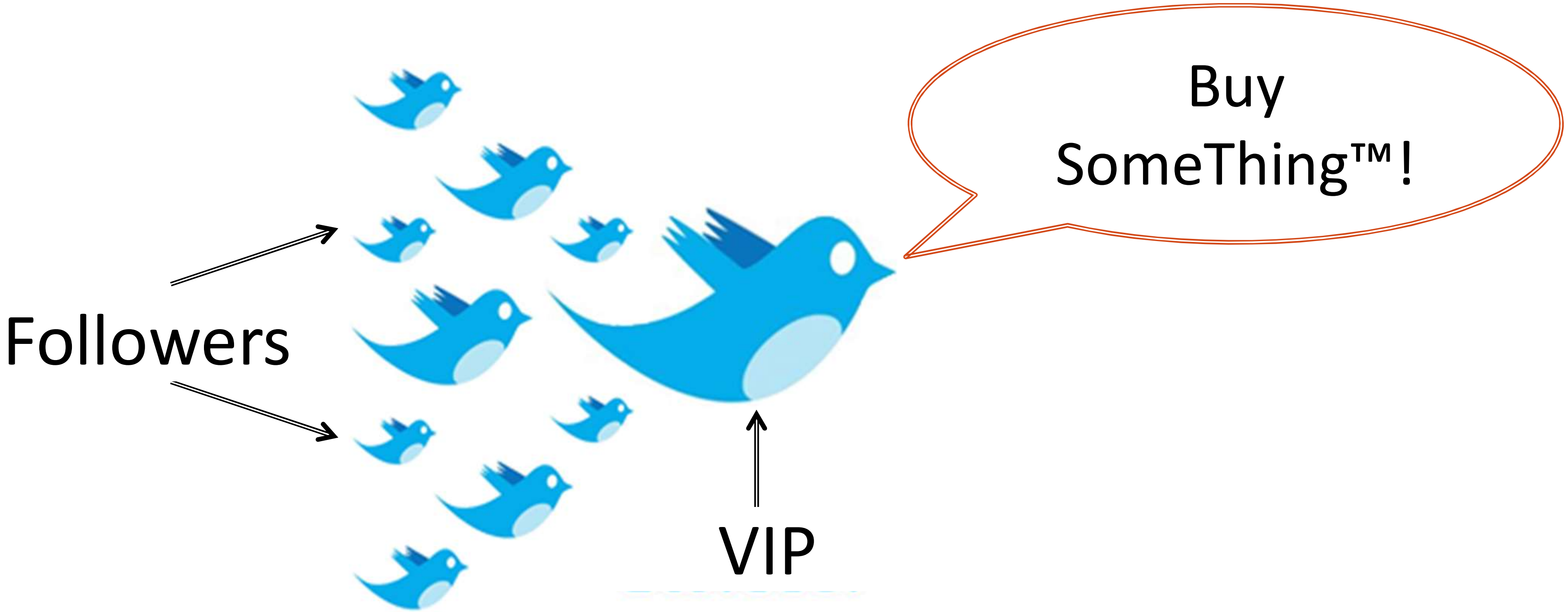
Minimum Target Set

Gennaro Cordasco, Luisa Gargano, Marco Mecchia, Adele A. Rescigno, Ugo Vaccaro
Discovering Small Target Sets in Social Networks: A Fast and Effective Algorithm.
Algorithmica 80(6): 1804-1833 (2018)

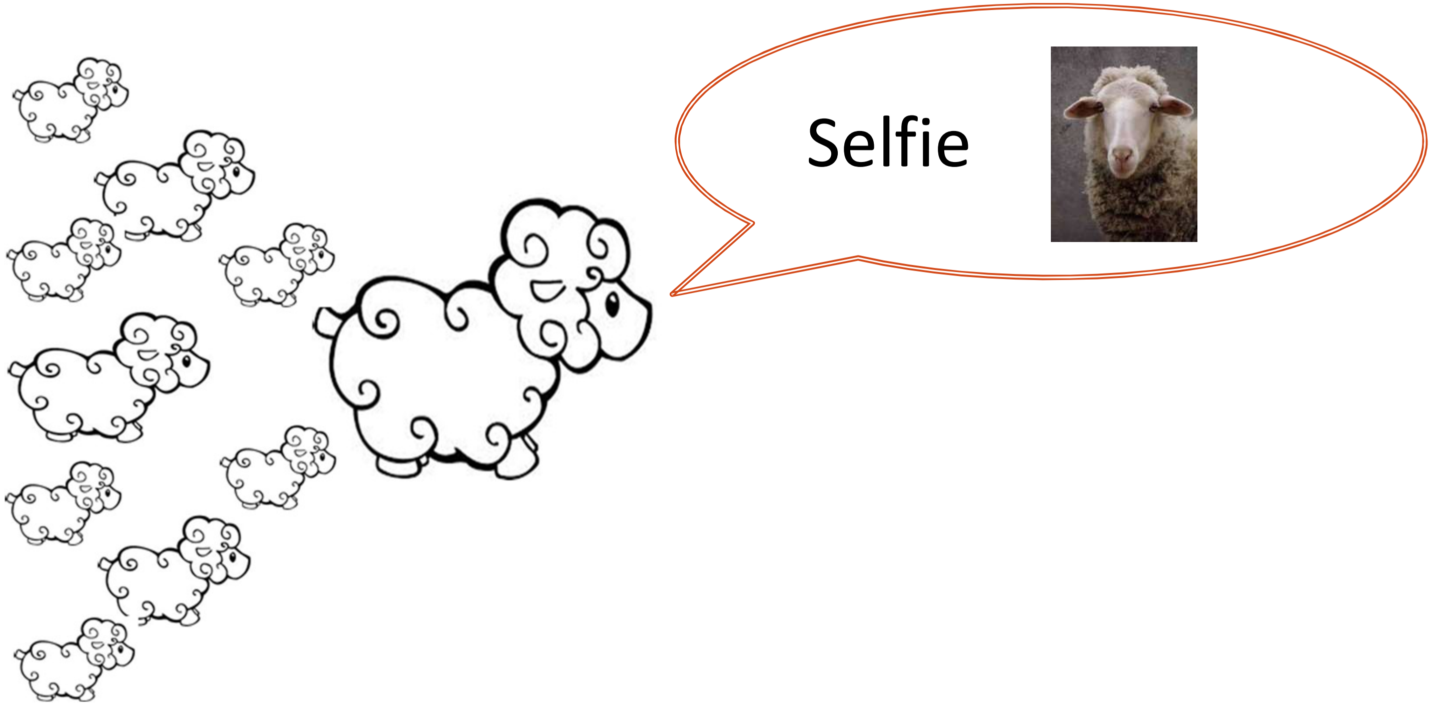
Spread of Influence in Social Networks

- Social Influence: Process in which individuals adjust their opinions/behaviors according to interactions with other people



