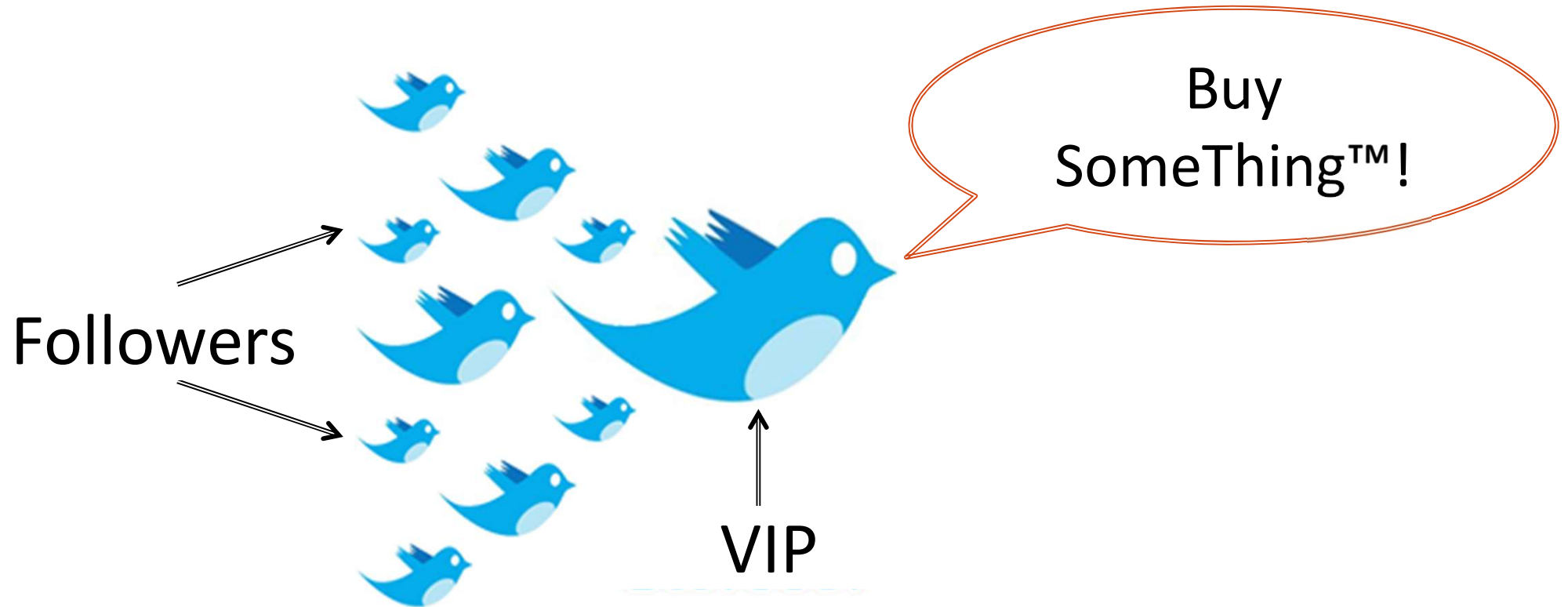


Dynamics in threshold networks

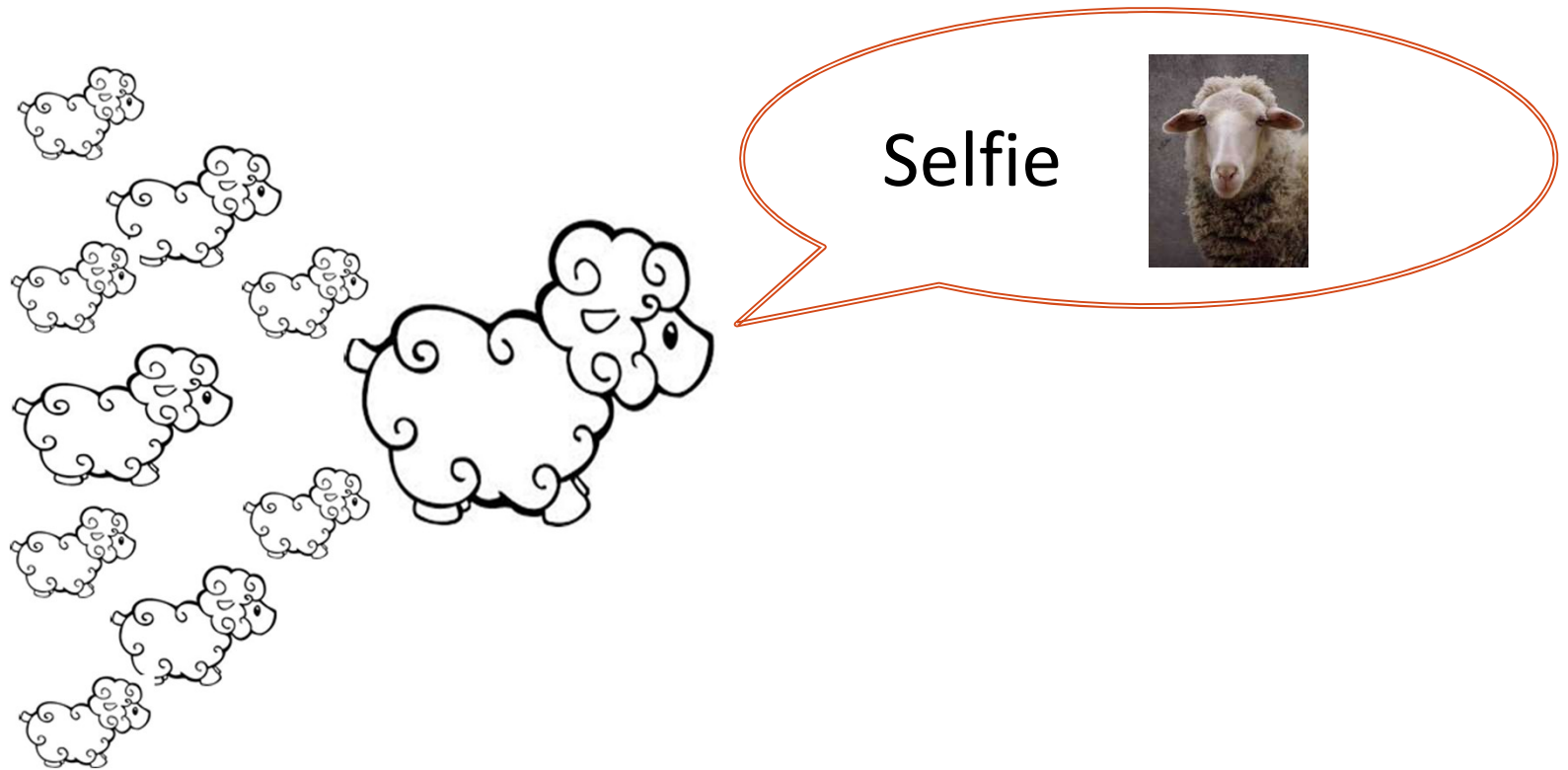
Spread of Influence in Social Networks

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



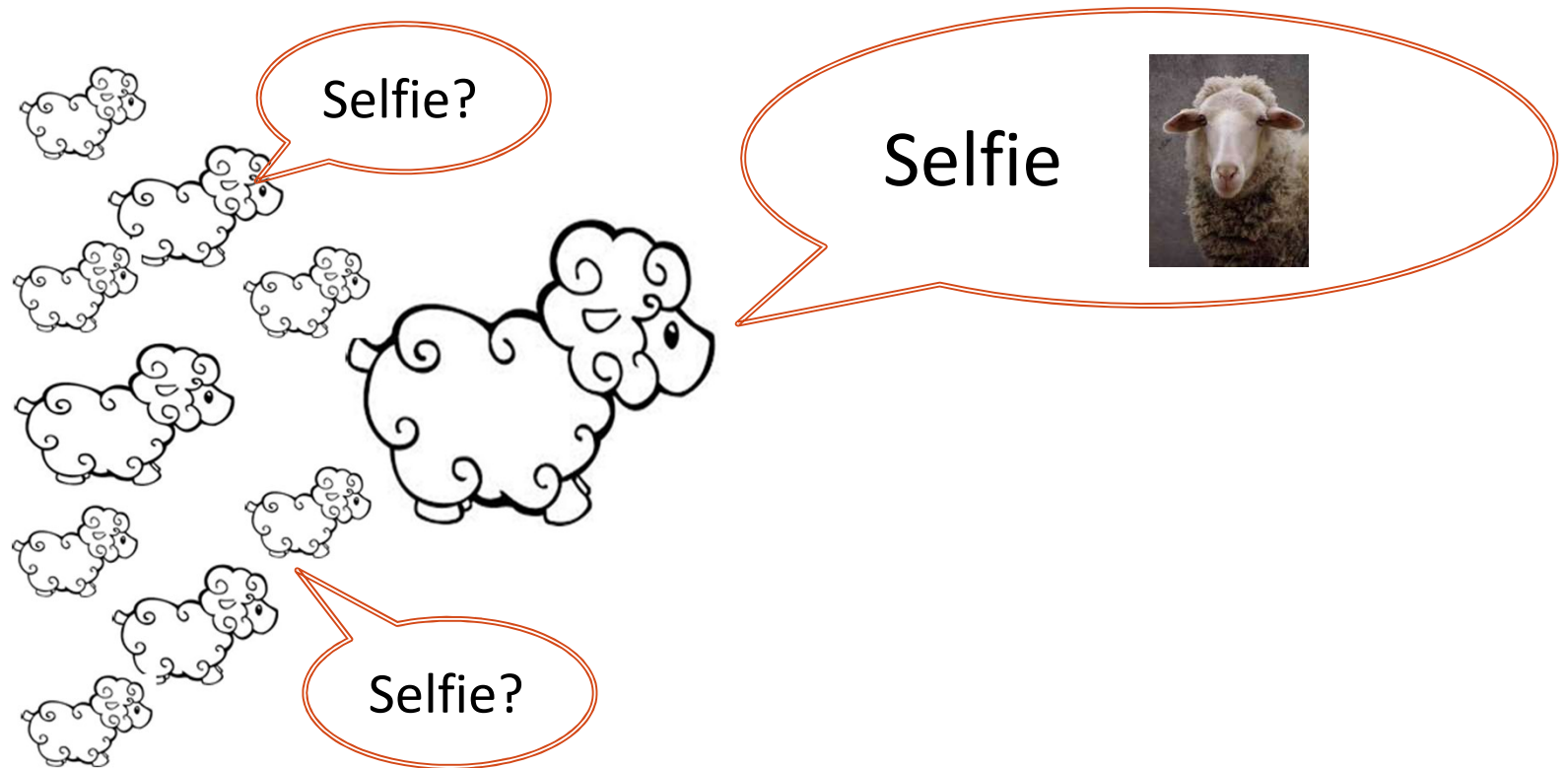
Spread of Influence in Social Networks

As individuals become Active by new ideas/products



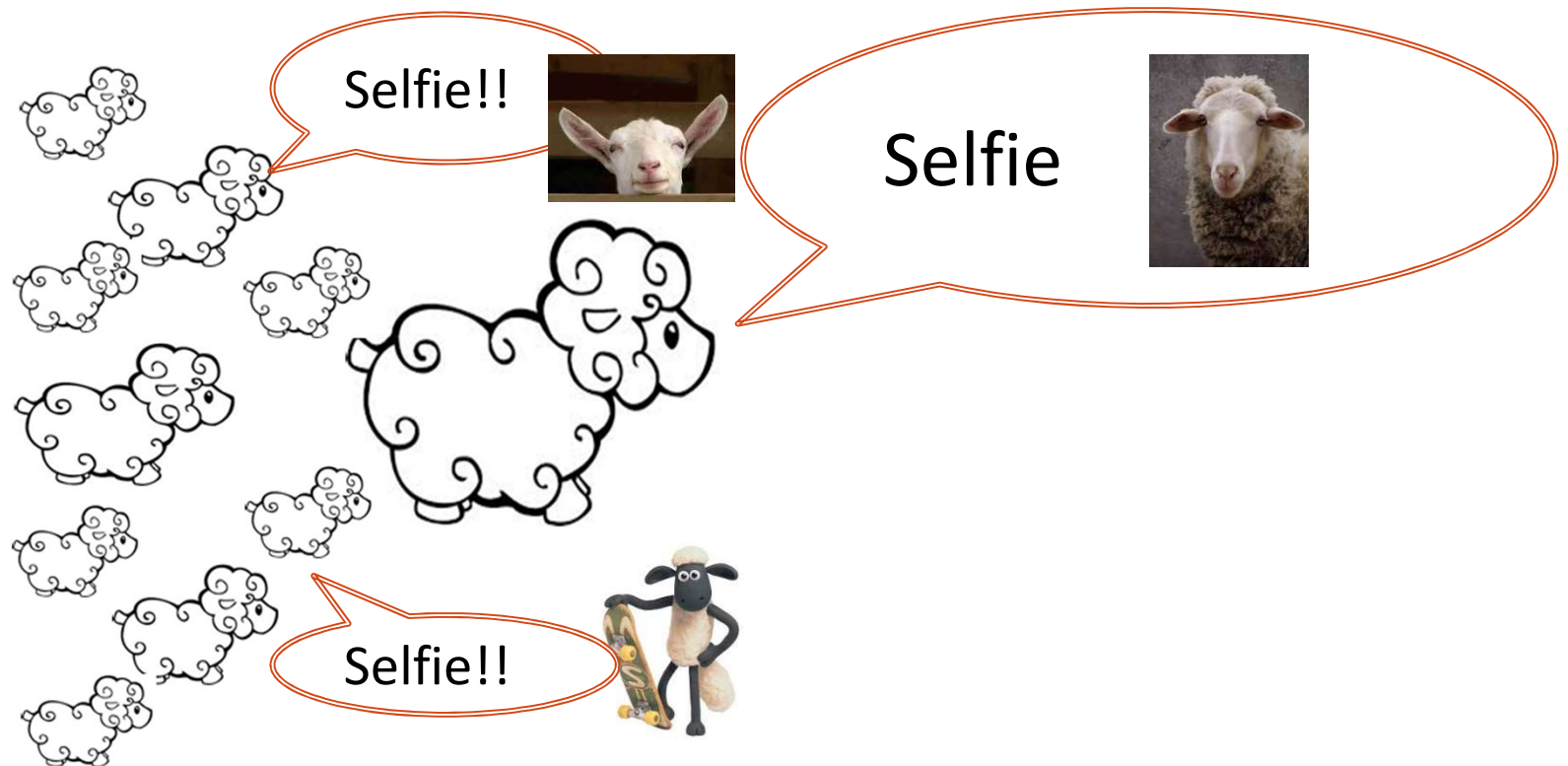
Spread of Influence in Social Networks

