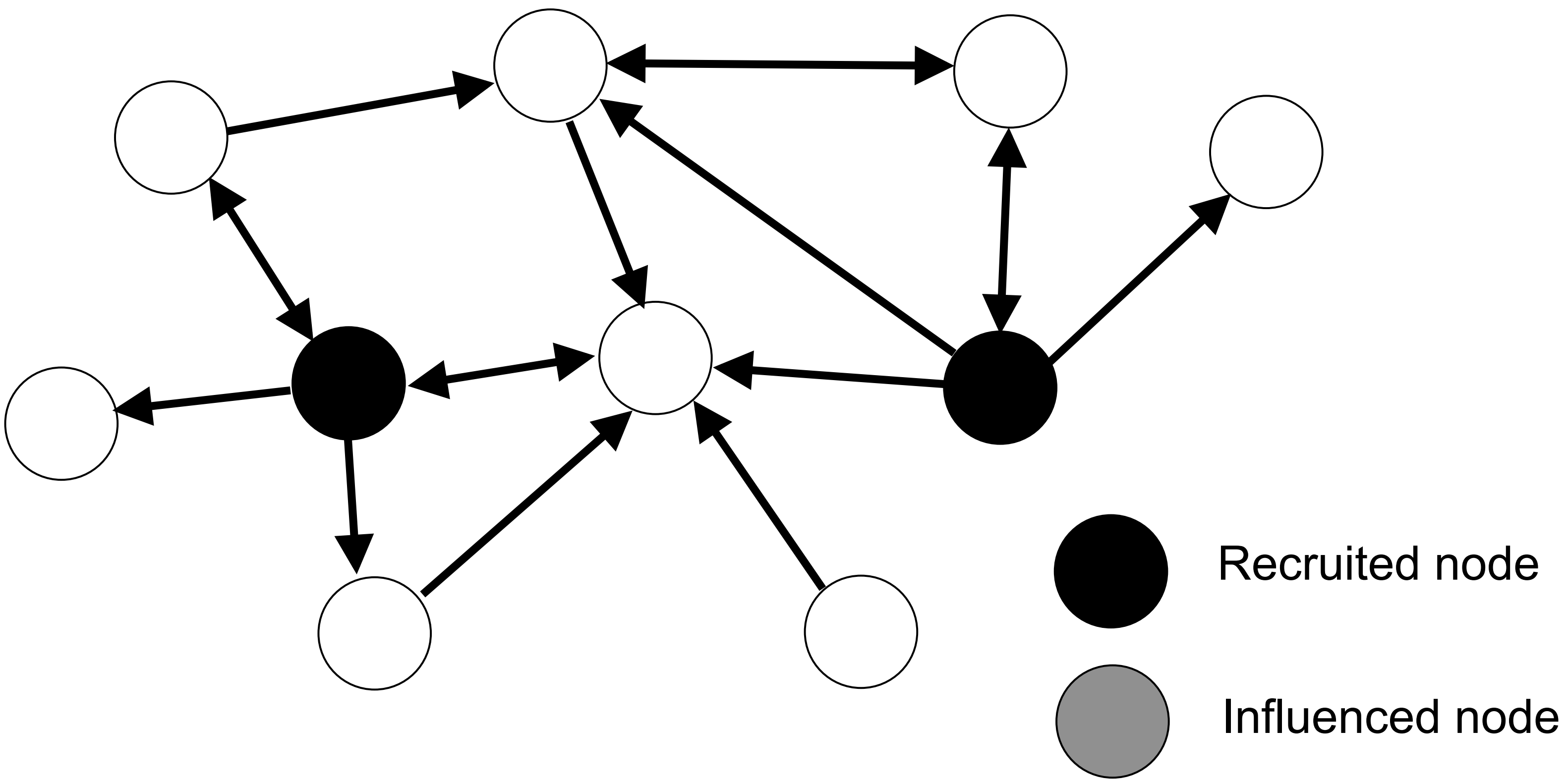


Influence propagation



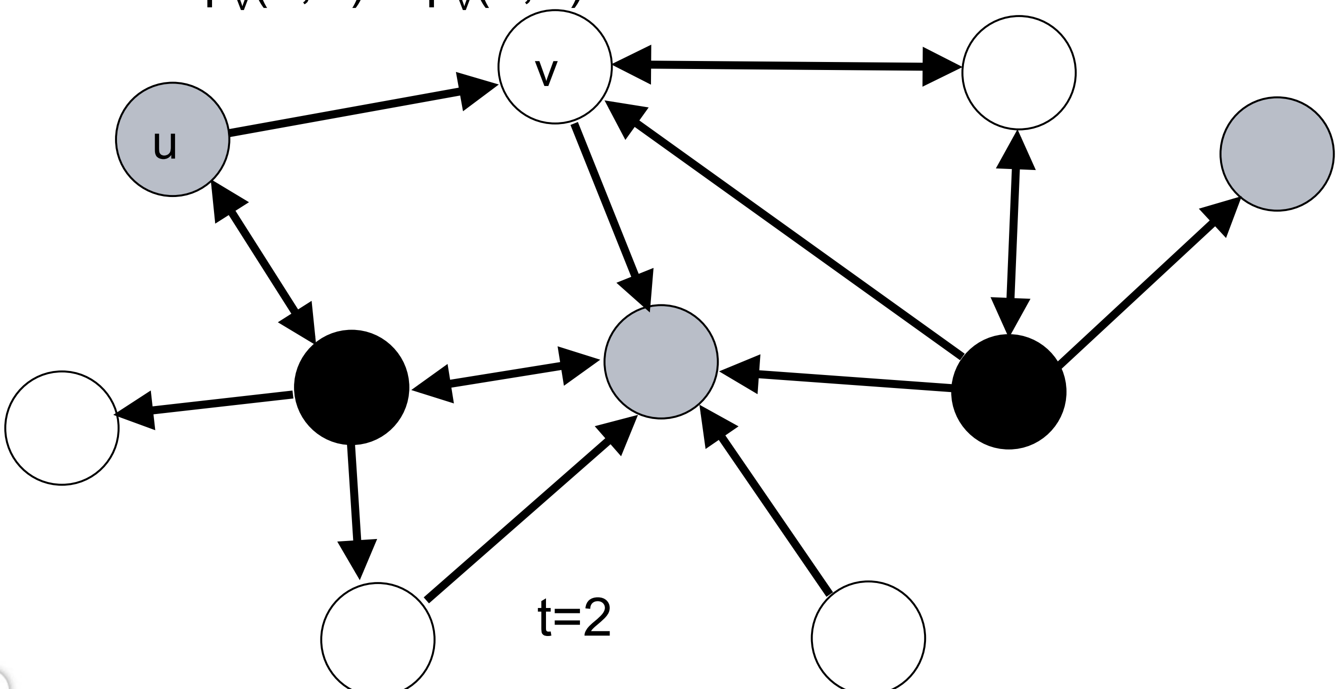
Influence maximization

Recruit a set A of K nodes to maximize the expected number of influenced nodes ($\sigma(A)=E[|\varphi(A)|]$)