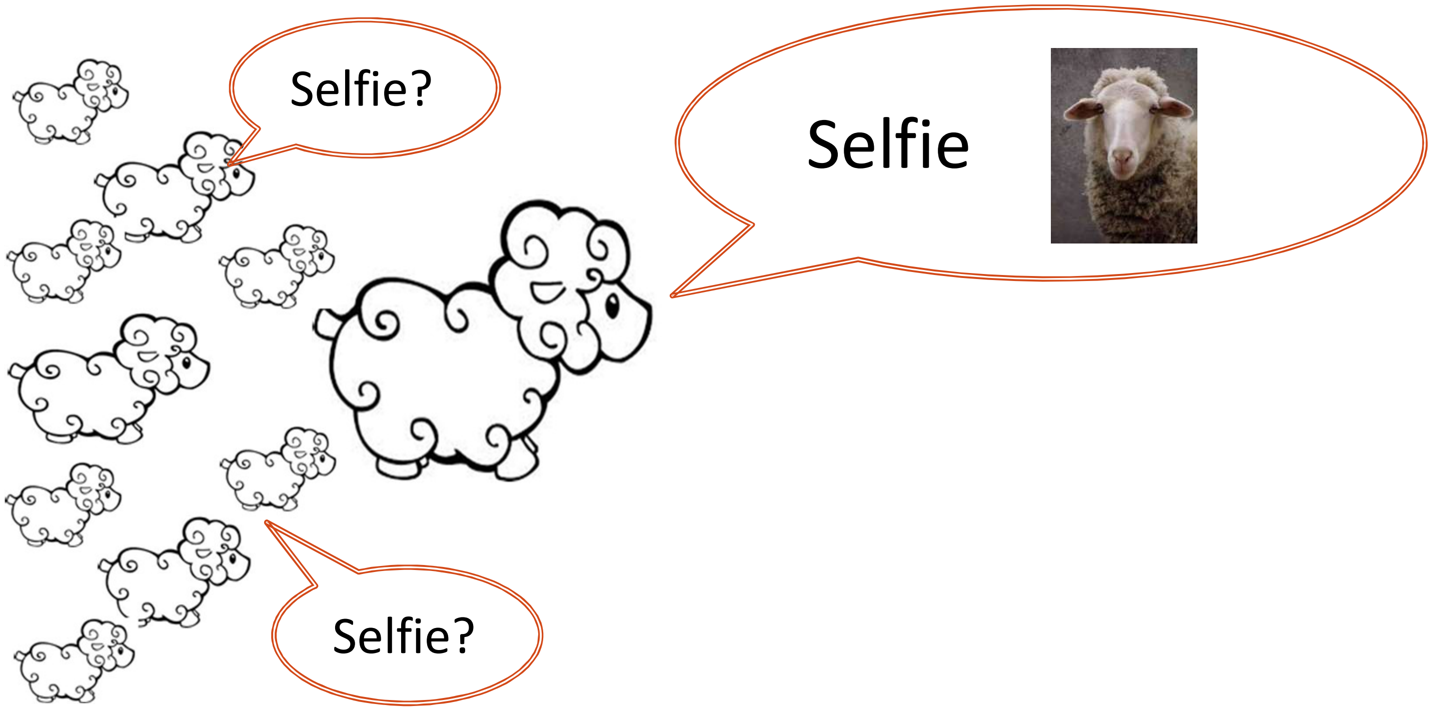
Spread of Influence in Social Networks

As individuals become Active by new ideas/products



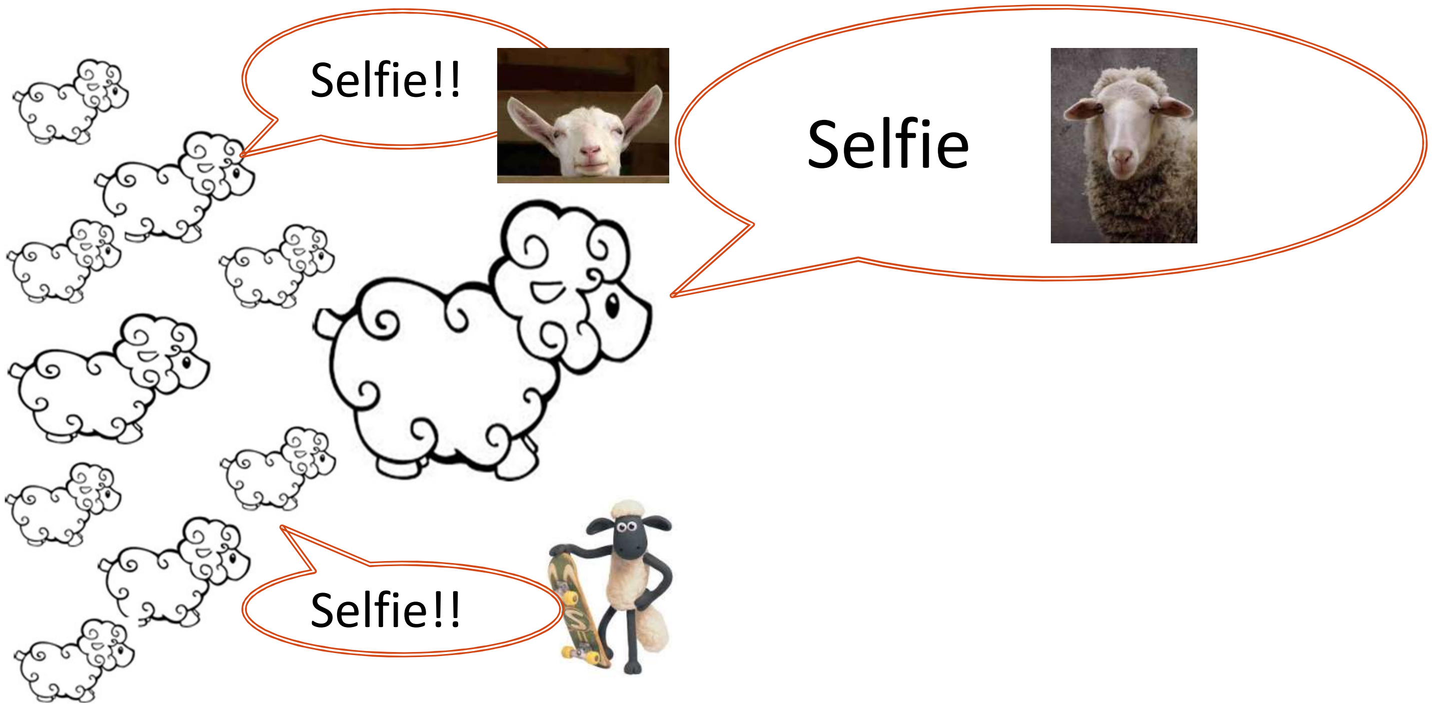
Spread of Influence in Social Networks

As individuals become Active by new ideas/products they have the potential to pass them to their friends



Spread of Influence in Social Networks

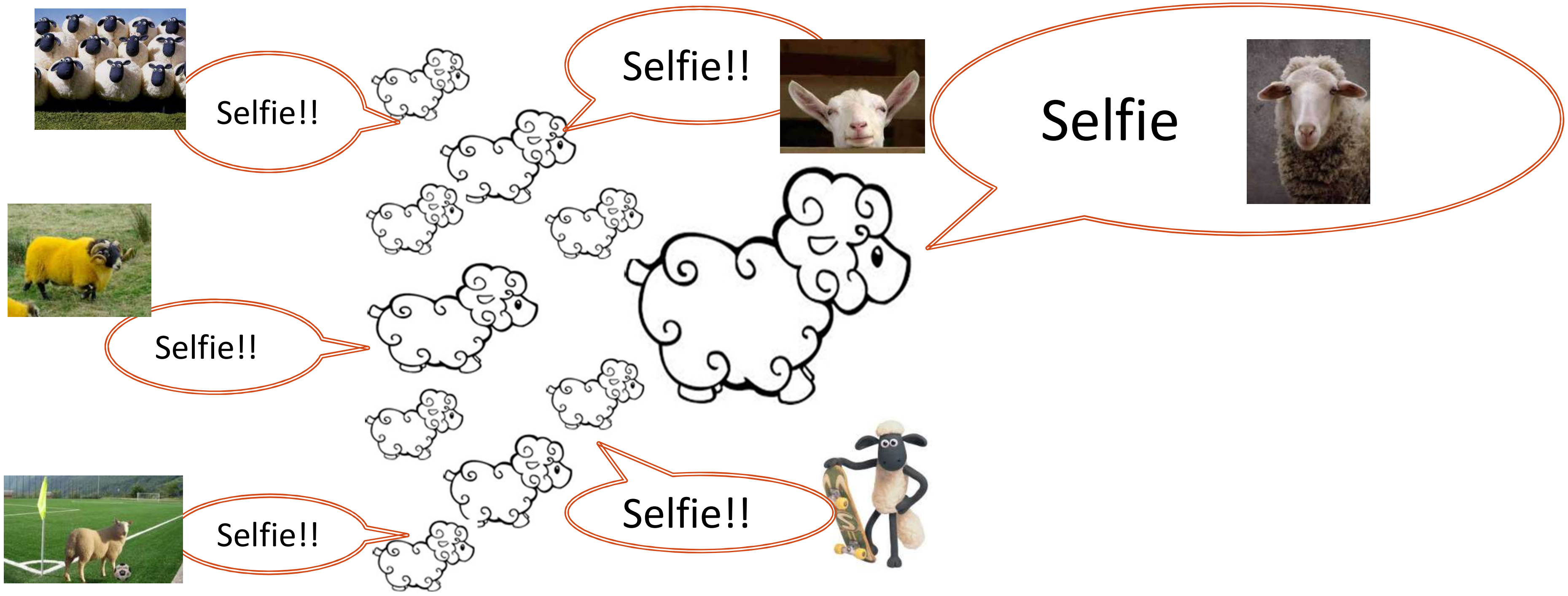
As individuals become aware of new ideas/products they have the potential to pass them to their friends



Spread of Influence in Social Networks

People have a tendency to conform

Effects of group pressure is observable even when the group is represented by a virtual community



Dynamics of Information diffusion

The Word of Mouth effect....

- Firms are increasingly recruiting customers to take part in **word-of-mouth** marketing campaigns.
- **Seeding** seeks to convert some *influential individuals* (*seeds*) who are expected to use their social network position, personal influence, and broad peer contacts to trigger cascades of product adoption.

Social Conformity

- Change a behaviour or a belief in order to fit in with a group

Dy… …on

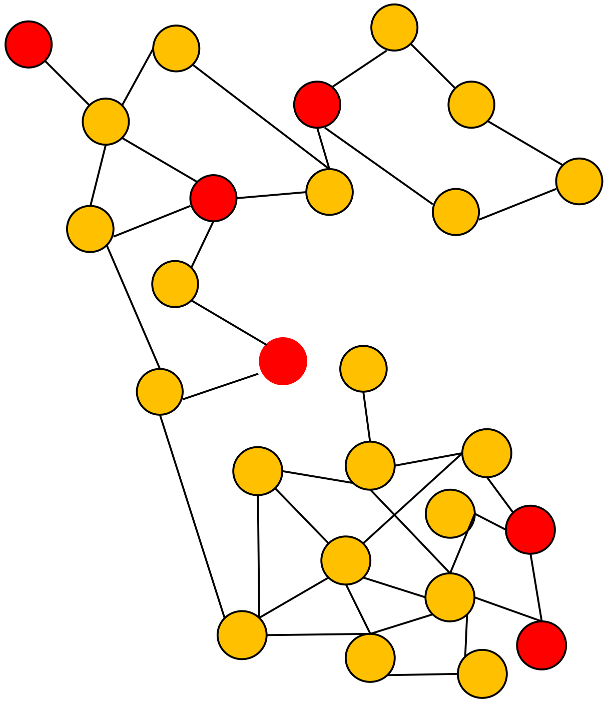
…stomers to
…ng campaigns.

…ential
… to use their
…fluence, and
…ades of

Social Confo…

- Change …
  belief in…
  group

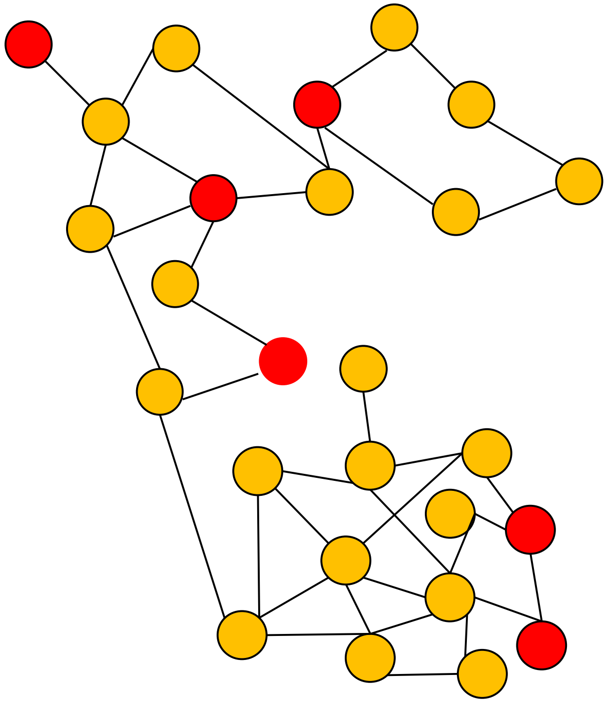
Viral Marketing



Node: User in a social network
(red – Target Set)
Edge: Friendship among users

- Given a social network, find a small number of individuals (**Target Set**), who when convinced about a ***new product*** will influence others by word-of-mouth, leading to a large number of adoptions of the product

Viral Marketing



Node: User in a social network
(red – Target Set)
Edge: Friendship among users

- **Tresholds**: a threshold *t(v)* quantifies how hard it is to influence node *v*, in the sense that
  - easy-to-influence elements of the network have ‘’low" threshold values,

  and

  - hard-to-influence elements have ‘’high" threshold

Influence Diffusion

Given a network $G = (V, E)$, a threshold function $t: V \to \{1,2,\dots\}$, and a seed set $S \subseteq V$, a dynamical process of **Influence Diffusion** on G is defined by the sequence of node subsets:

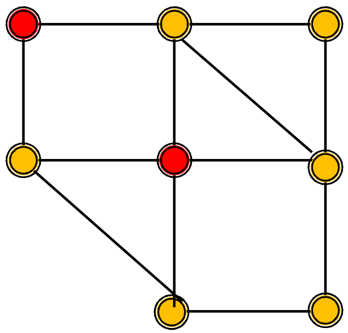
$$\textit{Influenced}[S,0],\ \textit{Influenced}[S,1],\ \dots,\ \textit{Influenced}[S,r],\ \dots \subseteq V$$

where

- $\textit{Influenced}[S,0] = S$
- $\textit{Influenced}[S,r] = \textit{Influenced}[S,r-1] \cup \{v : |N(v) \cap \textit{Influenced}[S,r-1]| \geq t(v)\}$

$N(v)$ =set of nodes adjacent to v

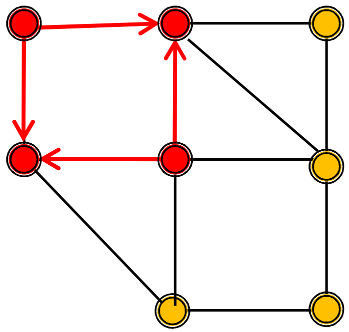
Influence Diffusion:



$t(v)$ =2 for each node v

- $Influenced[S, 0] = S$

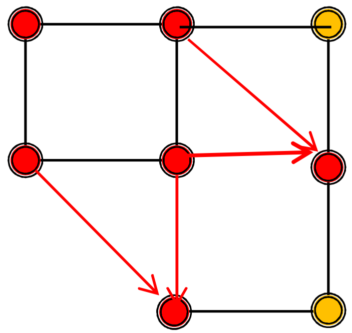
Influence Diffusion



$t(v)$ =2 for each node v

- $Influenced[S,1] = Influenced[S,0]$
$\cup \{v : |N(v) \cap\ Influenced[S,0]| \geq t(v)\}$