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



Spread of Influence in Social Networks

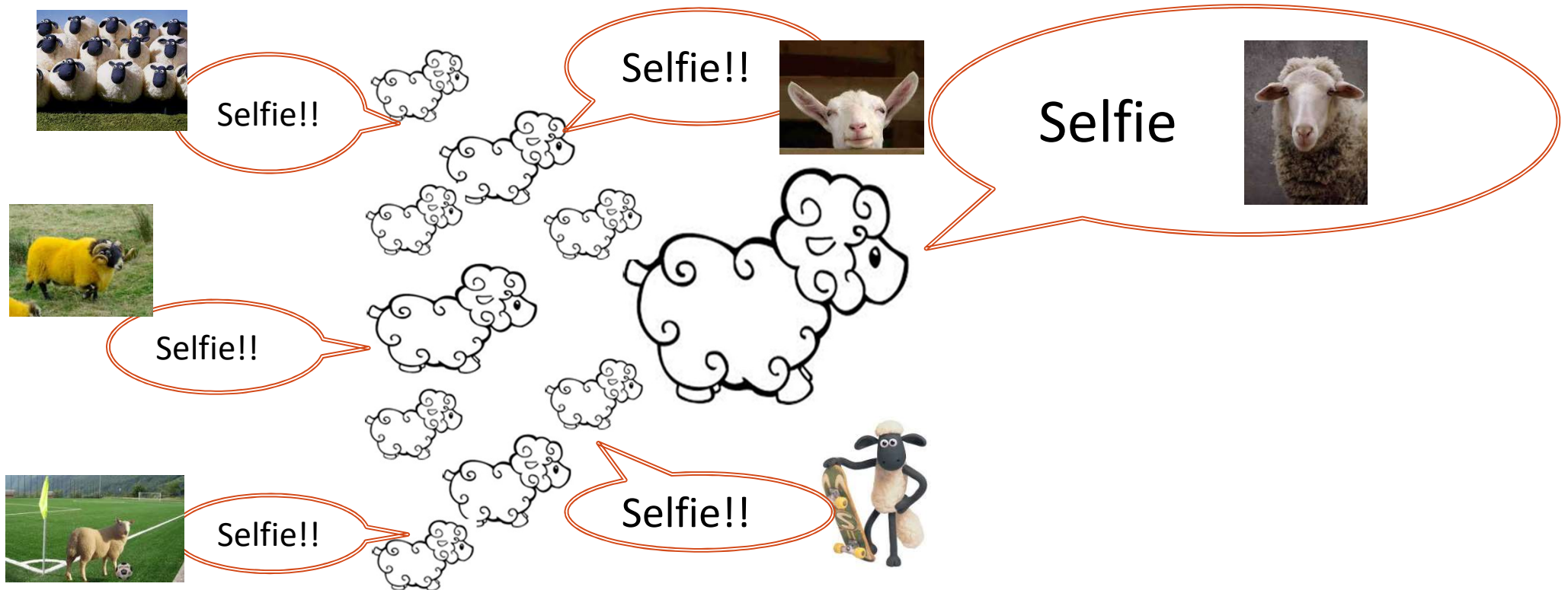
As individuals become aware of new ideas/products they have the potential to pass them to their friends



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

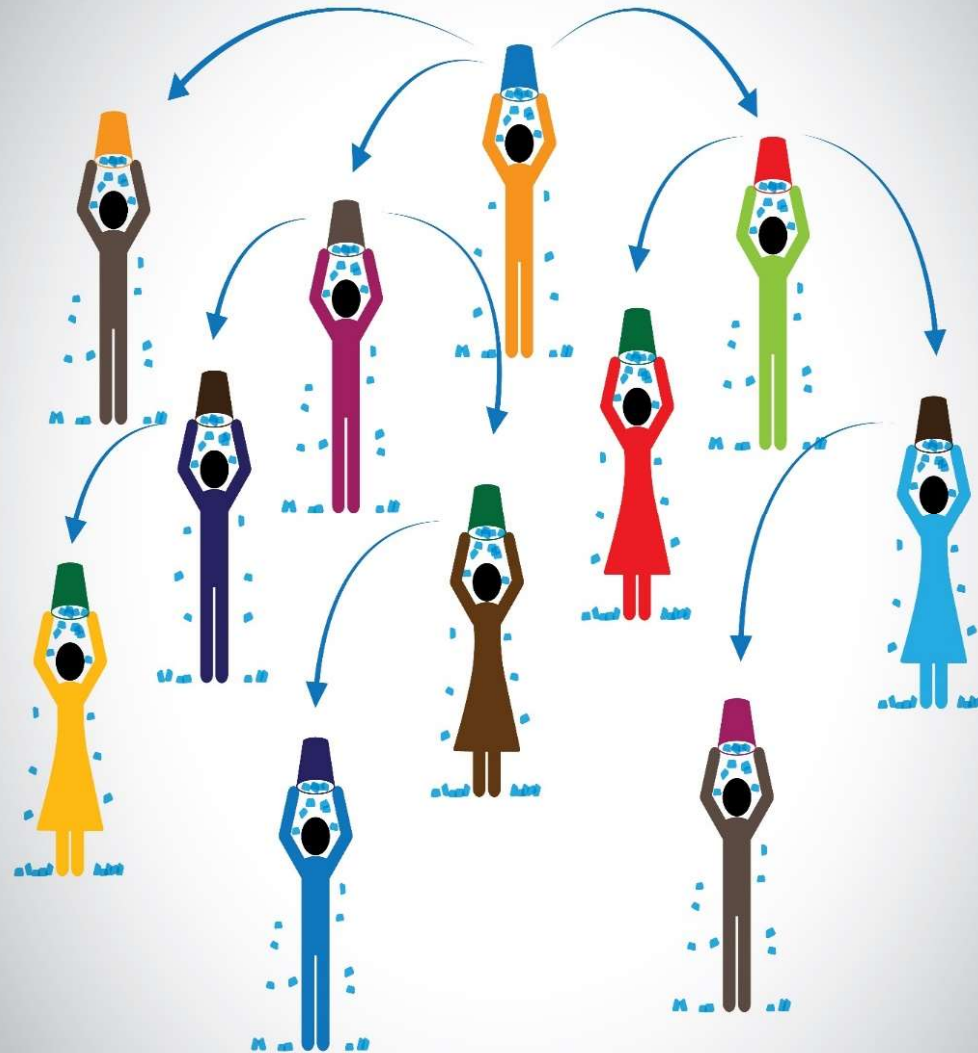
Ice Bucket Challenge

on



Social Confor

- Change belief in group



customers to
ing campaigns.

potential
to use their
fluence, and
ades of



Influence Diffusion

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

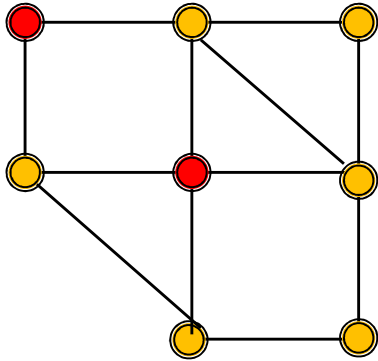
$$\textit{Influenced}[S, 0], \textit{Influenced}[S, 1], \dots, \textit{Influenced}[S, r], \dots \subseteq V$$

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

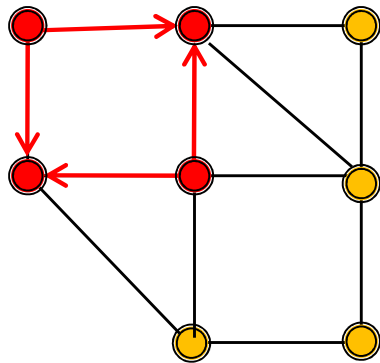
Influence Diffusion:



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

- $Influenced[S, 0] = S$

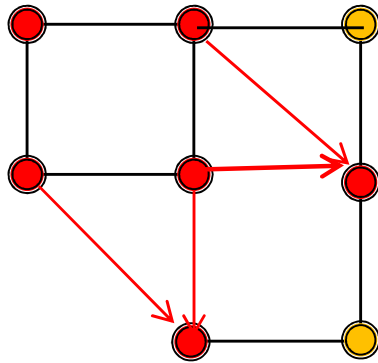
Influence Diffusion



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

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

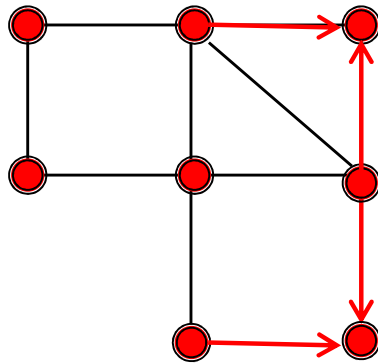
Influence Diffusion



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

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

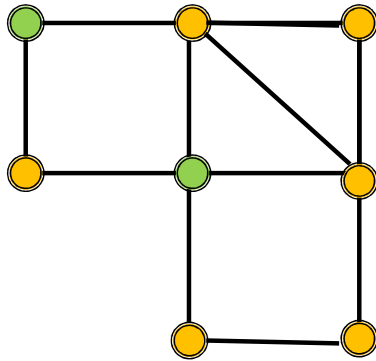
Influence Diffusion



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

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

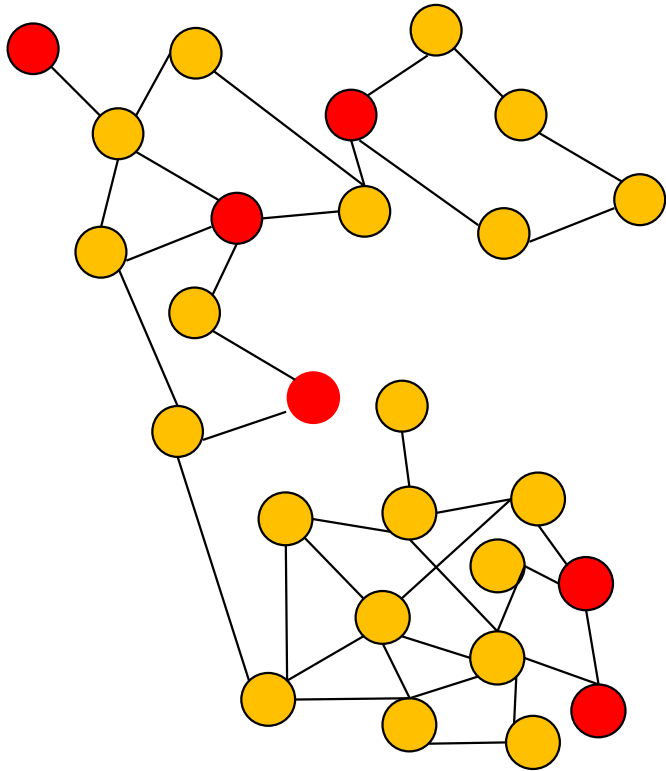
Influence Diffusion



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

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

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 $\text{Influenced}[S, \ell] = V$, for some $\ell \geq 0$.

Target Set Selection

Instance: A network $G = (V, E)$ with thresholds $t : V \rightarrow \mathbb{N}$.

Problem: Find a target set $S \subseteq V$ of minimum size for G .

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

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

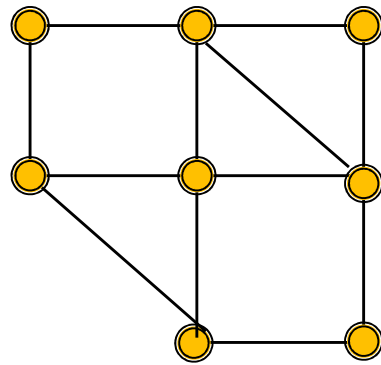
EXACT RESULT

- Hard to approximate within 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,

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)

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



IDEA:

Given the network G

Pick one node a time

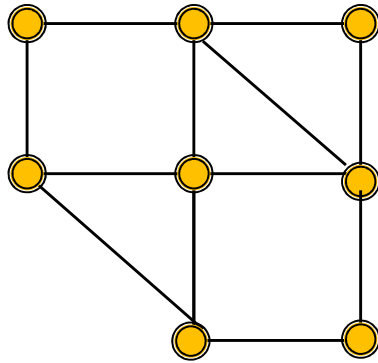
- **decide if it should go in the target set S**
- **eliminate it from G**

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
 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

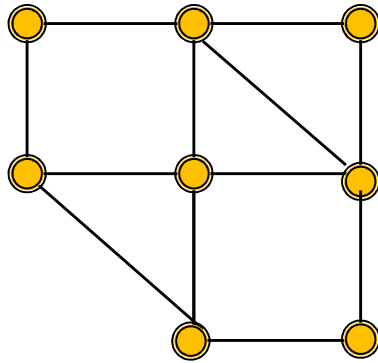
*There is a node v
whose threshold is 0 !
Just eliminate v from the graph*

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
 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

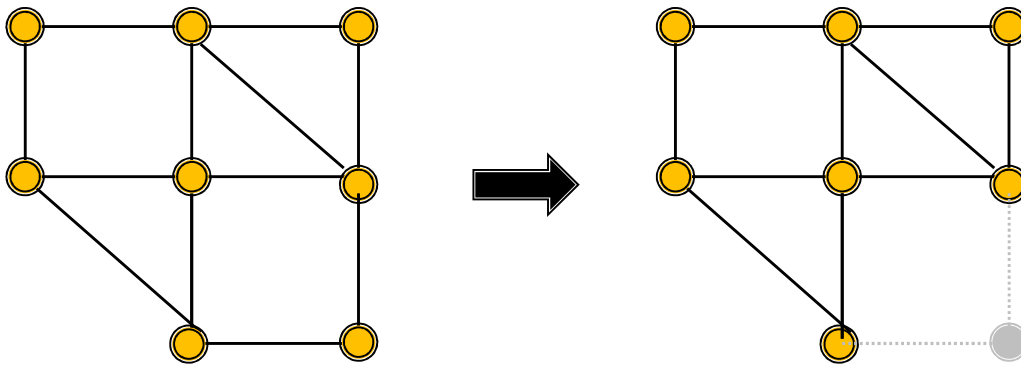
*There is an node v with
threshold larger than the degree!
Add v to S before eliminating from the
graph*

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
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

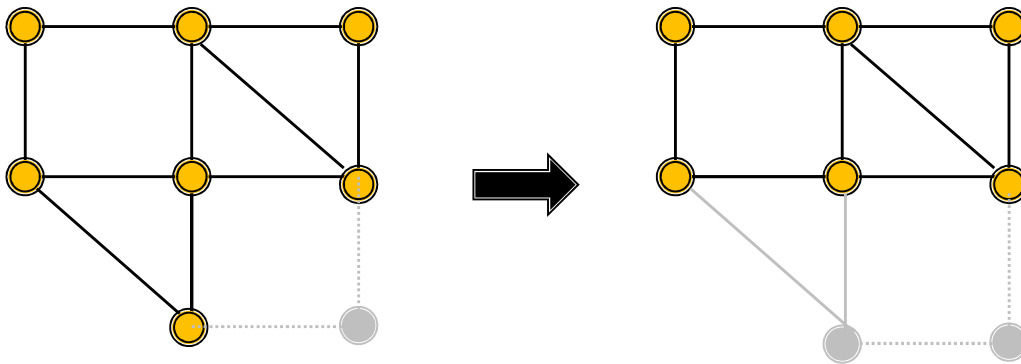
*Otherwise
Pick v with largest
ratio and eliminate it*

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
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

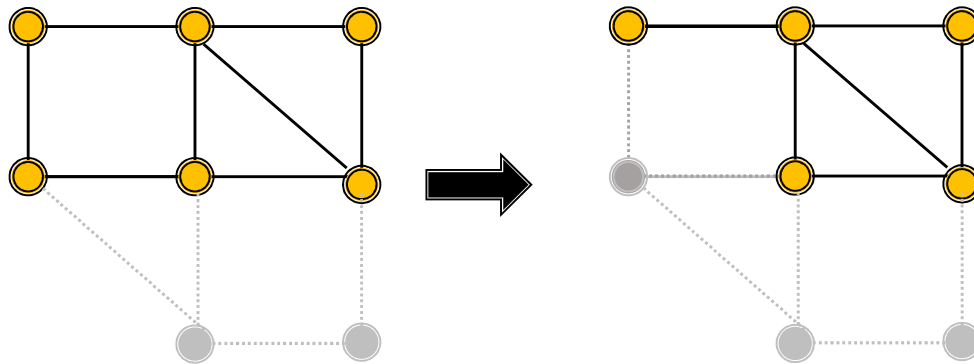
*Otherwise
Pick v with largest
ratio and eliminate it*