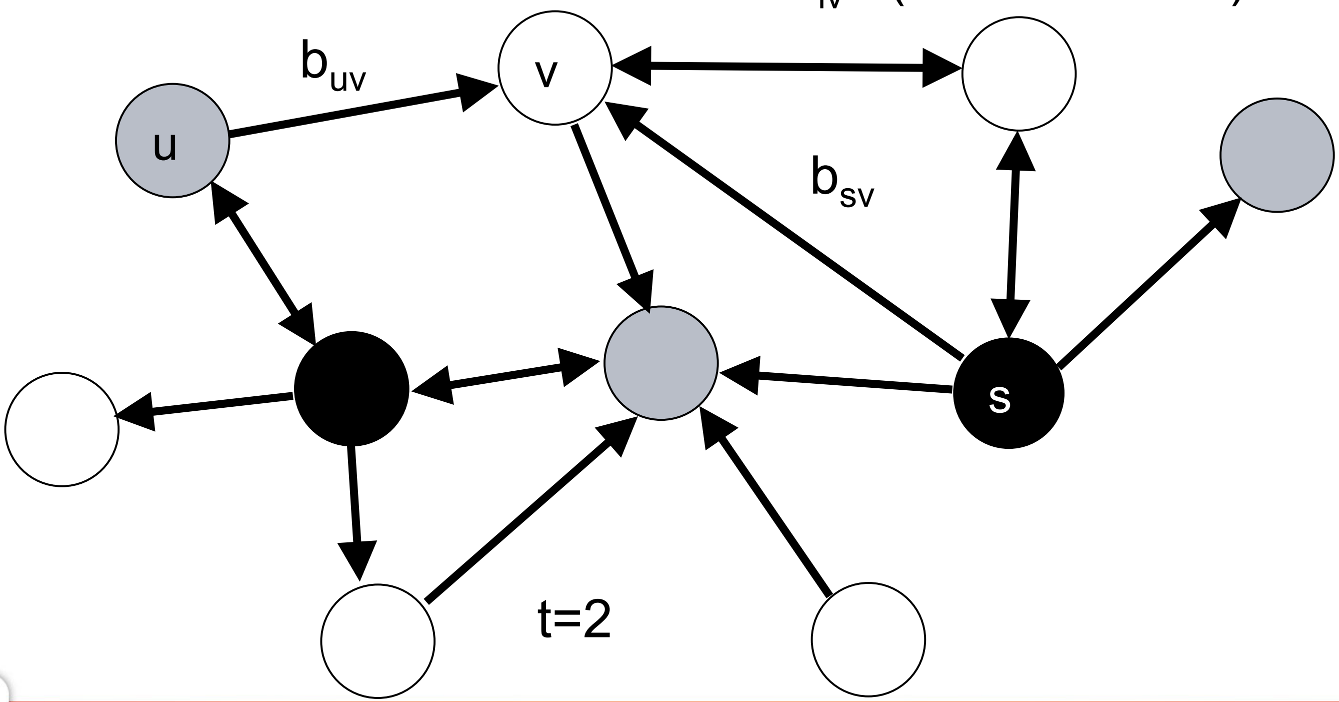
Kempe et al 2003

1. Decreasing cascade model:
   - $p_v(u,S)$ = prob. that u can influence v, given that nodes in S have already tried to influence v
   - $p_v(u,S) \geq p_v(u,T)$ if $S \subset T$



Kempe et al 2003

2. Linear Threshold Model
   - Node v has a threshold θ_v sampled from a uniform random variable in [0,1] and link (i,j) has a weight b_{ij}
   - Node v is influenced if $\Sigma\, b_{iv}\, \mathbf{1}(\text{i is influenced}) > \theta_v$



Kempe et al 2003

2. General Threshold Model
   - Node v has a threshold θ_v sampled from a uniform random variable in [0,1]
   - Node v has a monotone activation function f_v:2^V->[0,1] and is influenced at t if $f_v(S) > \theta_v$, where S is the set of influenced nodes at t

Kempe et al 2003

Their results:

I. Decreasing cascade model & General threshold model are equivalent
   - For each $\{p_v(u,S)\}$, it is possible to find $\{f_v(S)\}$ such that the probability distribution of $\varphi(A)$ is the same

Kempe et al 2003

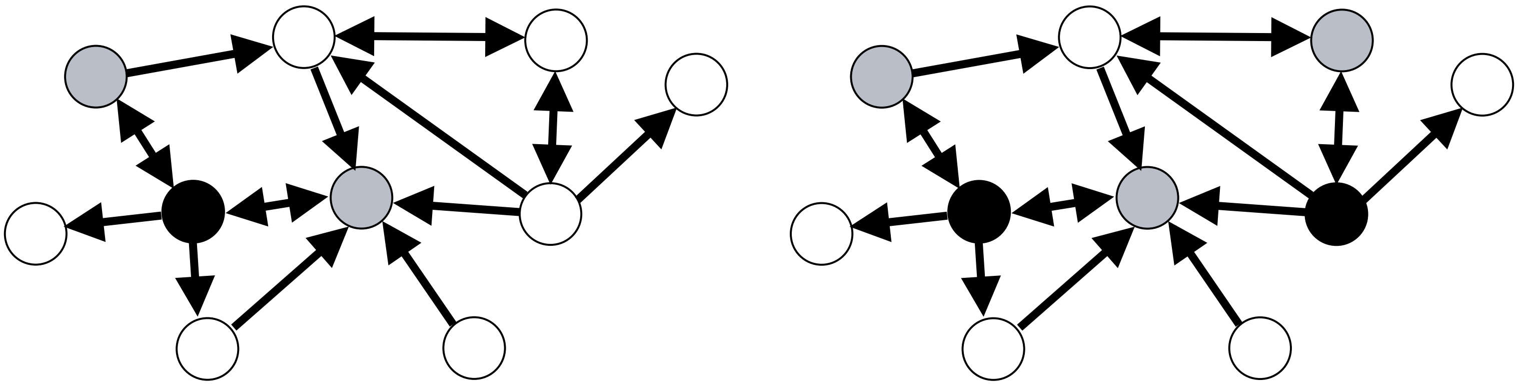
Their results:

I. Decreasing cascade model & General threshold model are equivalent
   - For each $\{p_v(u,S)\}$, it is possible to find $\{f_v(S)\}$ such that the probability distribution of $\varphi(A)$ is the same

II. The greedy algorithm achieves a (1-1/e) approximation ratio
   - This follows from a general result proven by Nemhauser, Wolsey, Fisher in '78 for non-negative, monotone, submodular functions

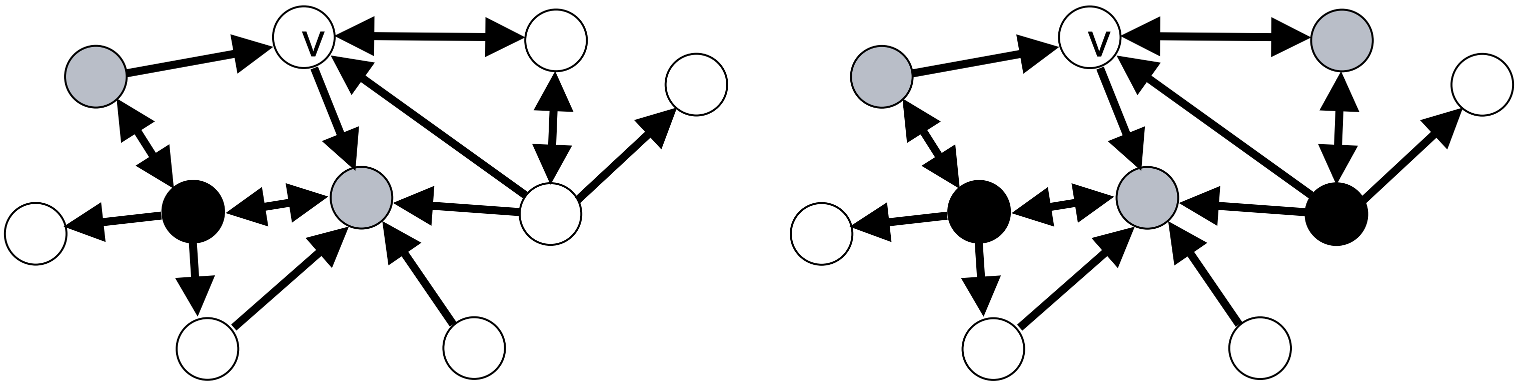
Monotonicity of σ(A)

- $\sigma(A_1) \le \sigma(A_2)$ if $A_1 \subset A_2$



Submodularity of σ(A)

- $\sigma(A_1 \cup \{v\}) - \sigma(A_1) \geq \sigma(A_2 \cup \{v\}) - \sigma(A_2)$ if $A_1 \subset A_2$



The greedy algorithm

1: start with $A=\{\}$

2: for $i = 1$ to K

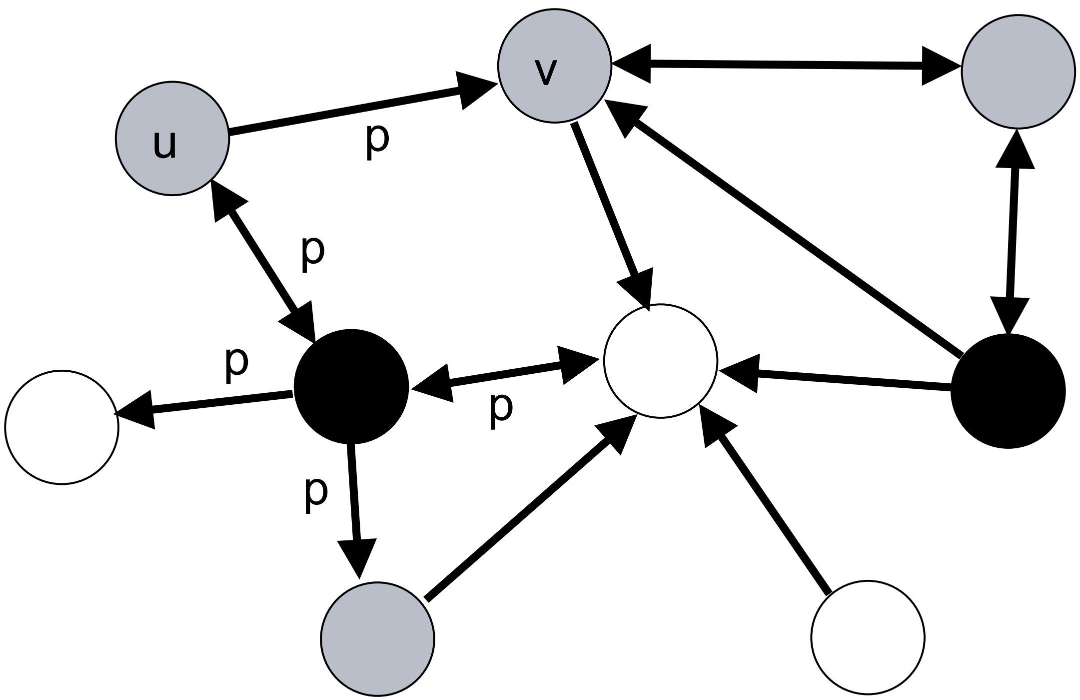
3: let v_i be the node maximizing the marginal gain

$$\sigma(A \cup \{v\}) - \sigma(A)$$

4: set $A := A \cup \{v_i\}$

Question: how to calculate $\sigma(A \cup \{v\}) - \sigma(A)$?