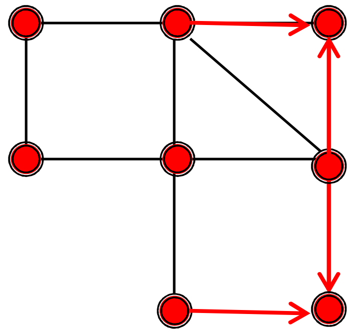
Influence Diffusion



$t(v)$ =2 for each node v

- $Influenced[S,2] = Influenced[S,1]$
  $\cup \{v : |N(v) \cap \ Influenced[S,1]| \geq t(v)\}$

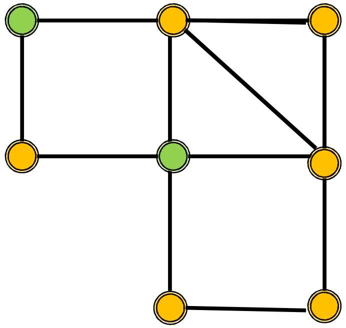
Influence Diffusion



$t(v)$ =2 for each node v

- $Influenced[S,3] = Influenced[S,2]$
  $\cup \{v : |N(v) \cap \ Influenced[S,2]| \geq t(v)\}$

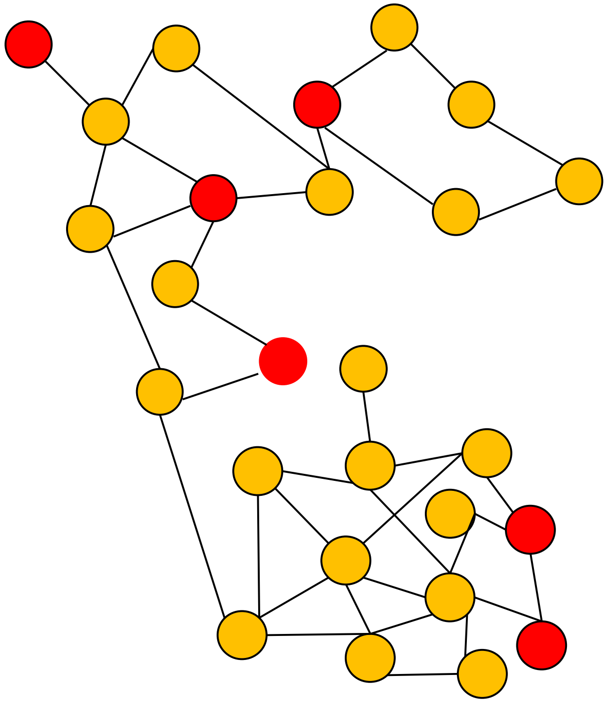
Influence Diffusion



$t(v)$ =2 for each node v

- $Influenced[\boldsymbol{S}, 3] = V$

Find smallest S such that $Influenced[S, \infty] = V$



Node: User in a social network
(red – Target Set)
Edge: Friendship among users

- We want to **flood** the network with a new product/behavior
- We want to advertise the product as less as possible
- Which is the **smallest number of early adopters**?

A target set for G is a set S such that it will activate the whole network, that is, for which it holds that Influenced[S, ℓ]=V, for some $\ell \geq 0$.

Target Set Selection

Instance: A network G = (V,E) with thresholds $t : V \to \mathbb{N}$.
Problem: Find a target set $S \subseteq V$ of minimum size for G.

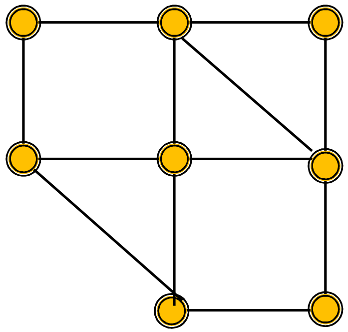
Find smallest S such that Influenced[S,∞]=V

Find smallest S such that $\mathit{Influenced}[S, \infty] = V$

EXACT RESULT

- Hard to approximate whithin a ratio of $O\left(2^{\log^{(1-\epsilon)} n}\right)$ (N. Chen, 2009)
- Fast exact algorithms for special classes of graphs: trees, cycles, complete graphs, ...

Find smallest S s.t. $\mathit{Influenced}[S,r] = V$: ***Our Heuristic***



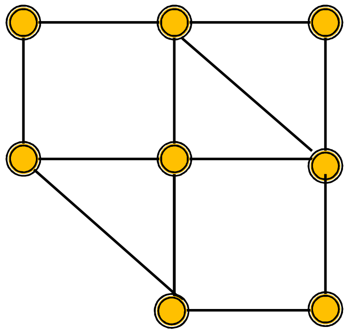
IDEA:
Given the network G
Pick one node a time
- **decide if it should go in the target set S**
- **eliminate it from G**

Algorithm TSS(G)

Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.

1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3.   if there exists $v \in V$ s.t. $t(v) = 0$ then
4   for each $u \in N(v)$ set $\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\ \}$
5.   else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6.   for each $u \in N(v)$ set$\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\}$
7.   else $v = \text{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8.   for each $u \in N(v)$ set $\{\ d(u) = d(u) - 1;\ N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

Find smallest S s.t. $Influenced[S, r] = V$: ***Our Heuristic***



$t(v)$ =2 for each node v

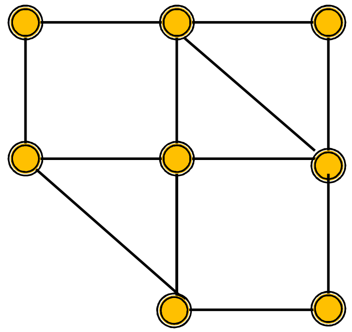
There is a node v
whose threshold is 0 !
Just eliminate v from the graph

Algorithm TSS(G)

Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.

1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3. if there exists $v \in V$ s.t. $t(v) = 0$ then
4 for each $u \in N(v)$ set $\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\ \}$
5. else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6. for each $u \in N(v)$ set$\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\}$
7. else $v = \text{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8. for each $u \in N(v)$ set $\{\ d(u) = d(u) - 1;\ N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

Find smallest S s.t. *Influenced*$[S, r] = V$: ***Our Heuristic***



$t(v) = 2$ **for each node** v

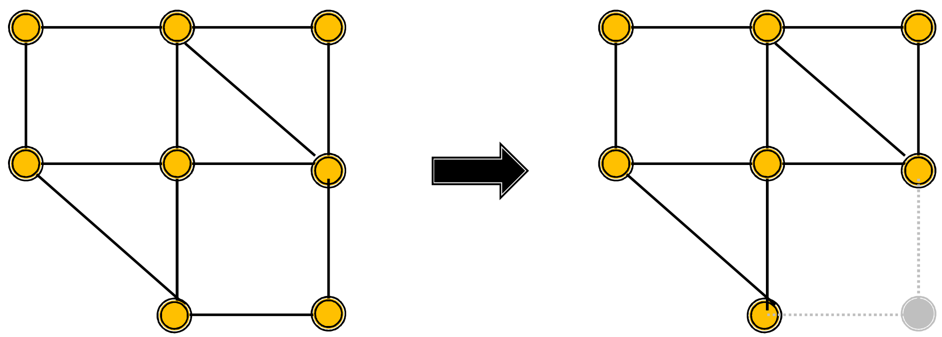
There is an node v with threshold larger than the degree! Add v to S before eliminating from the graph

Algorithm TSS(G)

Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.

1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3. if there exists $v \in V$ s.t. $t(v) = 0$ then
4 for each $u \in N(v)$ set $\{ t(u) = t(u) - 1; d(u) = d(u) - 1, N(u) = N(u) - \{v\} \}$
5. else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6. for each $u \in N(v)$ set$\{ t(u) = t(u) - 1; d(u) = d(u) - 1, N(u) = N(u) - \{v\}\}$
7. else $v = \text{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8. for each $u \in N(v)$ set $\{ d(u) = d(u) - 1; N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

Find smallest S s.t. $Influenced[S, r] = V$: ***Our Heuristic***



$t(v)$ =2 for each node v

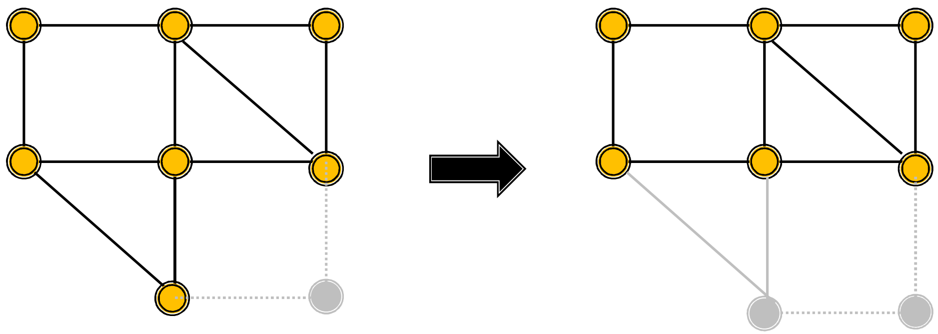
Otherwise
Pick v vith largest
ratio and eliminate it

Algorithm TSS(G)

Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.

1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3. if there exists $v \in V$ s.t. $t(v) = 0$ then
4 for each $u \in N(v)$ set $\{ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\} \}$
5. else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6. for each $u \in N(v)$ set$\{ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\}$
7. else $v = \text{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8. for each $u \in N(v)$ set $\{ d(u) = d(u) - 1;\ N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

Find smallest S s.t. $Influenced[S, r] = V$: ***Our Heuristic***



$t(v) = 2$ **for each node** v

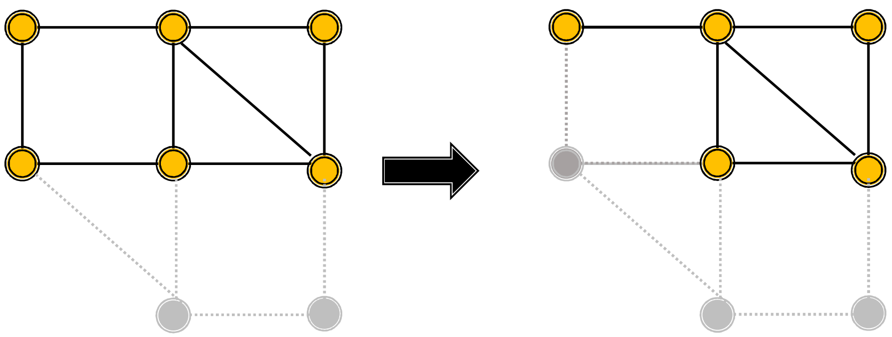
Otherwise
Pick v vith largest
ratio and eliminate it

Algorithm TSS(G)

Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.