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
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

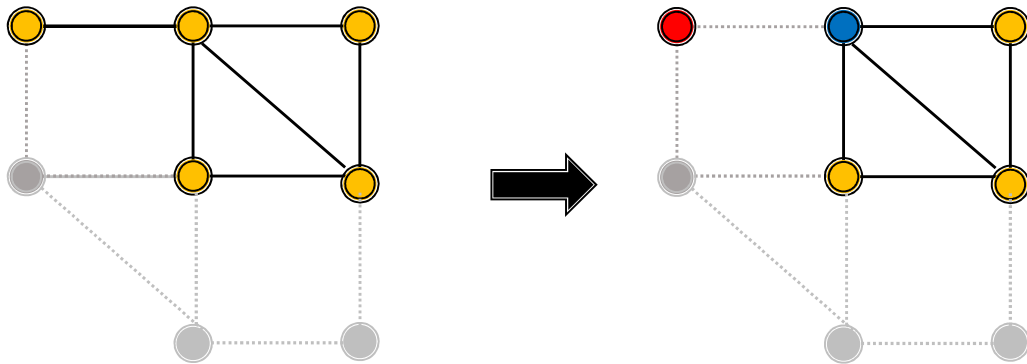
*Otherwise
Pick v with largest
ratio and eliminate it*

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
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!

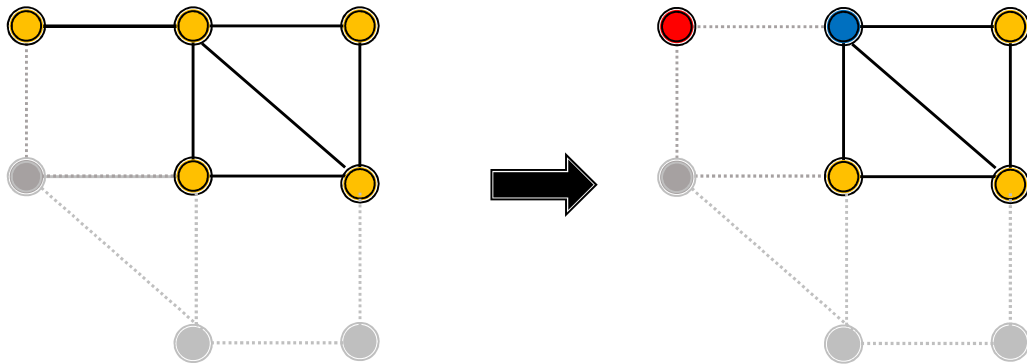
Add v to S before eliminating from the graph

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
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*



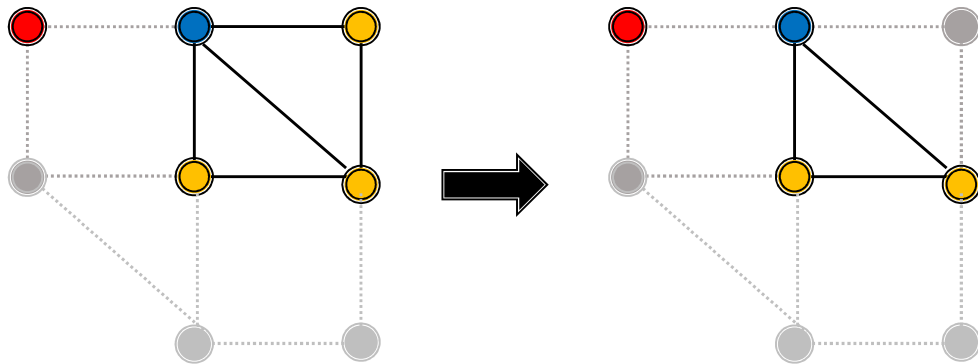
$t(v) = 2$ for each yellow node v ,
 $t(v) = 1$ for blue node

Algorithm TSS(G)

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

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
 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

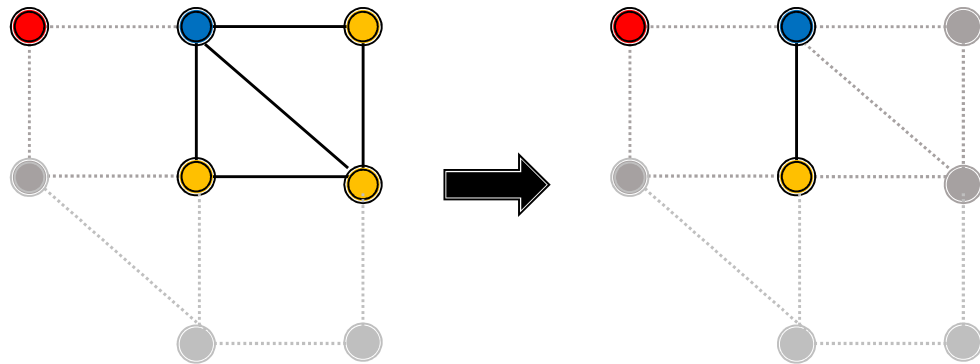
*Pick v with largest
ratio and eliminate it*

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
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

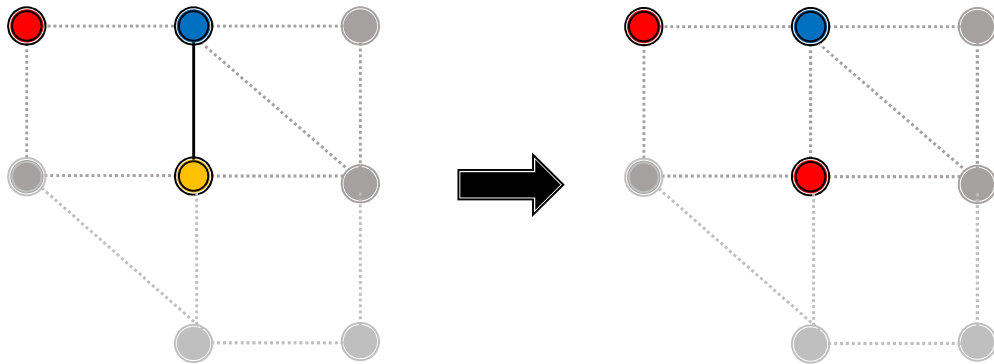
*Pick v with largest
ratio and eliminate it*

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
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

*There is an node v with
 threshold larger than the
 degree!*

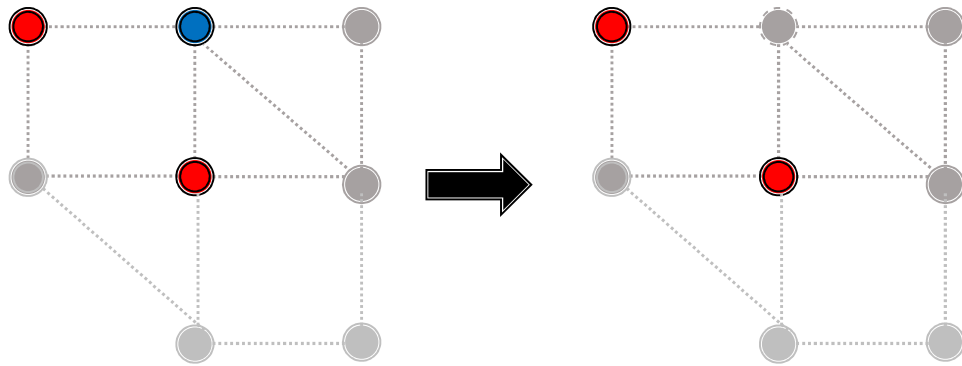
*Add v to S before eliminating
 from the graph*