2. How the problem changes in OSN

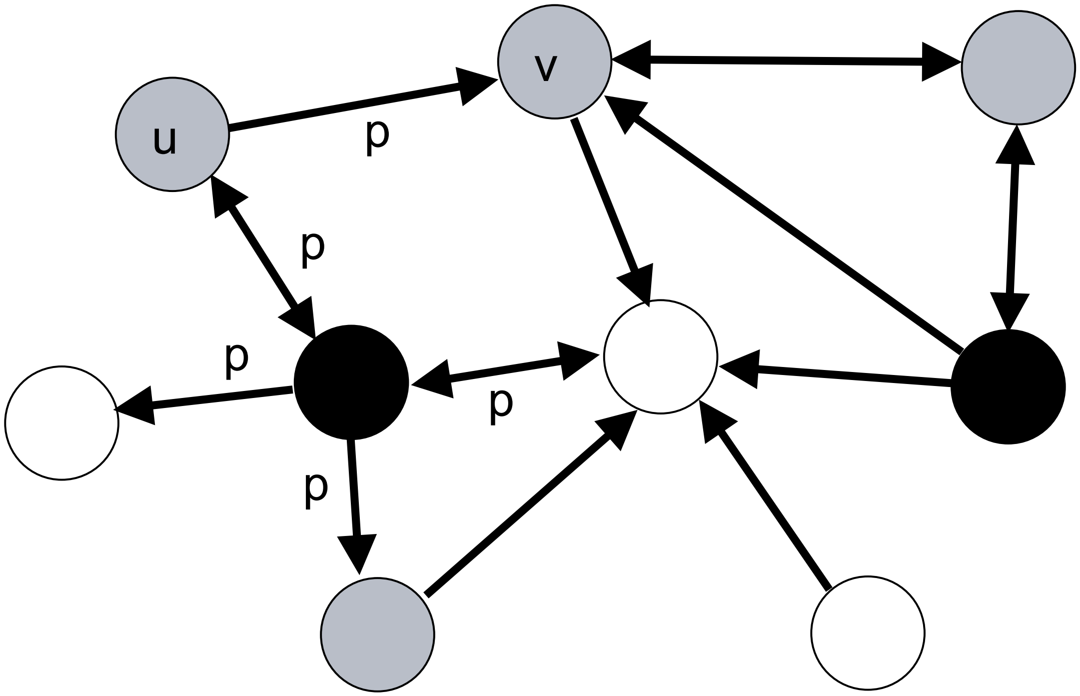


v follows u's tweet
v is a *follower* of u
u is a *following* of v

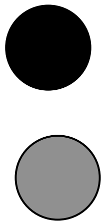
Tweeting node

Retweeting node

2. How the problem changes in OSN

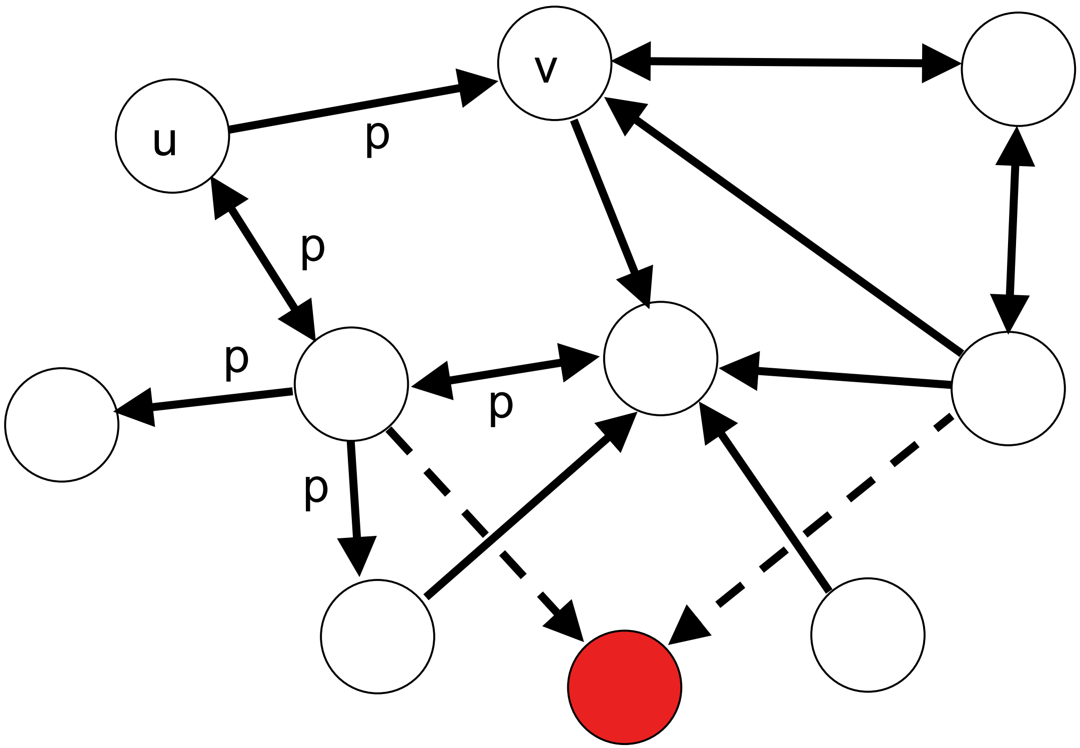


v follows u's tweet
v is a *follower* of u
u is a *following* of v



Assumption: a user can only influence people through Twitter itself

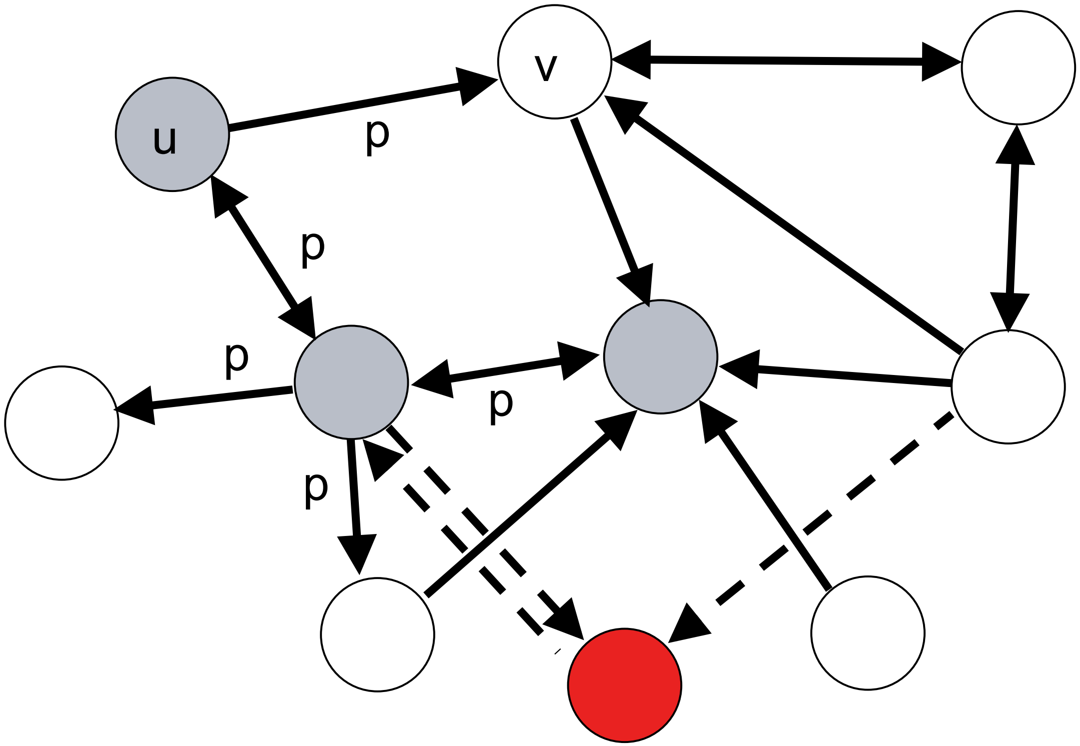
2. How the problem changes in OSN



v follows u's tweet
v is a *follower* of u
u is a *following* of v

The user can only select its followings (up to K=2000)…

2. How the problem changes in OSN



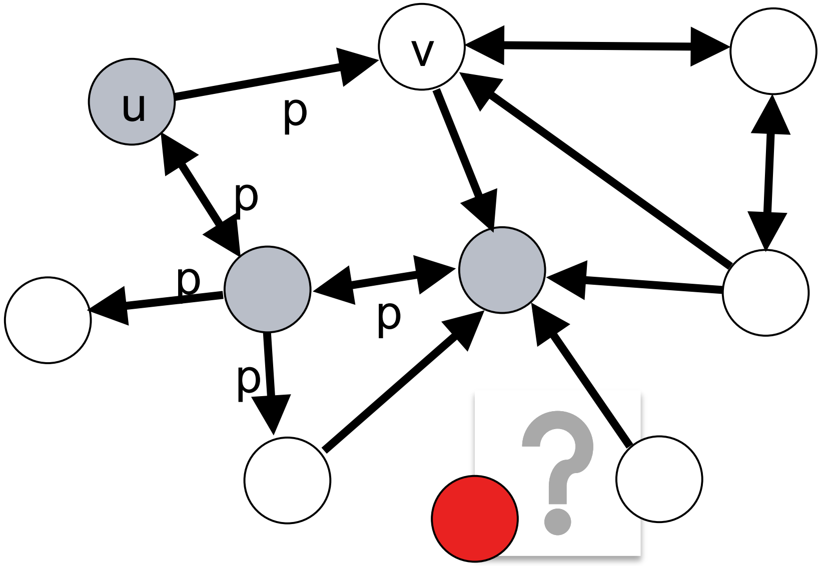
v follows u's tweet
v is a *follower* of u
u is a *following* of v

The user can only select its followings (up to K=2000)…
And hope that they follow back

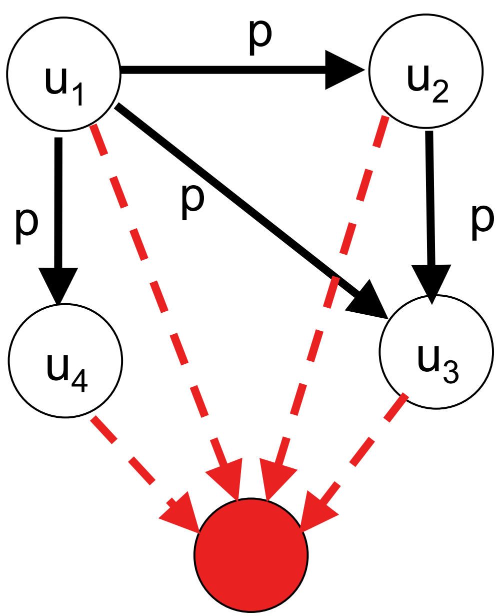
Our problem

Let the reciprocation probability r_v be known

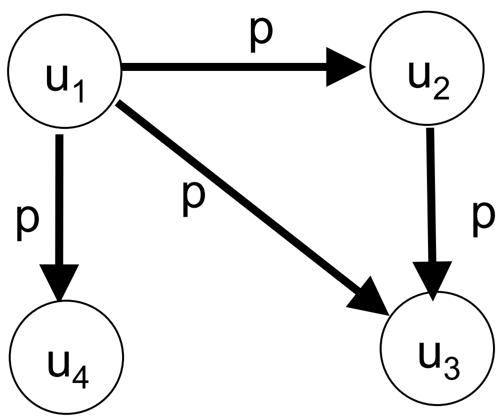
How should the user select the set of followings A in order to maximize $\sigma(A)=E[|\varphi(A)|]$? (all the choices at t=0)