1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3. if there exists $v \in V$ s.t. $t(v) = 0$ then
4 for each $u \in N(v)$ set $\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\ \}$
5. else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6. for each $u \in N(v)$ set$\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\}$
7. else $v = \text{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8. for each $u \in N(v)$ set $\{\ d(u) = d(u) - 1;\ N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

Find smallest S s.t. $Influenced[S, r] = V$: ***Our Heuristic***



$t(v) = 2$ **for each node** v

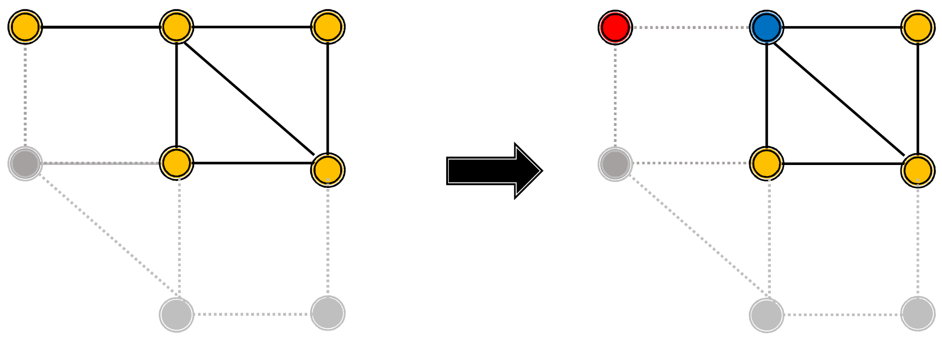
Otherwise
Pick v vith largest
ratio and eliminate it

Algorithm TSS(G)

Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.

1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3. if there exists $v \in V$ s.t. $t(v) = 0$ then
4 for each $u \in N(v)$ set $\{ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\} \}$
5. else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6. for each $u \in N(v)$ set$\{ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\}$
7. else $v = \text{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8. for each $u \in N(v)$ set $\{ d(u) = d(u) - 1;\ N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

Find smallest S s.t. $Influenced[S,r] = V$: ***Our Heuristic***

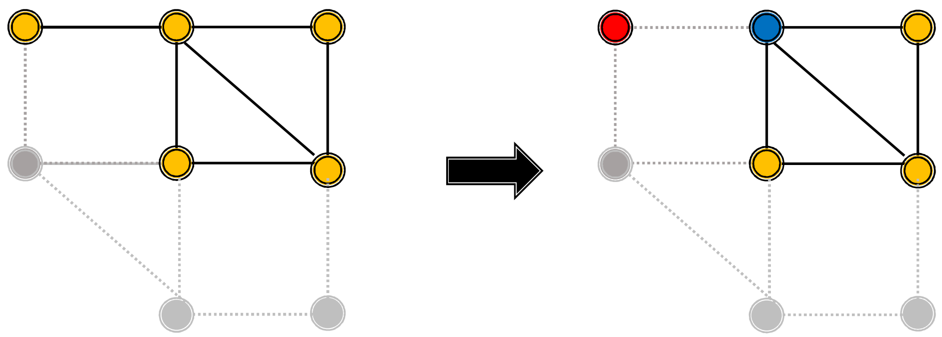


$t(v)$ =2 for each node v

There is an node v with threshold larger than the degree!
Add v to S before eliminating from the graph

Algorithm TSS(G)
Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.
1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3. if there exists $v \in V$ s.t. $t(v) = 0$ then
4 for each $u \in N(v)$ set $\{ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\} \}$
5. else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6. for each $u \in N(v)$ set$\{ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\}$
7. else $v = \text{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8. for each $u \in N(v)$ set $\{ d(u) = d(u) - 1;\ N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

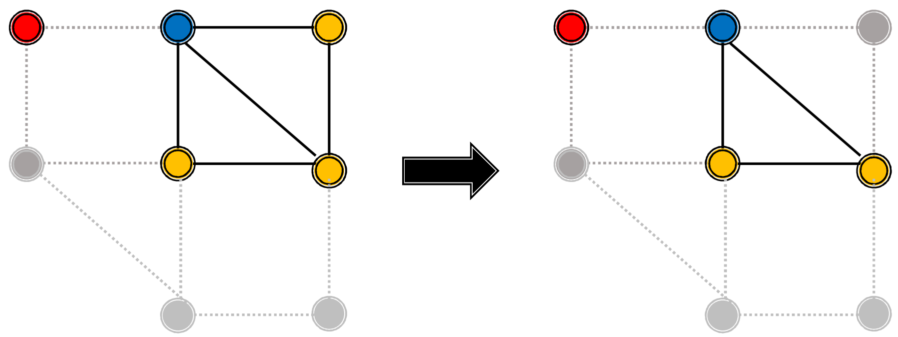
Find smallest S s.t. $Influenced[S, r] = V$: ***Our Heuristic***



$t(v) = 2$ for each yellow node v,
$t(v) = 1$ for blue node

```
Algorithm TSS(G)
Input: A graph G = (V, E) with thresholds t(v), and node degrees d(v), for v ∈ V.
Set S = ∅
  while V ≠ ∅ do
     if there exists v ∈ V s.t. t(v) = 0 then
        for each u ∈ N(v) set { t(u) = t(u) − 1; d(u) = d(u) − 1, N(u) = N(u) − {v} }
     else if there exists v ∈ U s.t. d(v) < t(v) then S = S ∪ {v}
           for each u ∈ N(v) set{ t(u) = t(u) − 1; d(u) = d(u) − 1, N(u) = N(u) − {v}}
        else v = argmax_{u∈U} { t(u) / (d(u)(d(u)+1)) }
           for each u ∈ N(v) set { d(u) = d(u) − 1; N(u) = N(u) − {v}}
     V = V − {v}
```

Find smallest S s.t. $Influenced[S, r] = V$: ***Our Heuristic***

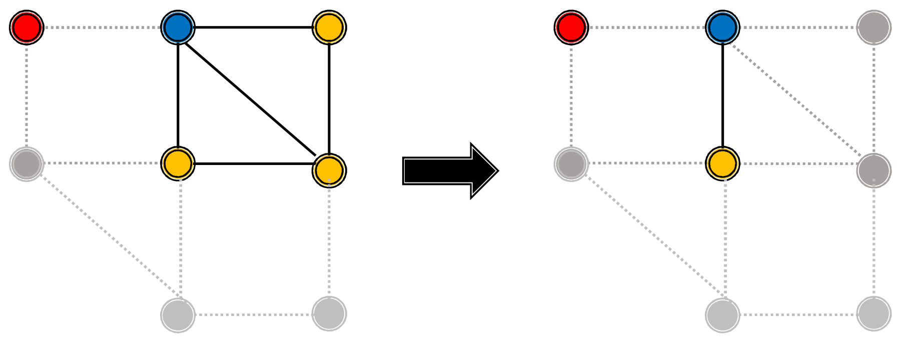


$t(v) = 2$ for each yellow node v,
$t(v) = 1$ for blue node

Pick v vith largest ratio and eliminate it

Algorithm TSS(G)
Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.
1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3. if there exists $v \in V$ s.t. $t(v) = 0$ then
4 for each $u \in N(v)$ set $\{ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\} \}$
5. else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6. for each $u \in N(v)$ set$\{ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\}$
7. else $v = \text{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8. for each $u \in N(v)$ set $\{ d(u) = d(u) - 1;\ N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

Find smallest S s.t. $Influenced[S, r] = V$: ***Our Heuristic***

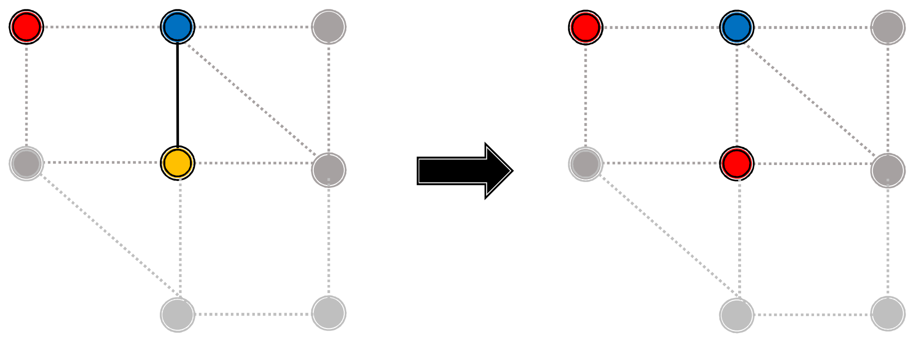


$t(v) = 2$ for each yellow node v,
$t(v) = 1$ for blue node

Pick v vith largest ratio and eliminate it

Algorithm TSS(G)
Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.
1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3. if there exists $v \in V$ s.t. $t(v) = 0$ then
4 for each $u \in N(v)$ set $\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\ \}$
5. else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6. for each $u \in N(v)$ set$\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\}$
7. else $v = \text{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8. for each $u \in N(v)$ set $\{\ d(u) = d(u) - 1;\ N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

Find smallest S s.t. $Influenced[S, r] = V$: ***Our Heuristic***



$\boldsymbol{t(v)} = \mathbf{2}$ **for each yellow node v,**
$\boldsymbol{t(v)} = \mathbf{1}$ **for blue node**

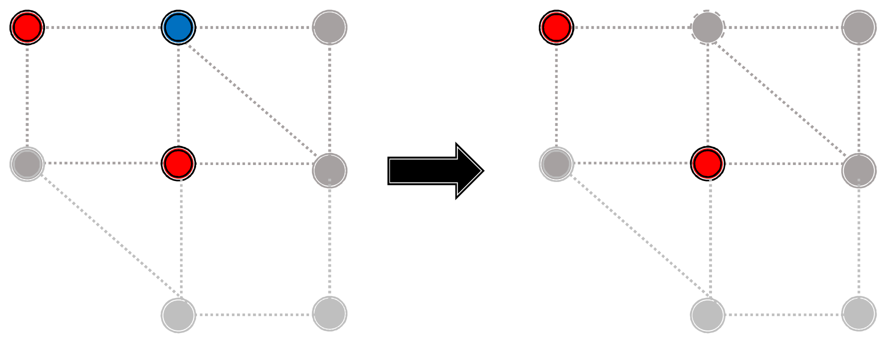
There is an node v with threshold larger than the degree!
Add v to S before eliminating from the graph

Algorithm TSS(G)

Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.

1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3. if there exists $v \in V$ s.t. $t(v) = 0$ then
4 for each $u \in N(v)$ set $\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\ \}$
5. else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6. for each $u \in N(v)$ set$\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\}$
7. else $v = \mathrm{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8. for each $u \in N(v)$ set $\{\ d(u) = d(u) - 1;\ N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

Find smallest S s.t. $Influenced[S, r] = V$: ***Our Heuristic***



$t(v)$ =1 **for blue node**

S= red nodes

There is a node v whose threshold is 0 ! Just eliminate v from the graph

Algorithm TSS(G)

Input: A graph $G = (V, E)$ with thresholds $t(v)$, and node degrees $d(v)$, for $v \in V$.