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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
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**



S = red nodes

$t(v) = 1$ for blue node

*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 $v = \operatorname{argmax}_{u \in U} \left\{ \frac{t(u)}{d(u)(d(u)+1)} \right\}$
 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

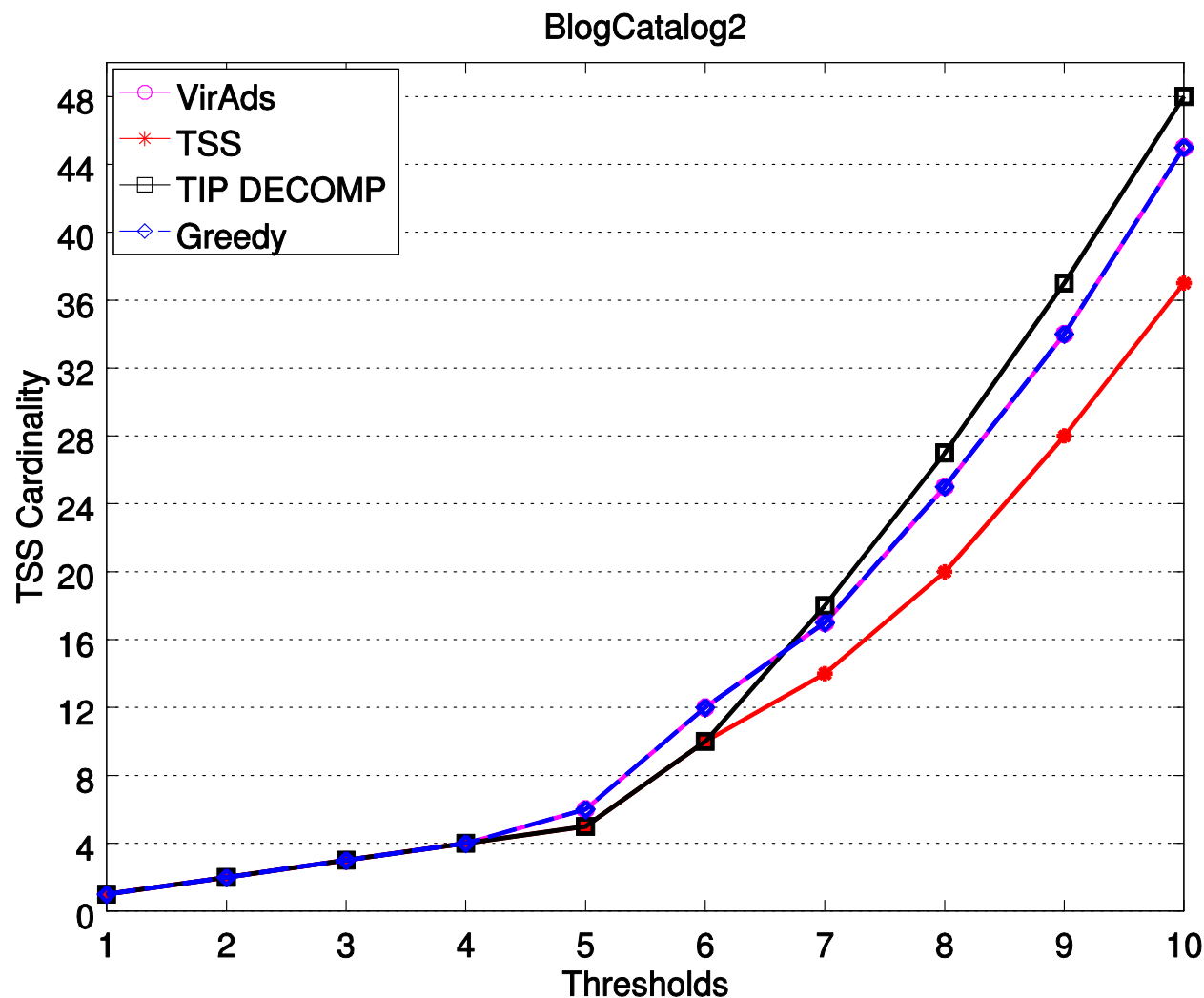
Our algorithm gives a unified proof for several results in the literature, obtained by means of different ad hoc algorithms/techniques

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

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

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 §



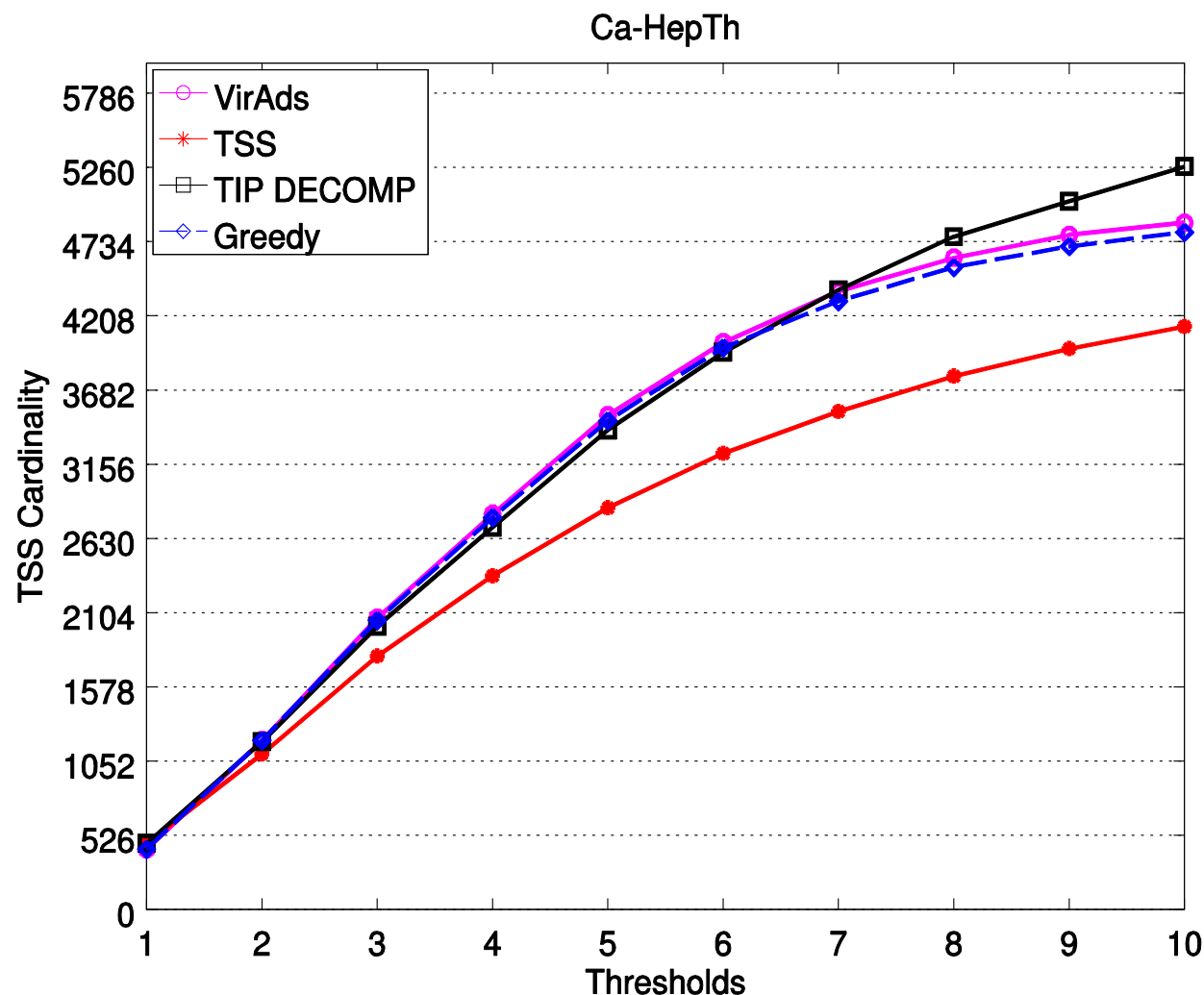
BlogCatalog

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 algorithm** 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 §