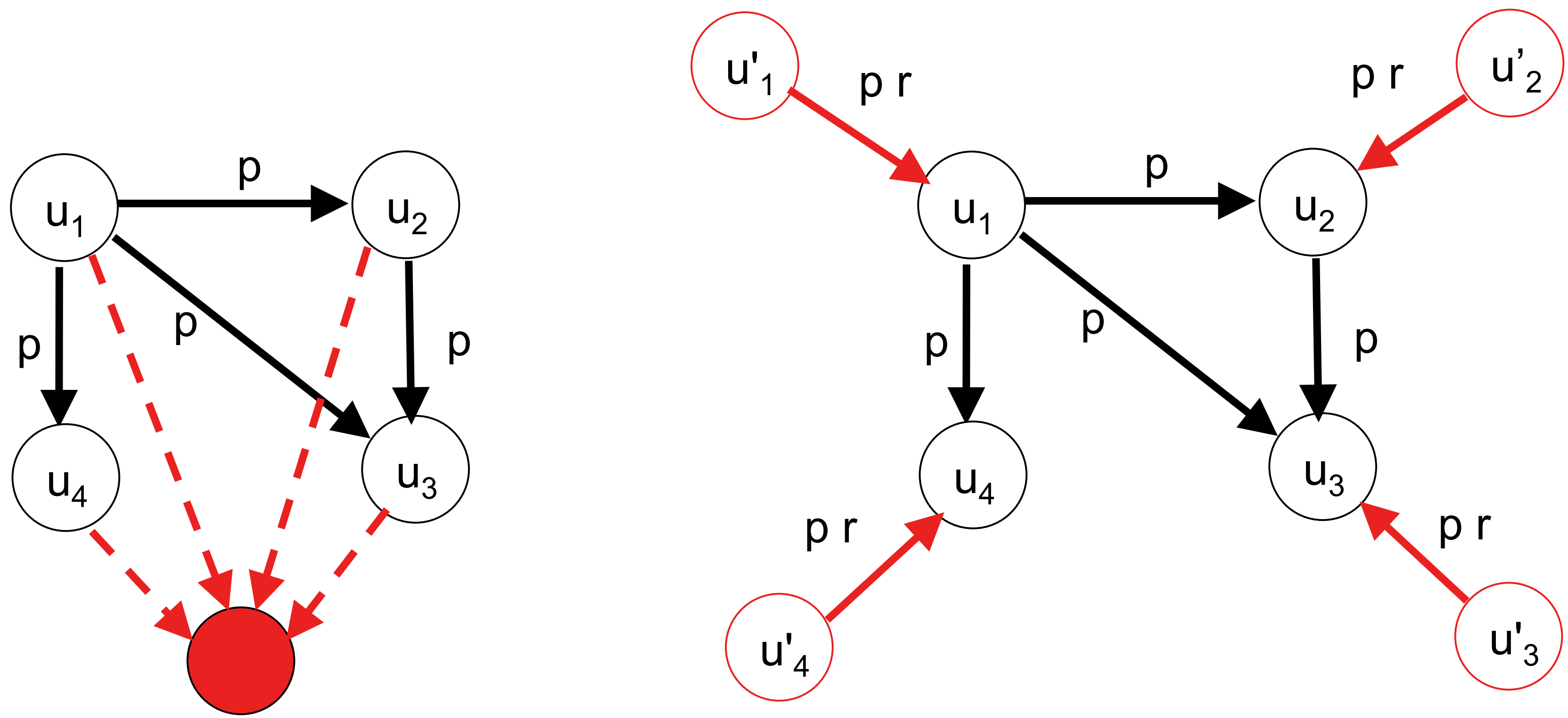
Map the new problem to the old one



Select K followers

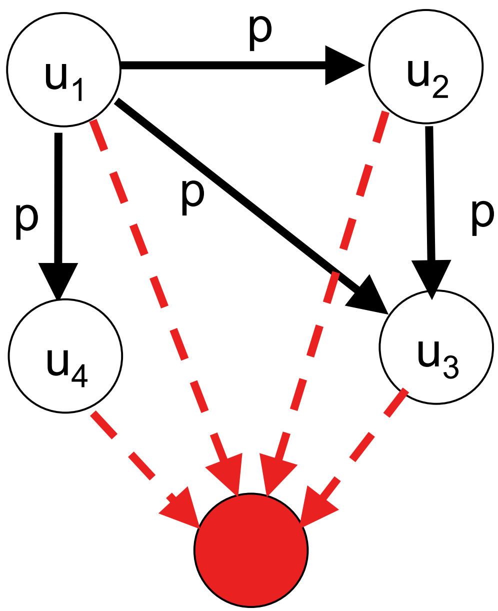


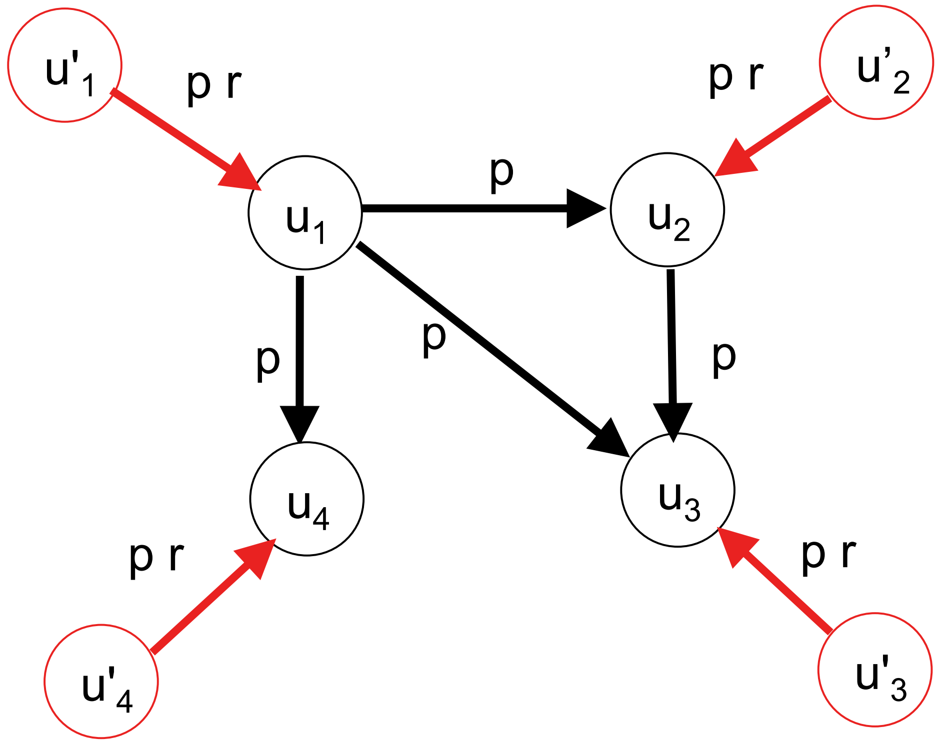
Map the new problem to the old one



Select K followings **equivalent to** Recruit K nodes in V'

Map the new problem to the old one





Select K followers **equivalent to** Recruit K nodes in V'

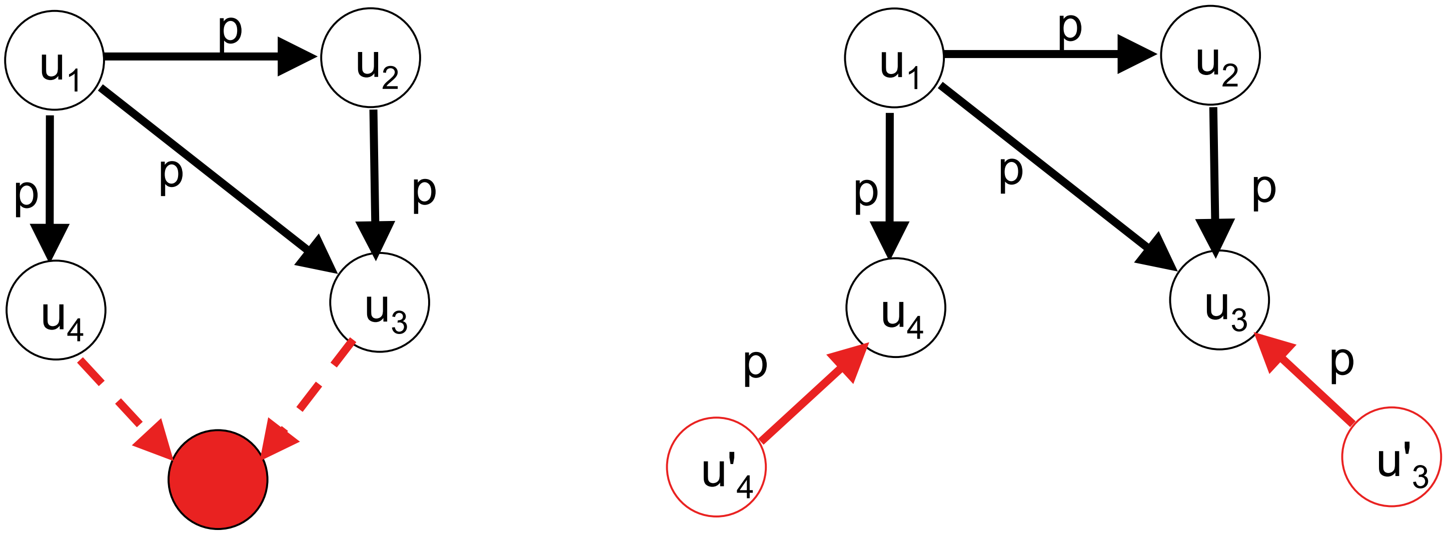
Greedy algorithm has the same approximation ratio

A 2nd twist: dynamic policies

- Following users is not expensive
- Idea: replace non-reciprocating users
- How to operate:
  - follow one user
  - if the user does not reciprocate by T
    - unfollow it and follow someone else
- It is now possible to follow over time more than K users, but only K at a given time instant