1. Set $S = \emptyset$
2. while $V \neq \emptyset$ do
3. if there exists $v \in V$ s.t. $t(v) = 0$ then
4 for each $u \in N(v)$ set $\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\ \}$
5. else if there exists $v \in U$ s.t. $d(v) < t(v)$ then $S = S \cup \{v\}$
6. for each $u \in N(v)$ set$\{\ t(u) = t(u) - 1;\ d(u) = d(u) - 1,\ N(u) = N(u) - \{v\}\}$
7. else $v = \text{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
8. for each $u \in N(v)$ set $\{\ d(u) = d(u) - 1;\ N(u) = N(u) - \{v\}\}$
9. $V = V - \{v\}$

Some theoretical results

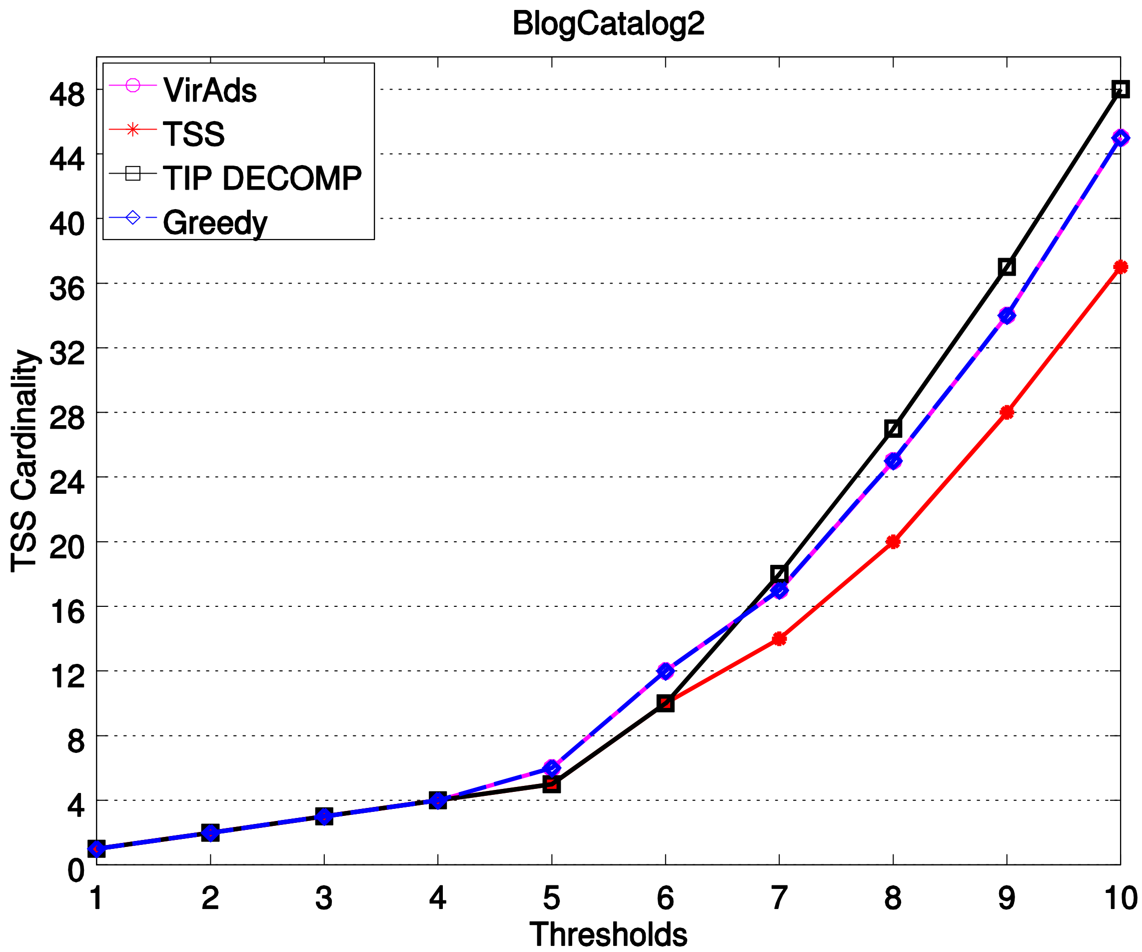
Our algorithm gives a unified proof for several results in the literature, obtained by means of different ad hoc algorithms/techniques

Theorem Algorithm TSS(G) always outputs a Target Set S of size

$$|S| \le \sum_{v \in V} \frac{t(v)}{d(v)+1}$$

Theorem Algorithm TSS(G) always outputs a minimum size Target Set whenever G is either a Tree, or a Cycle or a Clique

Some Computational results §

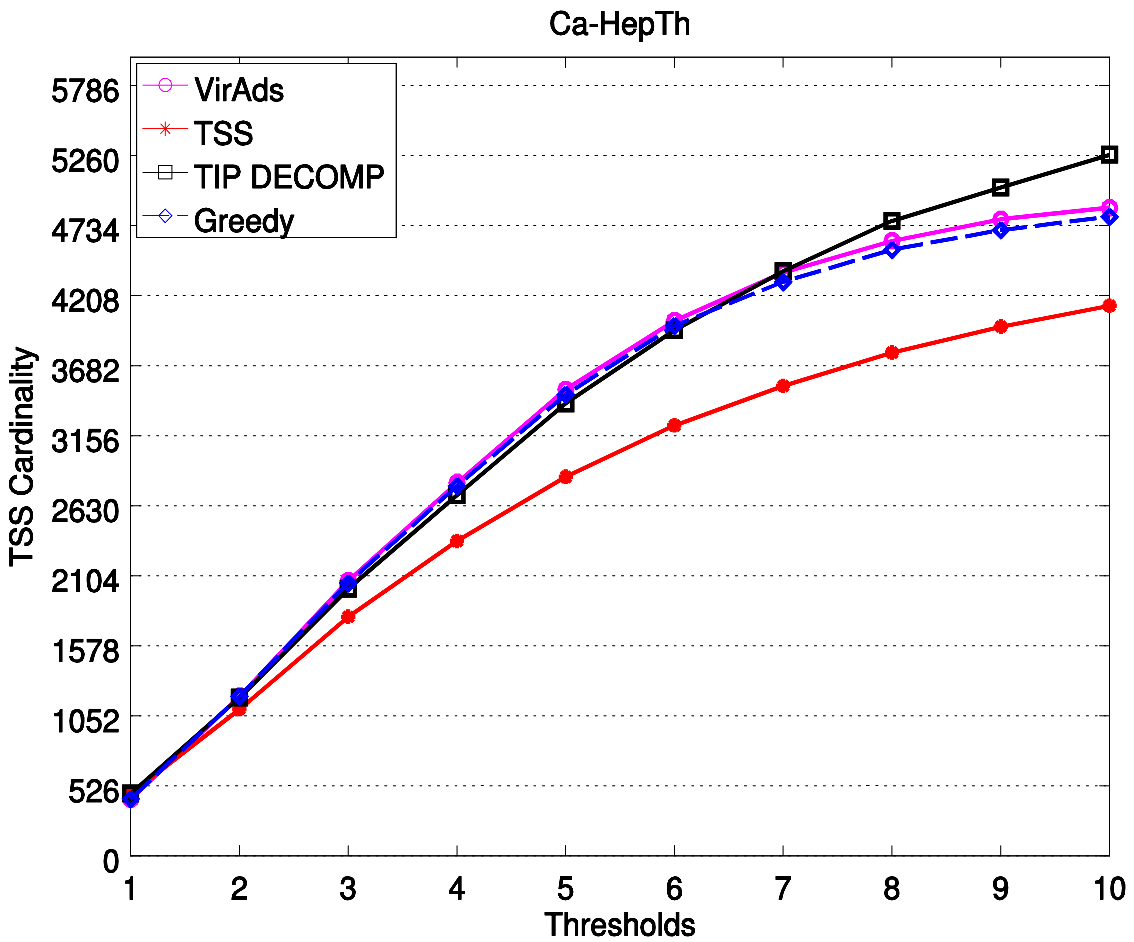


BlogCatalog is the social blog directory which manages the bloggers and their blogs. It has 97,884 nodes and 2,043,701 edges.

§Comparisons of the results of **our algoritm** to

- **Greedy Algorithm**
- ***[P. Shakarian et al, A scalable heuristic for viral marketing, Social Network Analysis and Mining, Springer, 2013.]***
- **[T. N. Dinh et al., Cost-Effective Viral Marketing for Time-Critical Campaigns in Large-Scale Social Networks, IEEE/ACM Trans. on Networking, 2014]**

Some Computational results §

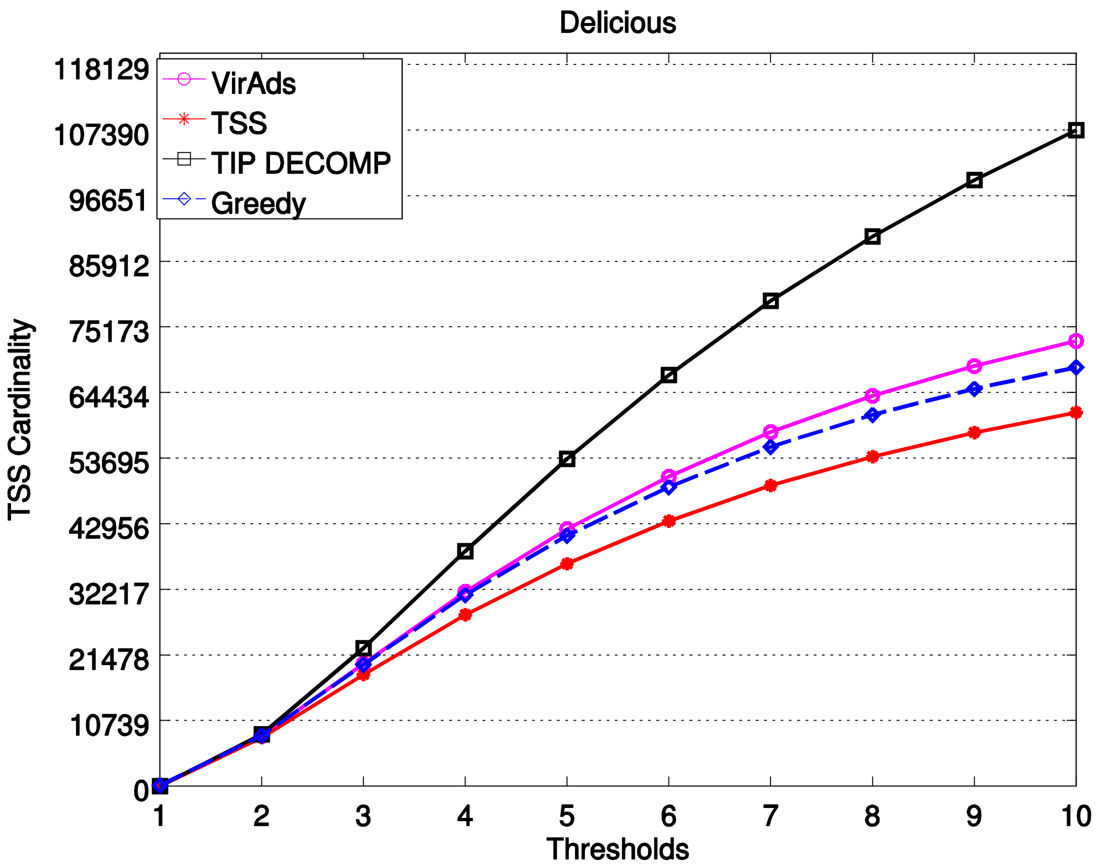


Collaboration network of Arxiv papers submitted to High Energy Physics – Theory category

§Comparisons of the results of **our algoritm** to

- **Greedy Algorithm**
- **[*P. Shakarian et al, A scalable heuristic for viral marketing, Social Network Analysis and Mining, Springer, 2013.*]**
- **[T. N. Dinh et al., Cost-Effective Viral Marketing for Time-Critical Campaigns in Large-Scale Social Networks, IEEE/ACM Trans. on Networking, 2014]**