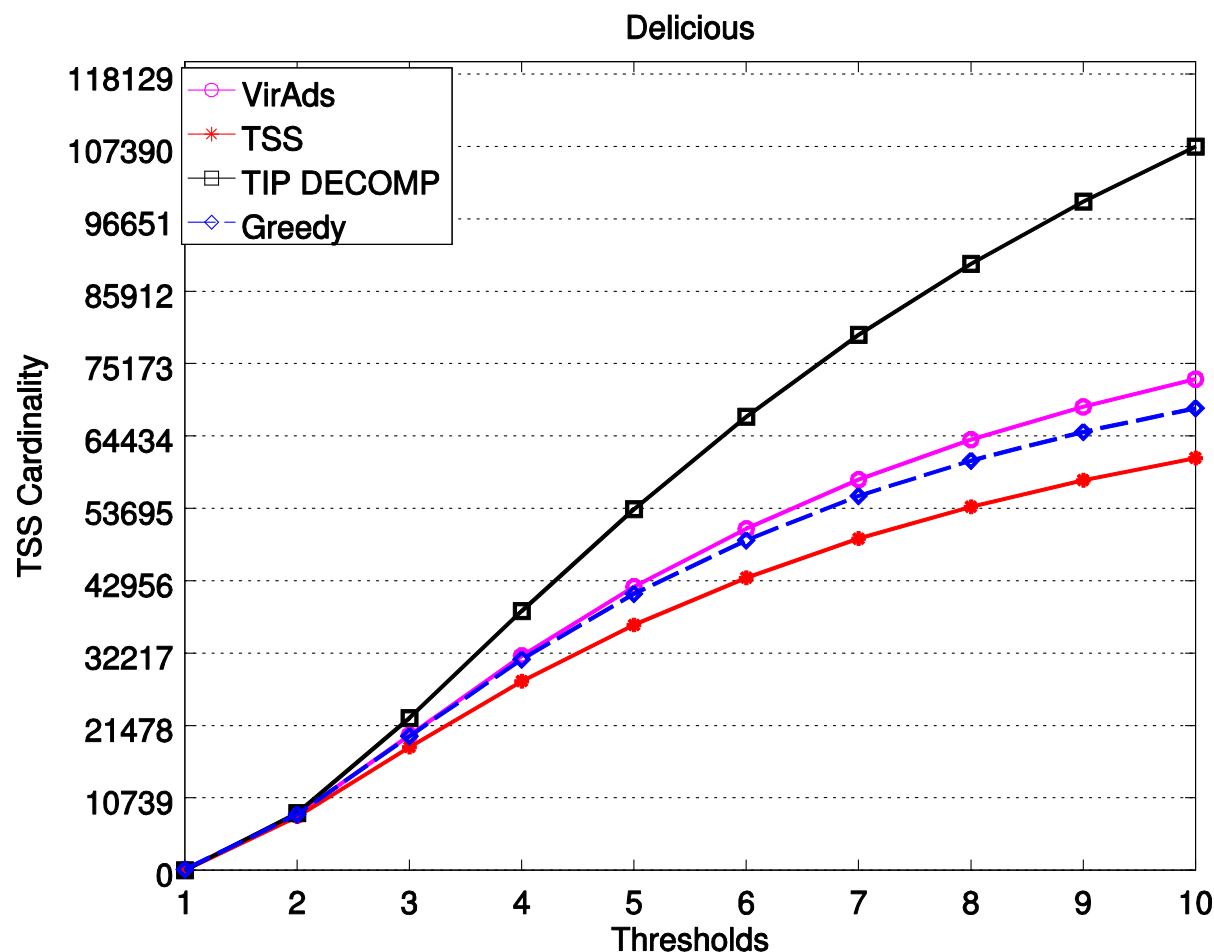


Collaboration
network of Arxiv
papers submitted to
High Energy Physics
– Theory category

§Comparisons of the results of **our algorithm** 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 §



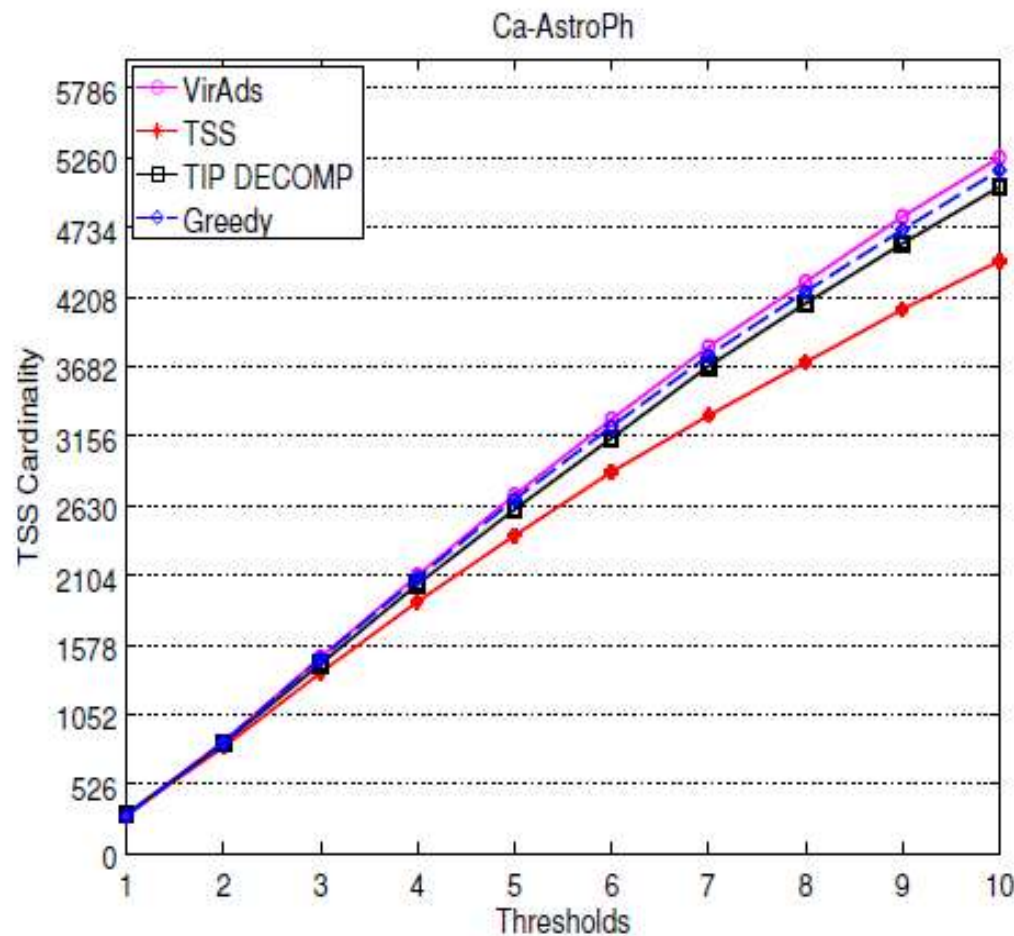
Delicious is a social bookmarking web service for storing, sharing, and discovering web bookmarks.

It has 103.144 nodes and 1.419.519 edges.