An ideal policy

- Imagine to know who is going to reciprocate by T



- The greedy algorithm with such knowledge would achieve an (1-1/e) approximation ratio

A practical greedy policy

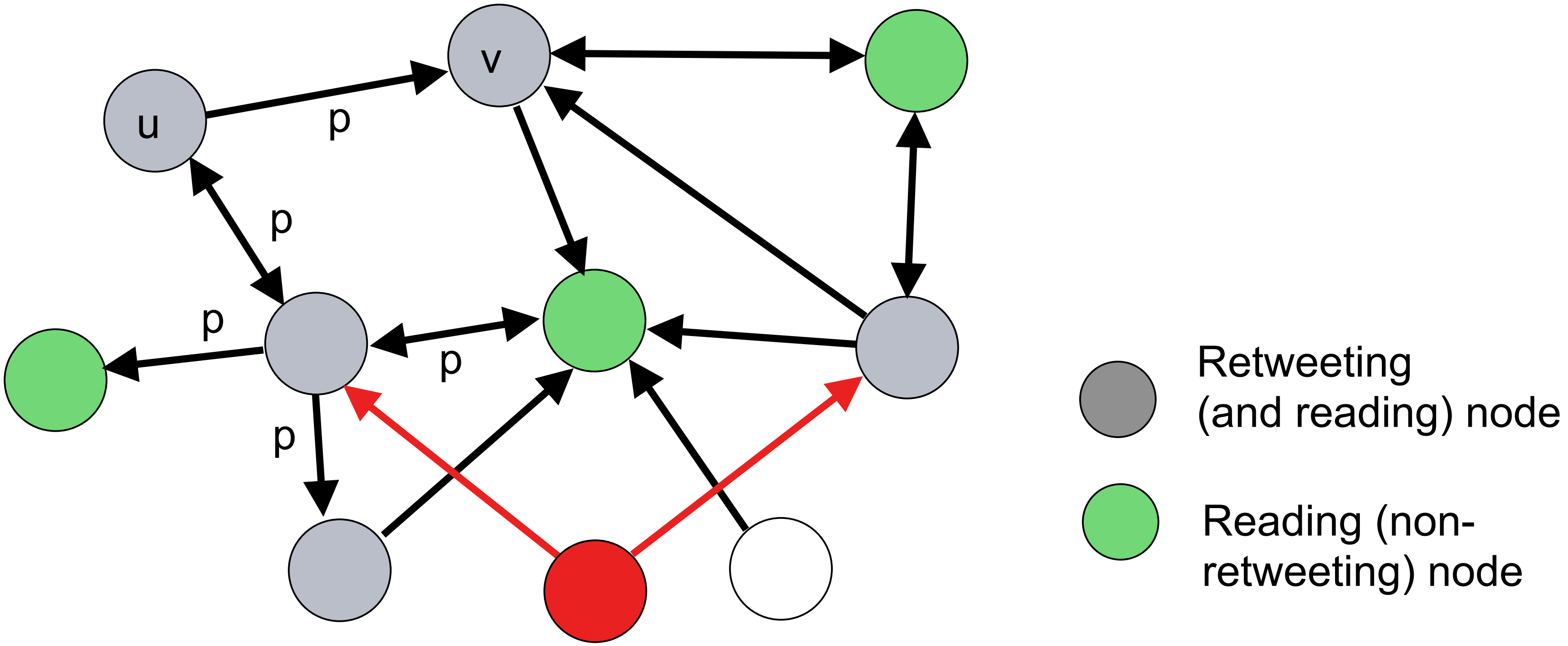
```
start with A={}, D={} i=0
while i ≤ K
    let v_i be the node in V-D maximizing the marginal
    gain σ(A U {v}) - σ(A), given that it reciprocates
    follow v_i
    if v_i reciprocates by T:
        A:=A U {v_i}, i=i+1
    else:
        D:=D U {v_i}
```

A practical greedy policy

```
start with A={}, D={} i=0
while i ≤ K
    let v_i be the node in V-D maximizing the marginal
    gain σ(A U {v}) - σ(A), given that it reciprocates
    follow v_i
    if v_i reciprocates by T:
        A:=A U {v_i}, i=i+1
    else:
        D:=D U {v_i}
```