Some Computational results §



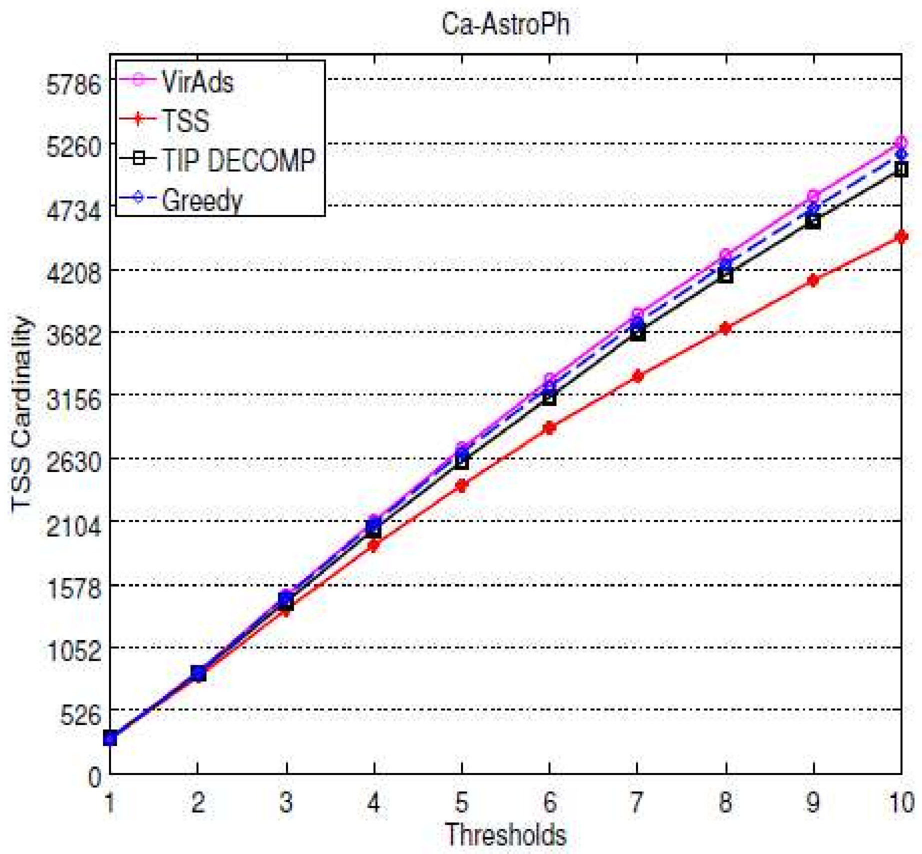
Delicious is a social bookmarking web service for storing, sharing, and discovering web bookmarks.

It has 103.144 nodes and 1.419.519 edges.

§Comparisons of the results of **our algoritm** to

- **Greedy Algorithm**
- **[*P. Shakarian et al, A scalable heuristic for viral marketing, Social Network Analysis and Mining, Springer, 2013.*]**
- **[T. N. Dinh et al., Cost-Effective Viral Marketing for Time-Critical Campaigns in Large-Scale Social Networks, IEEE/ACM Trans. on Networking, 2014]**

Some Computational results §



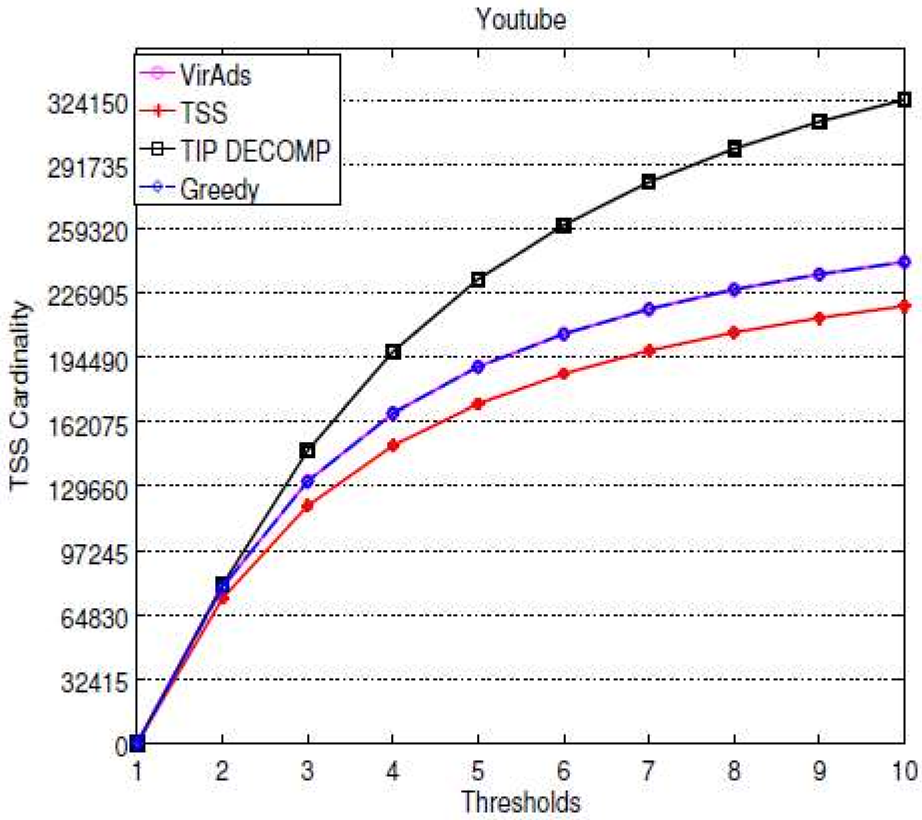
CA-Astro-Ph is a collaboration network of Arxiv ASTRO-PH (Astro Physics).

It has 18.777 nodes and 198.110 edges.

§Comparisons of the results of **our algoritm** to

- **Greedy Algorithm**
- **[*P. Shakarian et al, A scalable heuristic for viral marketing, Social Network Analysis and Mining, Springer, 2013.]***
- **[T. N. Dinh et al., Cost-Effective Viral Marketing for Time-Critical Campaigns in Large-Scale Social Networks, IEEE/ACM Trans. on Networking, 2014]**

Some Computational results §



YouTube2 is a data set crawled from YouTube. In the YouTube social network users form friendship each other and users can create groups which other users can join.

It has 1,138,499 nodes and 2,990,443 edges.

§Comparisons of the results of **our algoritm** to

- **Greedy Algorithm**
- ***[P. Shakarian et al, A scalable heuristic for viral marketing, Social Network Analysis and Mining, Springer, 2013.]***
- **[T. N. Dinh et al., Cost-Effective Viral Marketing for Time-Critical Campaigns in Large-Scale Social Networks, IEEE/ACM Trans. on Networking, 2014]**

Targeting with Partial Incentives

Gennaro Cordasco , Luisa Gargano , Manuel Lafond, Lata Narayanan, Adele A. Rescigno , Ugo Vaccaro, Kangkang Wu:
Whom to befriend to influence people.
Theor. Comput. Sci. 810: 26-42 (2020)

Each node $v \in V$ is associated with a non negative value $c(v)$ measuring how much it costs to initially convince the member v of the network to endorse a given product/behavior

Weighted Target Set Selection

Instance: A network $G = (V,E)$ with thresholds $t : V \rightarrow N$ and costs $c : V \rightarrow N$

Problem: Find a target set $S \subseteq V$ of minimum cost $C(S)=\sum_{v \in S} c(v)$ for G.

Find the less costly S such that Active[S,∞]=V

Targeting with Partial Incentives.

- The classical problem forces the optimizer to make a 0/1 choises on each node
- In a realistic scenario there could be more reasonable and effective solutions (partial incentives)

Targeting with Partial Incentives.

- An assignment of partial incentives to the vertices of a network $G = (V,E)$, with $V = \{v_1, \ldots, v_n\}$, is a vector $\mathbf{s} = (s(v_1), \ldots, s(v_n))$, where $s(v) \in \{0, 1, 2, \ldots\}$ represents the amount of influence we initially apply on $v \in V$.

- An activation process in G starting with a vector of incentives $\mathbf{s}$ is a sequence $\text{Active}[\mathbf{s},0] \subseteq \text{Active}[\mathbf{s},1] \subseteq \ldots \subseteq \text{Active}[\mathbf{s},\ell] \subseteq \ldots \subseteq V$ of vertex subsets, with
  - $\text{Active}[\mathbf{s},0] = \{v : s(v) \geq t(v)\}$, and such that for all $\ell > 0$,
  - $\text{Active}[\mathbf{s},\ell] = \text{Active}[\mathbf{s},\ell-1] \cup \{u : |N(u) \cap \text{Active}[\mathbf{s}, \ell-1]| \geq t(u)-s(u)\}$

A target vector **s** is an assignment of partial incentives that triggers an activation process influencing the whole network, that is, such that Active[**s**, ℓ] = V for some $\ell \geq 0$.

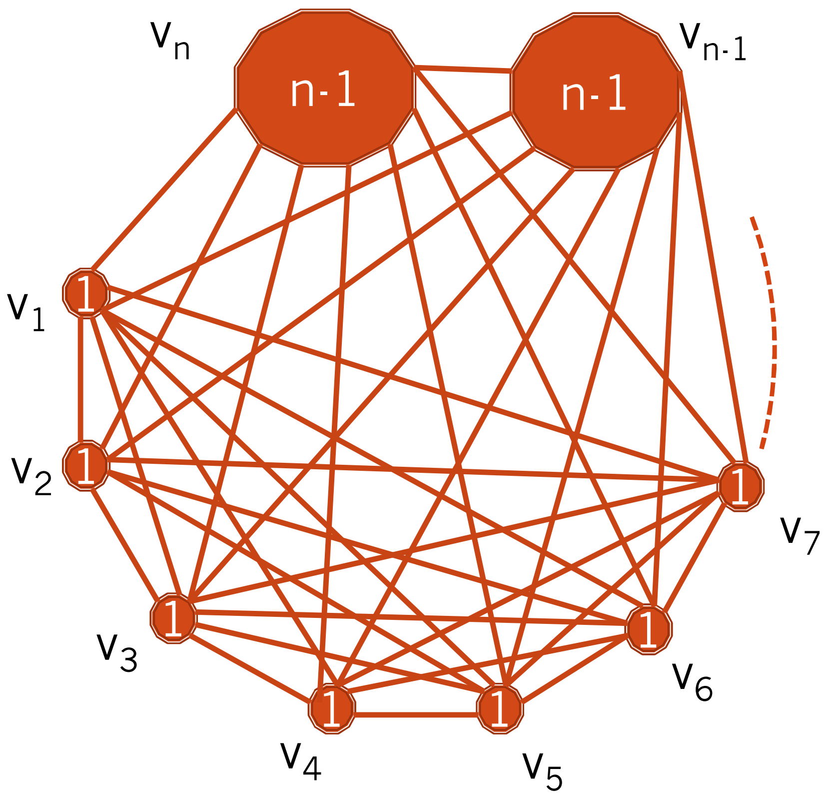
Targeting with Partial Incentives

Instance: A network G = (V,E) with thresholds t : V → N.

Problem: Find a target vector **s** which minimizes C(**s**)=$\sum_{v \in V} s(v)$

Find the target vector **s** that minimizes the total incentives to have Active[**s**,∞]=V

An Example



- An optimal solution for the WTSS problem is $S=\{v_n\}$.
  - $C(S)=n-1$
- An optimal solution for the TPI problem is $\mathbf{s}(v_1)=\mathbf{s}(v_n)=1$, otherwise $\mathbf{s}(v)=0$;
  - $C(\mathbf{s})=2$

Hardness of WTSS and TPI

- **Theorem** WTSS and TPI cannot be approximated within a ratio of $O(2^{log^{1-\varepsilon}|V|})$ for any fixed $\varepsilon > 0$, unless NP ⊆ DTIME($n^{polylog(n)}$).

Algorithm TPI(G)

Input: A graph $G = (V, E)$ with thresholds $t(v)$, for each $v \in V$.

Output: **s** a target vector for G.