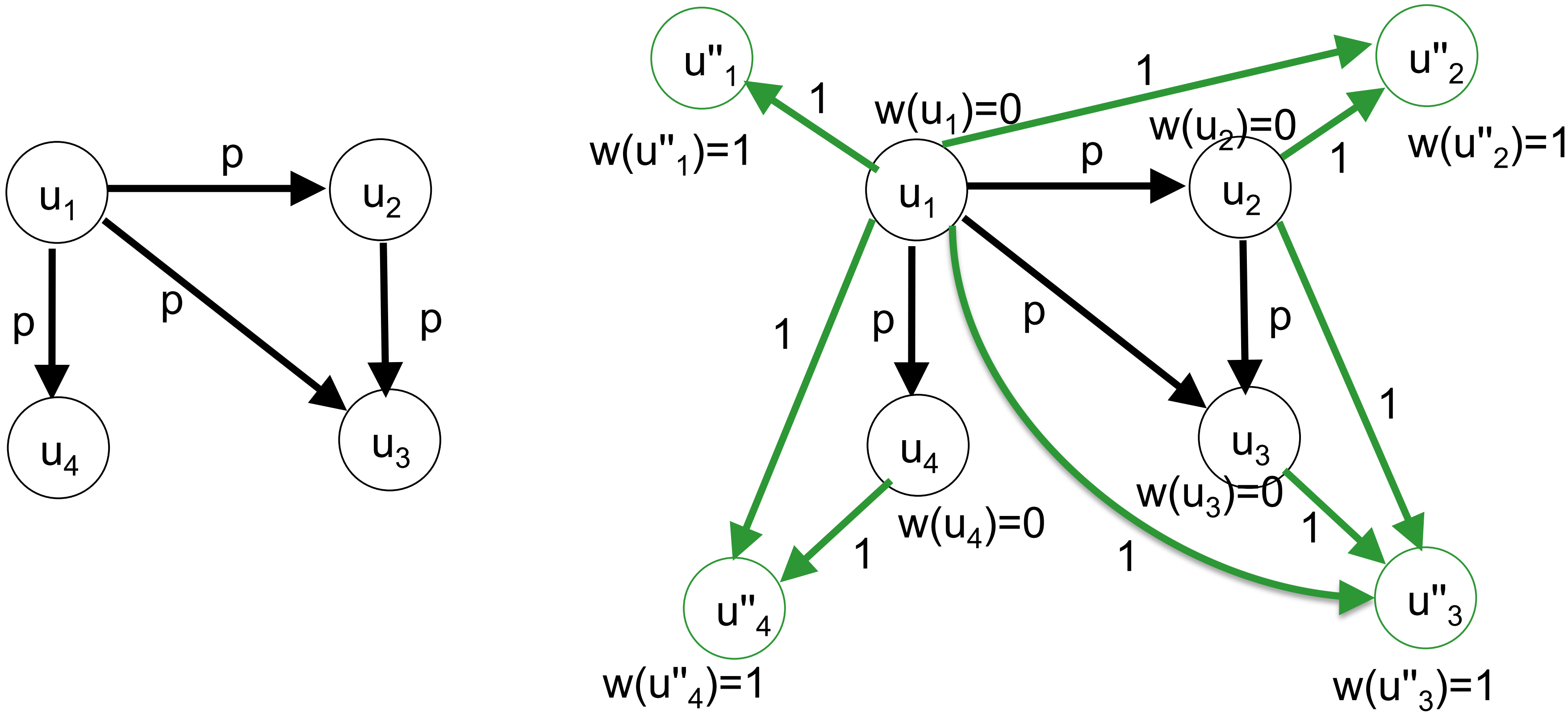
practical greedy = ideal greedy

#Readers vs #Retwitters (3rd twist)



What if we consider as performance metric #readers?

Map the new problem to the old one



Select K nodes to maximize $E[\Sigma\, w(u_i)\, \mathbf{1}(u_i \text{ is active})]$

An ideal policy

- Is $E[\Sigma\, w(u_i)\, \mathbf{1}(u_i \text{ is active})]$ submodular?
  - Yes it is (need to go carefully through the steps of Kempe et al)
- then greedy is a (1-1/e) approximation algorithm

Wrap up

- The point of view of a user in an OSN introduces new twists, but does not change fundamentally the problem
  - In particular the greedy algorithm guarantees a (1-1/e) approximation ratio

Wrap up

- The point of view of a user in an OSN introduces new twists, but does not change fundamentally the problem
  - In particular the greedy algorithm guarantees a (1-1/e) approximation ratio
- Limits:
  - need to know the whole topology, $p_v(u,S)$, r_v
  - How to calculate the marginal gain? Montecarlo simulations…

Outline

1. Influence maximization problem (Kempe, Kleinberg and Tardös in 2003)

2. How the problem changes for a user in an online social network

3. Simulation results on Twitter's complete graph (2012)

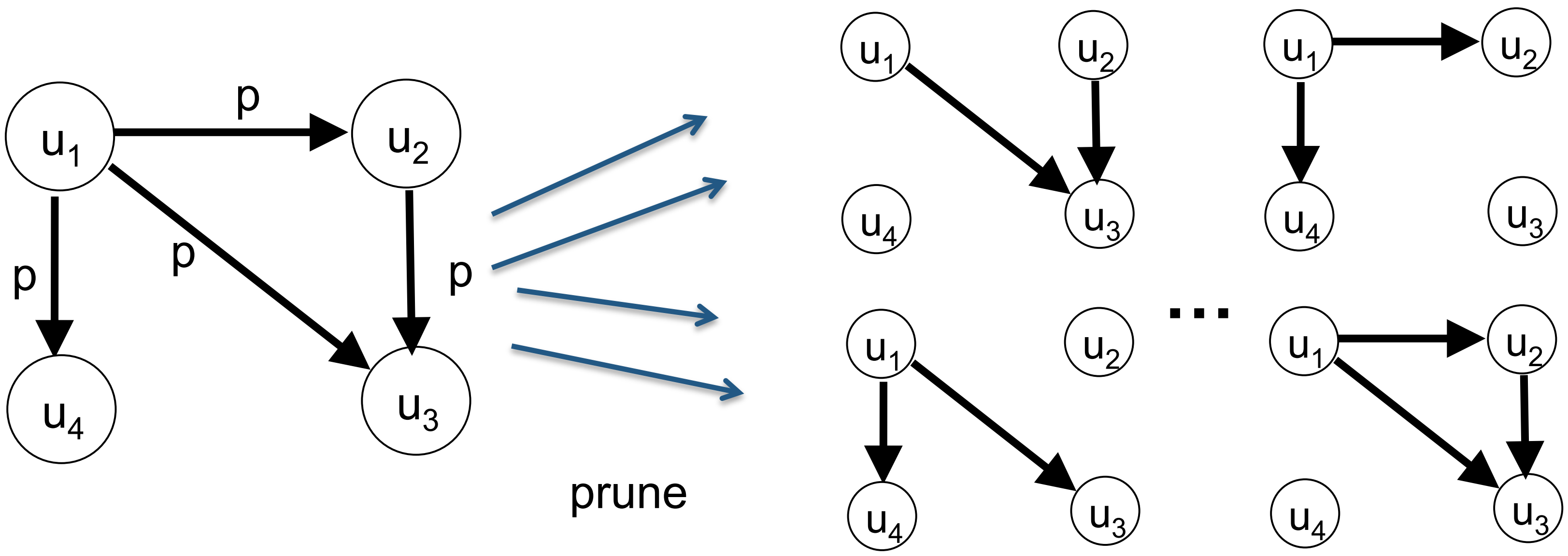
Know your enemy

- Crawl of the whole Twitter in June 2012
- 500 million of nodes
- 23 billion of arcs
- 417GB as an edgelist

Montecarlo simulations

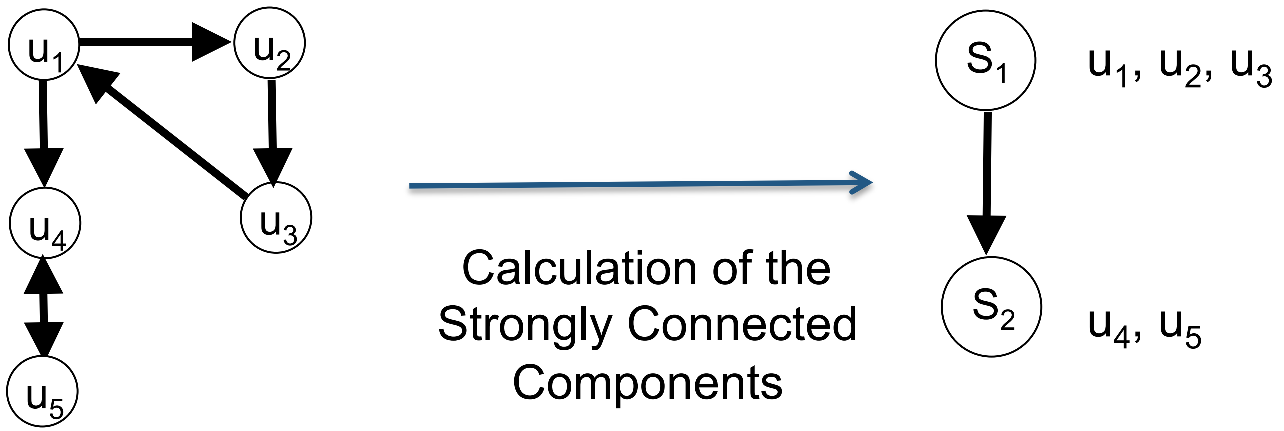
- Naive implementation
  - O(NKS) simulations,
    - where S is #simulations to achieve the required confidence
  - ≈100GB to store the graph in RAM

Trade RAM for Storage



- Influenced node of a cascade = reachable nodes in the pruned graph
- Need to store S * p * 417GB
- RAM still a problem for p≥1%

Useful preprocessing



Pruned graph

SCCs' graph

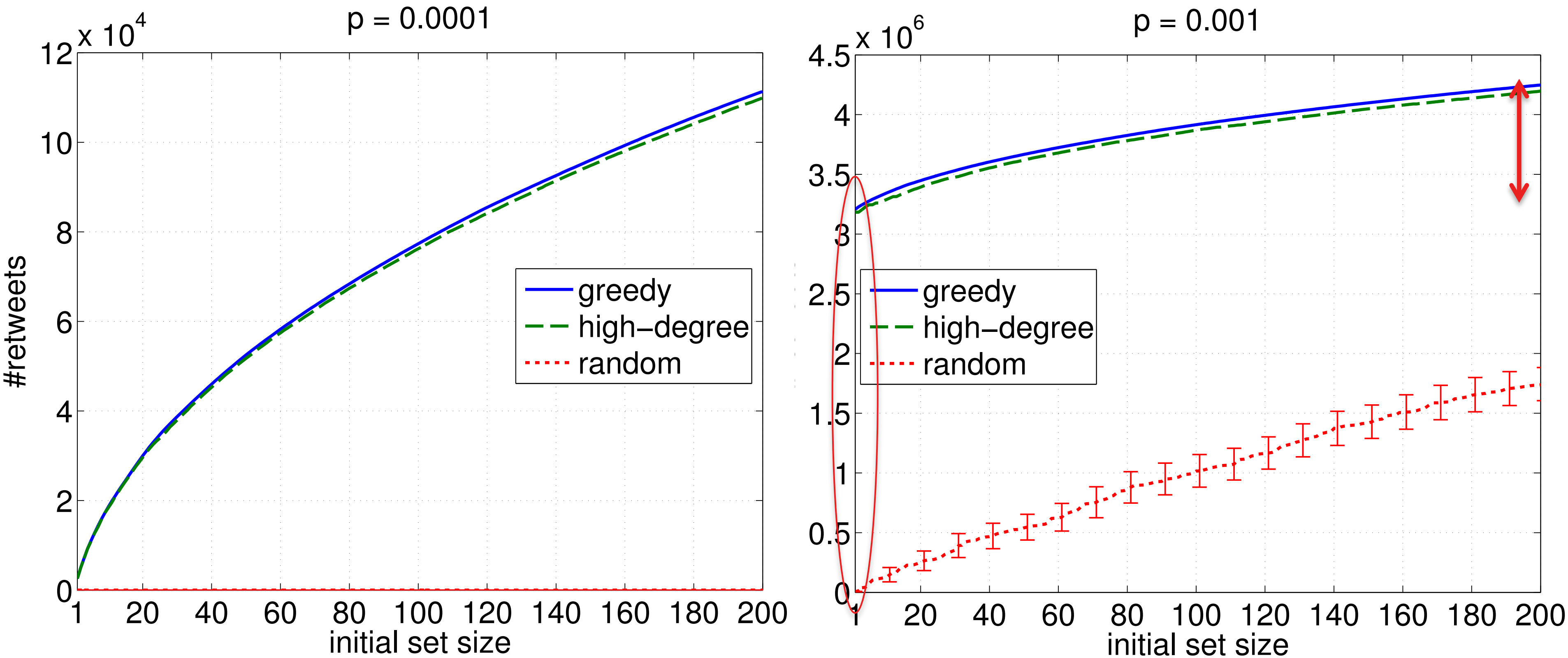
- Reachability can also be calculated on the SCCs' graph
- For larger p we save memory, storage and computation

How many samples?

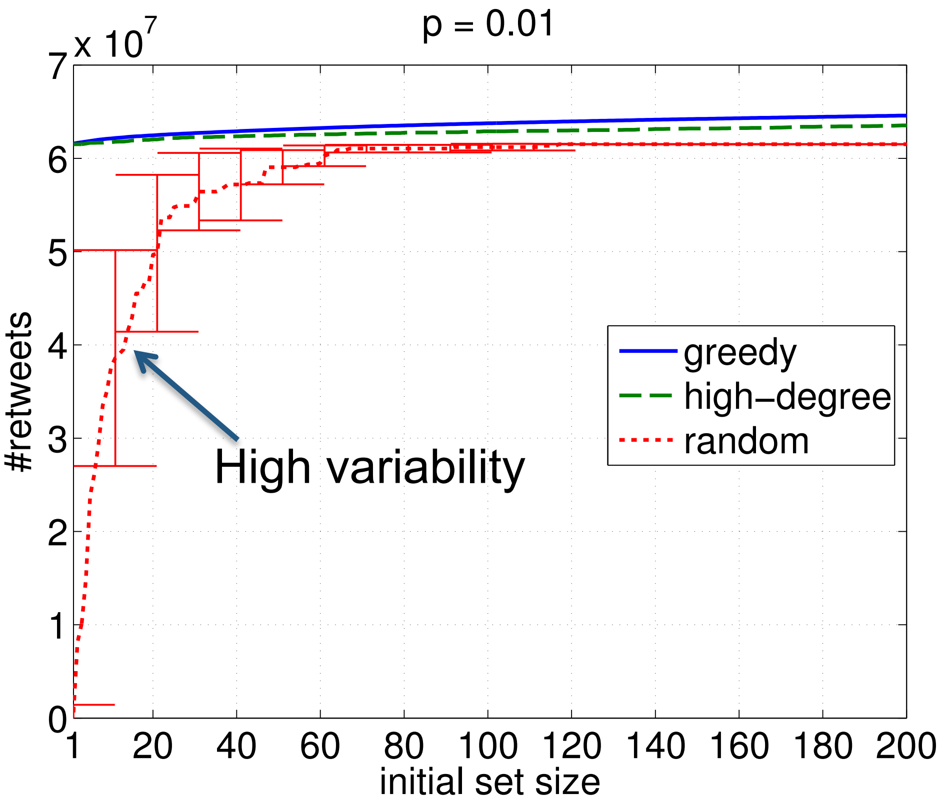
- We tried to estimate it analytically
  - Random configuration model
  - Subcritical branching process for small p
  - All-or-nothing supercritical branching process for large p
- S≤100 for all the values of p

Different algorithms

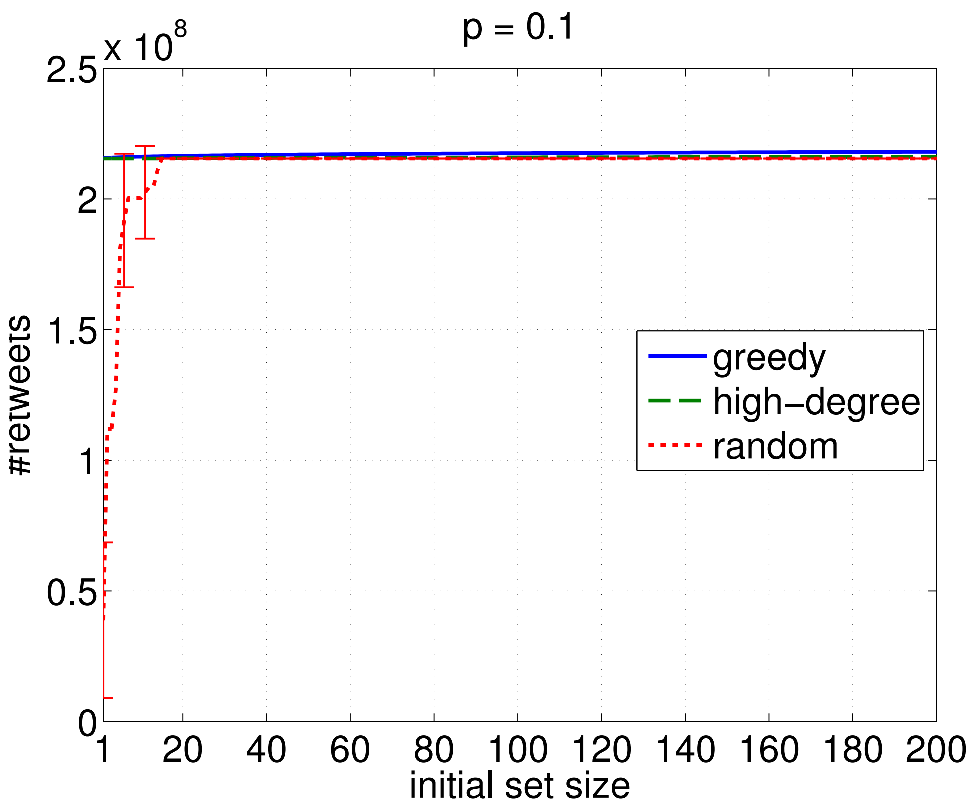
1. Greedy
   - Know topology, probabilities
2. Highest degree
   - Know nodes' degrees
3. Random
   - Know nodes' ids