$U = V$

for each $v \in V$ **do**

    $s(v) = 0$ *[Partial incentive initially assigned to v]*

    $\delta(v) = d_G(v)$

    $k(v) = t(v)$

    $N(v) = \Gamma_G(v)$

while $U \neq \emptyset$ **do**

    [Select one vertex and either update its incentive or remove it from the graph]

    if there exists $v \in U$ s.t. $k(v) > \delta(v)$

        then *[Case 1: Increase s(v) and update k(v)]*

            $s(v) = s(v) + k(v) - \delta(v)$

            $k(v) = \delta(v)$

            if $k(v) = 0$ **then** $U = U - \{v\}$ *[here $\delta(v) = 0$]*

        else *[Case 2: Choose a vertex v to eliminate from the graph]*

            $v = \mathtt{argmax}_{u \in U} \left\{ \frac{k(u)(k(u)+1)}{\delta(u)(\delta(u)+1)} \right\}$

            for each $u \in N(v)$ **do** $\{\delta(u) = \delta(u) - 1; N(u) = N(u) - \{v\}\}$

            $U = U - \{v\}$

TPI Results

- Correcteness
  - **Theorem**
    For any graph G and threshold function t, the algorithm TPI(G) outputs a target vector for G.

- Efficiency
  - **Theorem**
    For any G = (V,E), the algorithm TPI(G) returns a target vector s with
    $$C(s) \leq \sum_{v \in V} \frac{t(v)(t(v)+1)}{2(d(v)+1)}$$

- Proved to be optimal on complete graphs and trees

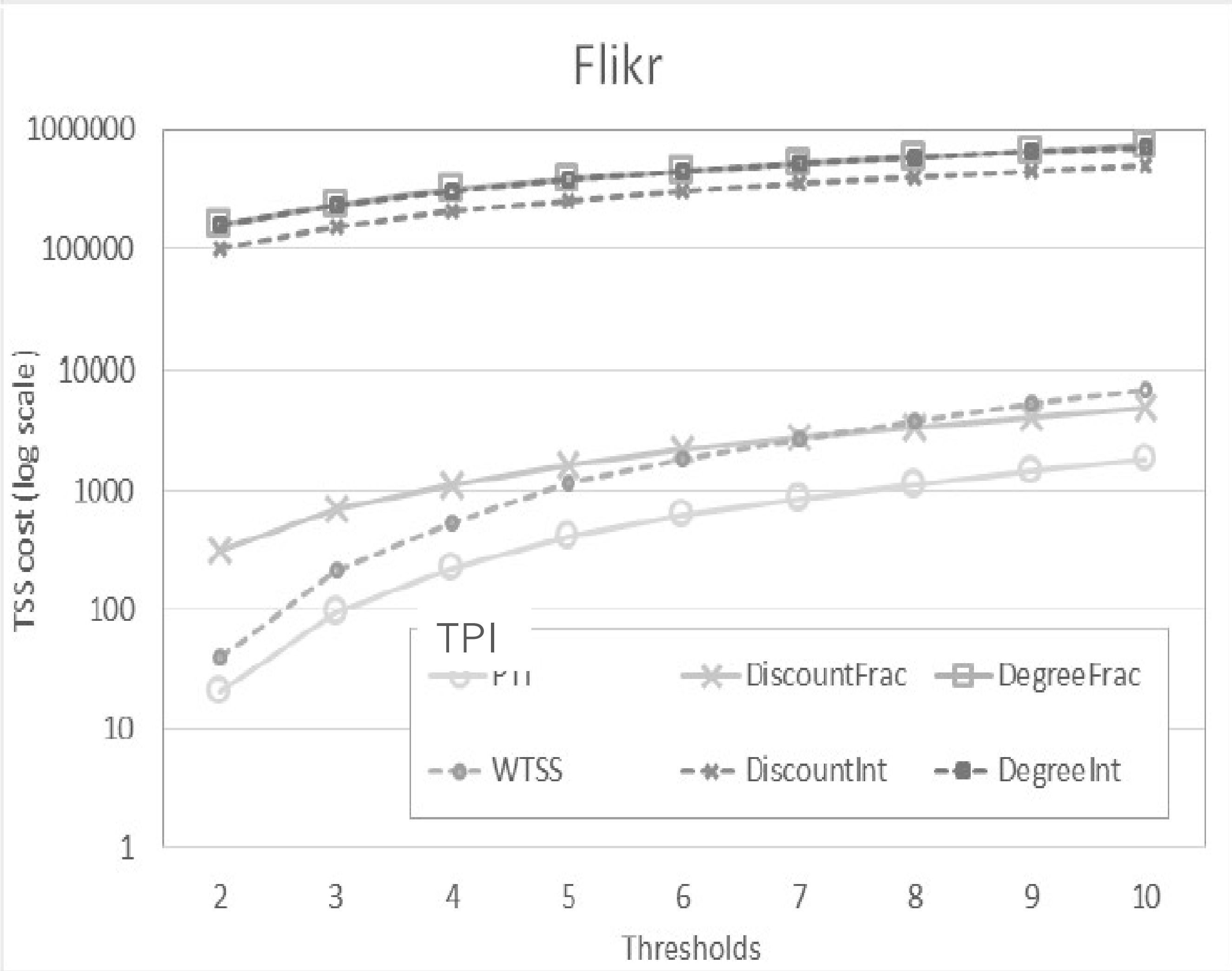
Experiments

- Test Setting
  - **18 real network** form
    - Stanford Large Network Dataset Collection (SNAP)
    - the Social Computing Data Repository at Arizona State University
    - Newman's Network data
  - **4 Competiting Algorithms**
    - DegreeInt [Demaine et al 2014]
    - DiscountInt[Chen et al 2009]
    - DegreeFrac[Demaine et al 2014]
    - DiscountFrac[Demaine et al 2014]
  - **Thresholds**
    - Random
    - Constant
    - Degree Proportional

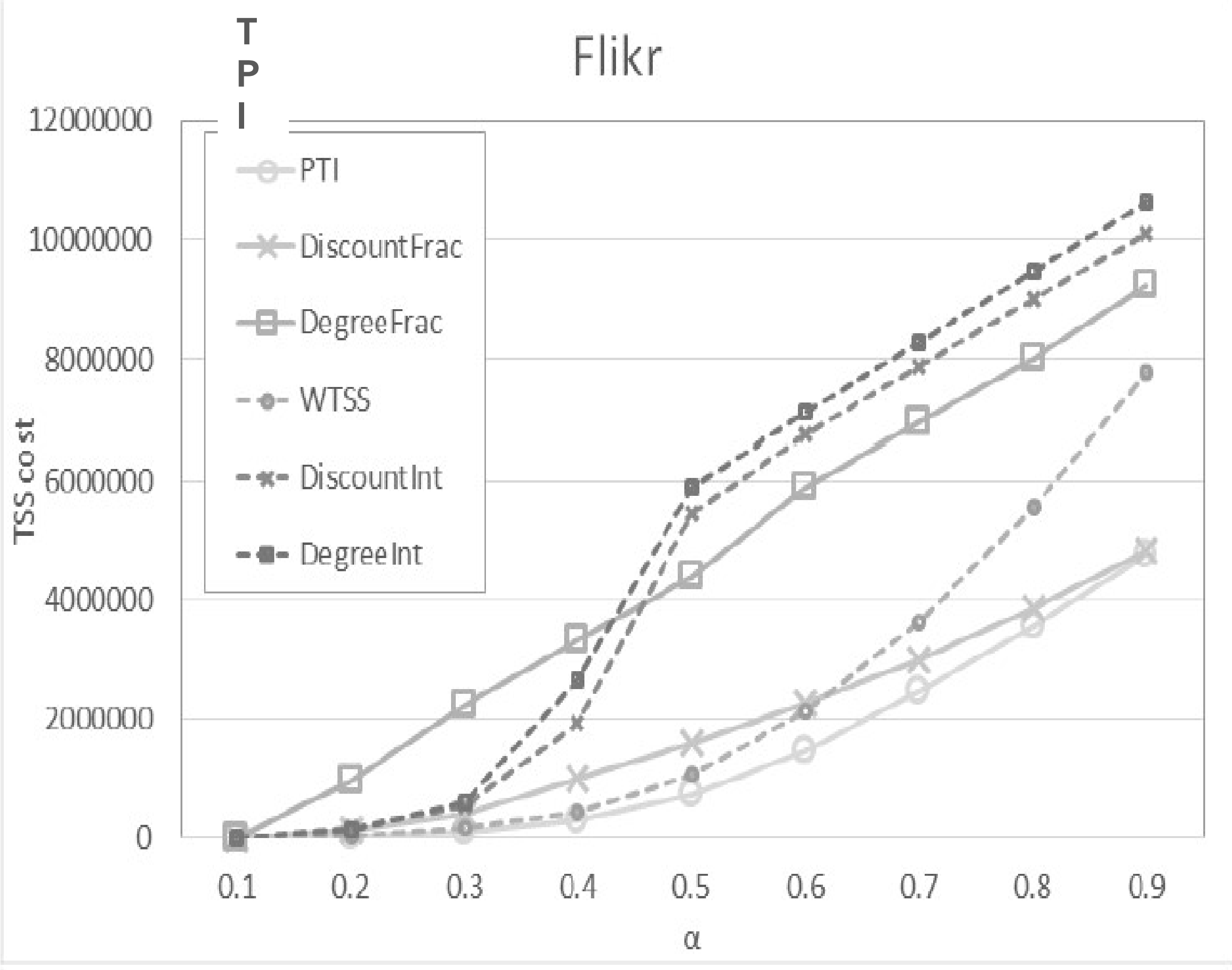
	Targeting with Partial Incentives			Weighted Target Set Selection with $c(\cdot) = t(\cdot)$		
Name	TPI	DiscountFrac	DegreeFrac	WTSS	DiscountInt	DegreeInt
Amazon0302	52703	328519 (623%)	879624 (1669%)	85410	596299 (698%)	890347 (1042%)
BlogCatalog	21761	824063 (3787%)	980670 (4507%)	82502	1799719 (2181%)	2066014 (2504%)
BlogCatalog2	16979	703383 (4143%)	178447 (1051%)	67066	1095580 (1634%)	1214818 (1811%)
BlogCatalog3	161	3890 (2416%)	3113 (1934%)	3925	3890 (99%)	3890 (99%)
BuzzNet	50913	1154952 (2268%)	371355 (729%)	166085	1838430 (1107%)	2580176 (1554%)
ca-AstroPh	4520	67189 (1486%)	198195 (4385%)	13242	183121 (1383%)	198195 (1497%)
ca-CondMath	5694	31968 (561%)	94288 (1656%)	10596	76501 (722%)	94126 (888%)
ca-GrQc	1422	5076 (357%)	15019 (1056%)	2141	12538 (586%)	15019 (701%)
ca-HepPh	4166	42029 (1009%)	120324 (2888%)	11338	118767 (1048%)	120324 (1061%)
ca-HepTh	2156	9214 (427%)	26781 (1242%)	3473	25417 (732%)	26781 (771%)
Douban	51167	140676 (275%)	345036 (674%)	91342	194186 (213%)	252739 (277%)
Facebook	1658	29605 (1786%)	54508 (3288%)	5531	77312 (1398%)	86925 (1572%)
Flikr	31392	2057877 (6555%)	134017 (427%)	110227	5359377 (4862%)	5879532 (5334%)
Hep	4122	11770 (286%)	33373 (810%)	5526	33211 (601%)	33373 (604%)
LastFM	296083	1965839 (664%)	4267035 (1441%)	631681	2681610 (425%)	4050280 (641%)
Livemocha	26610	861053 (3236%)	459777 (1728%)	57293	1799468 (3141%)	2189760 (3822%)
Power grid	767	2591 (338%)	4969 (648%)	974	3433 (352%)	4350 (447%)
Youtube2	313786	1210830 (386%)	3298376 (1051%)	576482	2159948 (375%)	3285525 (570%)

Table 1. Random Threshold Results.