$\langle d \rangle p \approx 4*10^{-3}$ $\langle d \rangle p \approx 4*10^{-2}$

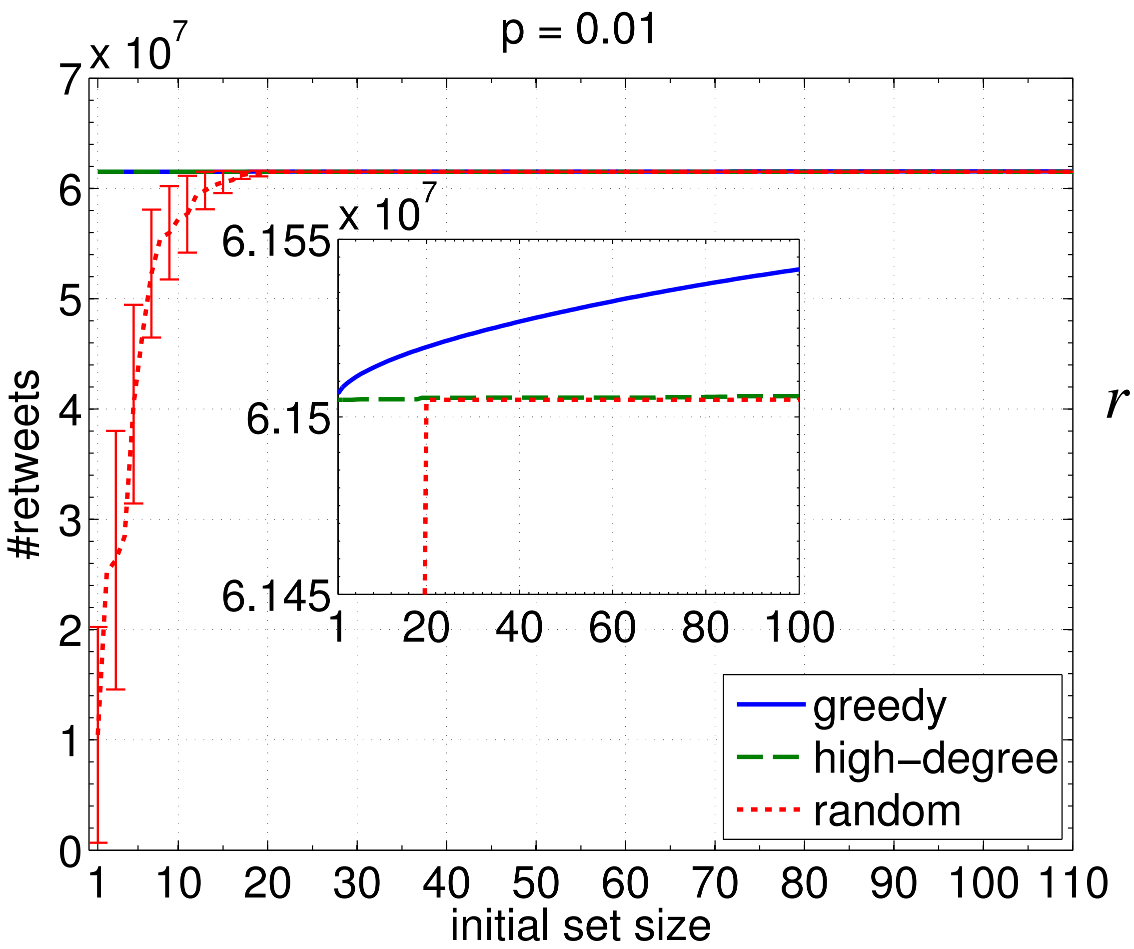

p = 0.01
x 10^7
7
6
5
4
3
2
1
0
#retweets
1
20
40
60
80
100
120
140
160
180
200
initial set size
greedy
high–degree
random
High variability

<d>p≈0.4


p = 0.1
x 10^8
2.5
2
1.5
1
0.5
0
#retweets
1
20
40
60
80
100
120
140
160
180
200
initial set size
greedy
high–degree
random

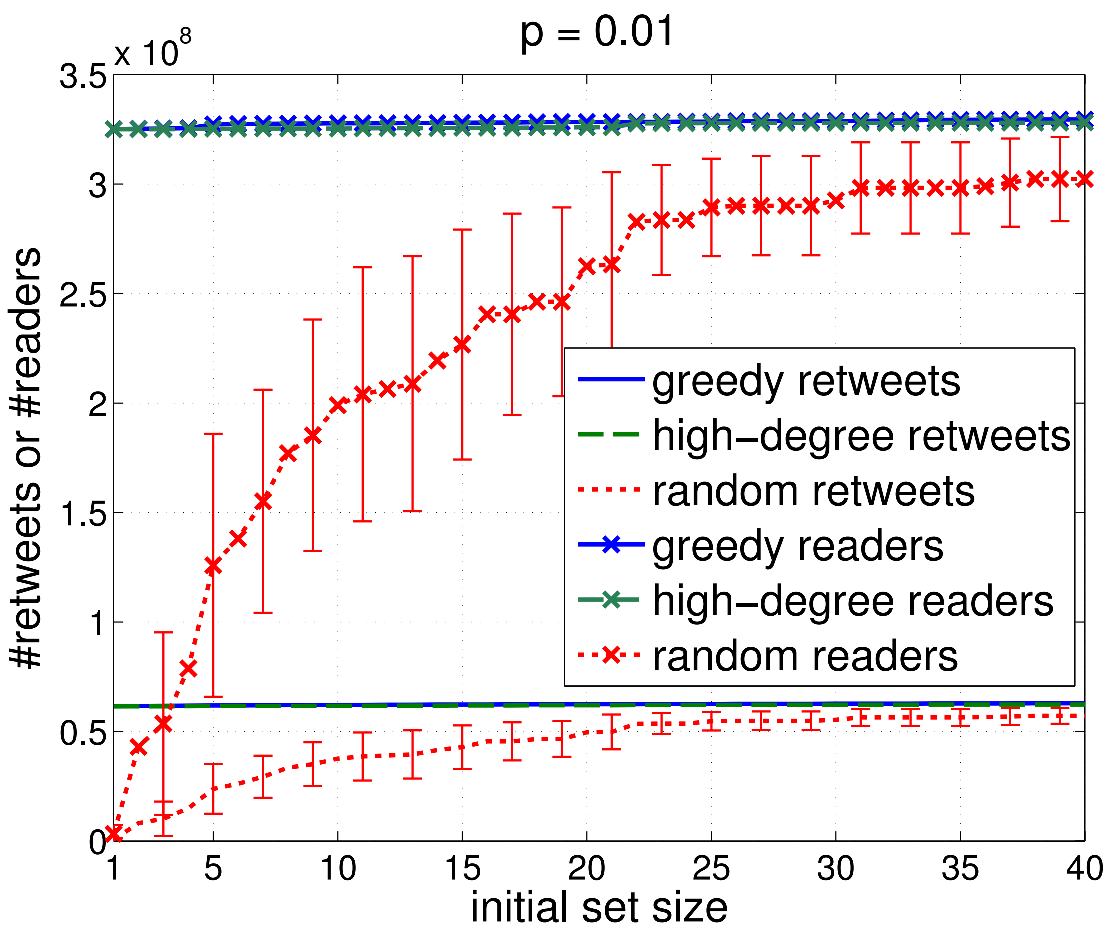
<d>p≈4

The effect of reciprocity



$$r = \min\left(\frac{\#\ followings}{\#\ followers + 100}, 1\right)$$

#Readers vs #Retwitters



Take Home Lesson

- For sparse graphs, highest degree (1-hop ahead) works as well as greedy
- For dense graphs, any strategy, even random, works as well as greedy
- Only in the middle, greedy can outperform highest degree…
  - Remarks in Habiba and Berger-Wolf, 2011
- … but we do not observe it

Thank you!

Questions?

INVENTORS FOR THE DIGITAL WORLD