Results: Flikr, Constant Thresholds



Results: Flikr, Degree Proportional Thresholds

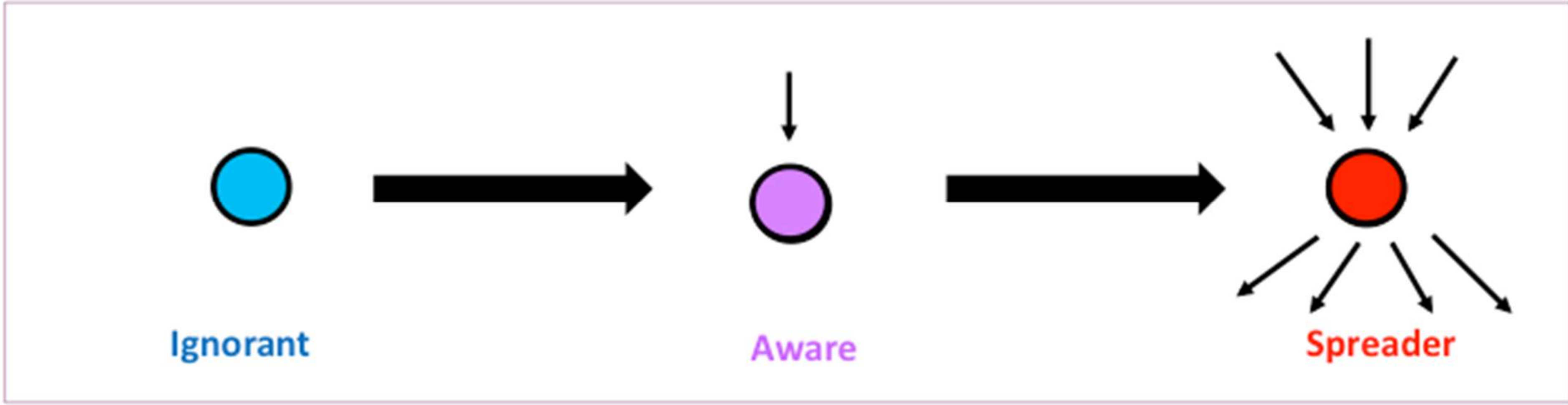


Perfect Awareness

Gennaro Cordasco , Luisa Gargano , Adele A. Rescigno :
Active influence spreading in social networks.
Theor. Comput. Sci. 764: 15-29 (2019)

Active Diffusion Spread: A 3-states model

- In LT nodes have two possible states (Inactive and Active)
- We propose a novel model (Active Diffusion Spread) which considers three states:
  - Ignorant
  - Aware
  - Spreader



Active Diffusion Process

- Given a social network $G(V,E)$ and threshold function $t : V \to N_0$
- An Active Diffusion Process starting at $S \subseteq V$ is a sequence of node subsets: $Spreader[S, \tau]$, $\tau = 0, 1, \ldots,$ such that
  - $Spreader[S, 0] = S$
  - $Spreader[S, \tau] = Spreader[S, \tau-1] \cup$
    $\{ u : |N(u) \cap Spreader[S, \tau-1]| \geq t(u)\}$, for $\tau \geq 1$
- The process terminates when $Spreader[S,\rho] = Spreader[S,\rho-1]$ for some $\rho>1$. We denote by $Spreader[S] = Spreader[S, \rho]$.
- When the process stops the set of aware nodes is
  - $Aware[S] = Spreader[S] \cup \{u : N(u) \cap Spreader[S] \neq \emptyset\}$.

Perfect Awareness problem (PA)

- Identify the smallest set $S \subseteq V$ (perfect seed set) needed to achieve awareness of the whole network $Aware[S]=V$.

Complexity

Theorem *The PA problem cannot be approximate within a ratio of $O(2^{\log^{1-\varepsilon} n})$, for any $\varepsilon > 0$, unless $NP \subseteq DTIME(n^{polylog(n)})$.*

Algorithm

The algorithm works greedly by iteratively deprecating nodes from the input graph unless a certain condition occurs which makes a node be added to the seed set

Theorem *For any graph $G = (V,E)$ and threshold function $t()$, the algorithm $PA(G,t)$ returns a perfect seed set for G in $O(|E| \log |V|)$ time.*

Theorem *The algorithm $PA(T,t)$ returns an optimal perfect seed set for any* ***tree T*** *and threshold function $t( )$.*

The algorithm works greedly by iteratively deprecating nodes from the input graph unless a certain condition occurs which makes a node be added to the seed set:

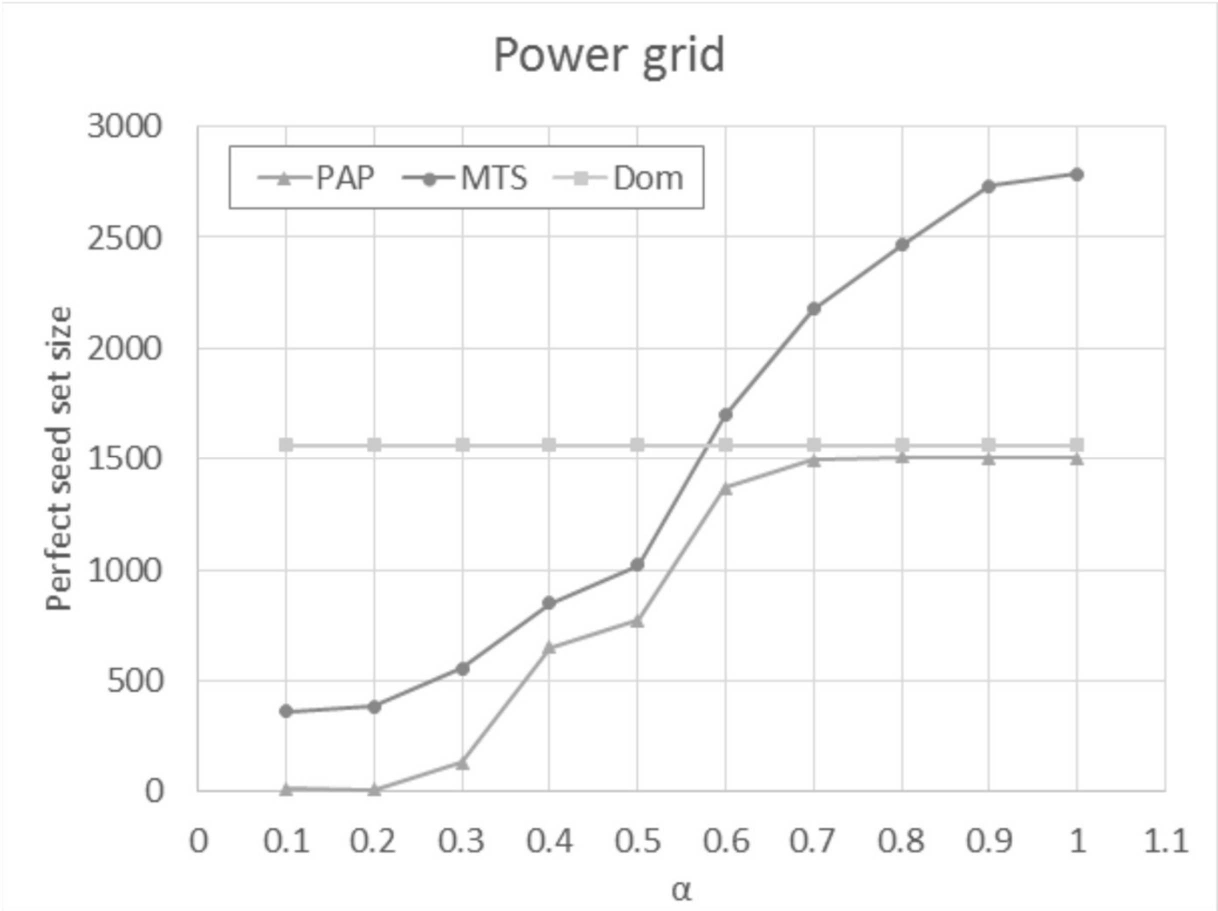
- As long as there exists at least a non-aware node:
  - A node v is opportunely selected and is moved into a *temporary waiting state*;
    - as a consequence, all the neighbors of v will not count on v for being Active
    - at least a neighbour of v must become a spreader
  - Due to this update, some nodes in the surviving graph may remain with a number of *usable* neighbors less that their threshold;
    - in such a case these nodes are added to the seed set and removed from the graph, while the thresholds of their neighbors are decreased by 1 (since they receive influence of the new seeds).
  - If the surviving graph contains a node v whose threshold has been decreased down to 0
    - that is, the nodes which have been already added to the seed set suffice to make v a spreader,
    - v is deleted from the graph and the thresholds of its neighbors are decreased by 1 (since once v becomes a spreader, they will receive its influence).

Algorithm 1: PA(G, t) //$G = (V, E)$ is a graph with thresholds $t(v)$ for $v \in V$

$S = \emptyset$; $Temp = \emptyset$; $U = V$; $R = \emptyset$; $A = \emptyset$;
foreach $v \in V$ **do**
    $k(v) = t(v)$;
    $\delta(v) = |N(v)|$;
while $A \neq V$ ***OR*** $R \neq \emptyset$ **do**
    if $\exists v \in U$ $s.t.$ $k(v) = 0$ **then** // **Case 1)**: v is a spreader, thanks to its neighbors outside U
        foreach $u \in N(v) \cap U$ **do**
            $k(u) = \max(k(u) - 1, 0)$; $A = A \cup \{u\}$;
            if $v \notin Temp$ **then** $\delta(u) = \delta(u) - 1$;
        $U = U - \{v\}$; $R = R - \{v\}$; $A = A \cup \{v\}$;
    else
        if $\exists v \in (U - Temp) \cap R$ $s.t.$ $\delta(v) < k(v)$ ***OR*** $\exists v \notin A$ $s.t.$ $\delta(v) = 0$ **then**
            // **Case 2)**: v must be a seed
            $S = S \cup \{v\}$;
            foreach $u \in N(v) \cap U$ **do**
                $k(u) = k(u) - 1$;
                $\delta(u) = \delta(u) - 1$;
            $U = U - \{v\}$; $R = R - \{v\}$; $A = A \cup \{v\}$;
        else
            if $U - Temp - R \neq \emptyset$ **then** // **Case 3)**: v is moved in the temporary repository
                $v = \mathtt{argmin}_{w \in U - Temp - R}\{\delta(w)\}$
                if $v \notin A$ **then**
                    $R = R \cup \{u\}$ where $u = \mathtt{argmax}_{w \in N(v) \cap (U - Temp)}\{\delta(w)\}$
                    foreach $z \in N(u) \cap U$ **do** $A = A \cup \{z\}$;
            else
                $v = \mathtt{argmax}_{w \in R}\left\{\frac{k(w)}{\delta(w)(\delta(w)+1)}\right\}$;
            foreach $u \in N(v) \cap U$ **do** $\delta(u) = \delta(u) - 1$;
            $Temp = Temp \cup \{v\}$; $R = R - \{v\}$; $A = A \cup \{v\}$;
return S

Esperiments

Experiments show that algorithm PA(G,t) is quite effective in practice



thank you for your attention