§Comparisons of the results of **our algorithm** 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 §



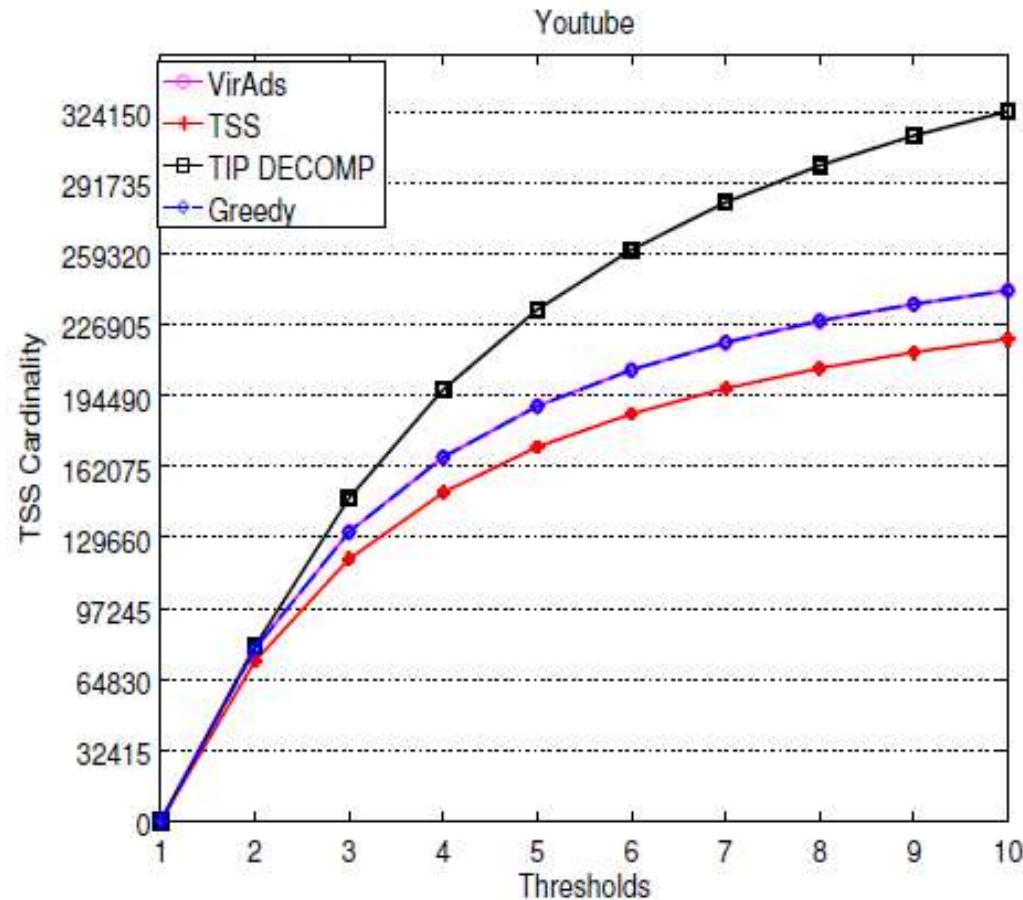
CA-Astro-Ph is a collaboration network of Arxiv ASTRO-PH (Astro Physics).

It has 18.777 nodes and 198.110 edges.

§Comparisons of the results of **our algorithm** 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 algorithm** 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 $\text{Active}[S, \infty] = V$

Targeting with Partial Incentives.

- The classical problem forces the optimizer to make a 0/1 choices 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 $V = \{v_1, \dots, v_n\}$, is a vector $\mathbf{s} = (s(v_1), \dots, s(v_n))$, where $s(v) \in \{0, 1, 2, \dots\}$ 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 $\text{Active}[\mathbf{s}, 0] \subseteq \text{Active}[\mathbf{s}, 1] \subseteq \dots \subseteq \text{Active}[\mathbf{s}, \ell] \subseteq \dots \subseteq V$ 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 \mathbf{s} is an **assignment of partial incentives** that triggers an activation process influencing the whole network, that is, such that $\text{Active}[\mathbf{s}, \ell] = V$ for some $\ell \geq 0$.

Targeting with Partial Incentives

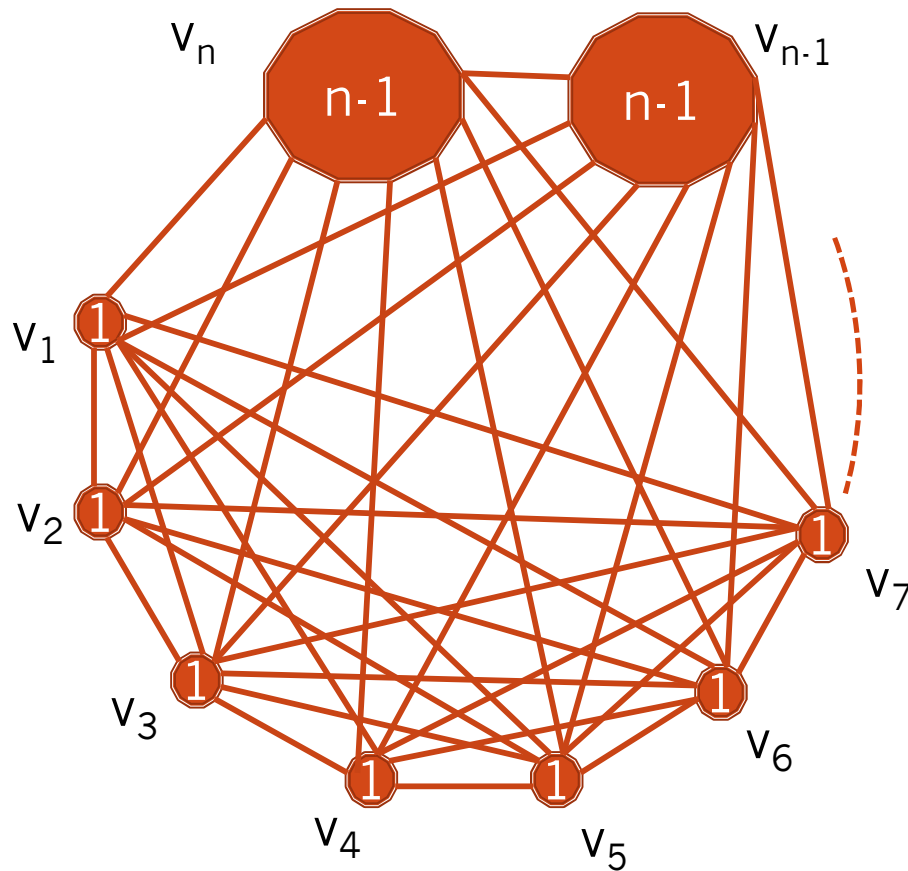
Instance: A network $G = (V, E)$ with thresholds $t : V \rightarrow \mathbb{N}$.

Problem: Find a target vector \mathbf{s} which minimizes

$$C(\mathbf{s}) = \sum_{v \in V} s(v)$$

Find the target vector \mathbf{s} that **minimizes the total incentives** to have $\text{Active}[\mathbf{s}, \infty] = V$

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 $\text{NP} \subseteq \text{DTIME}(n^{\text{polylog}(n)})$.

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 .

1. $U = V$
 2. **for** each $v \in V$ **do**
 3. $s(v) = 0$ [*Partial incentive initially assigned to v*]
 4. $\delta(v) = d_G(v)$
 5. $k(v) = t(v)$
 6. $N(v) = \Gamma_G(v)$
 7. **while** $U \neq \emptyset$ **do**
 8. [*Select one vertex and either update its incentive or remove it from the graph*]
 9. **if** there exists $v \in U$ s.t. $k(v) > \delta(v)$
 10. **then** [*Case 1: Increase $s(v)$ and update $k(v)$*]
 11. $s(v) = s(v) + k(v) - \delta(v)$
 12. $k(v) = \delta(v)$
 13. **if** $k(v) = 0$ **then** $U = U - \{v\}$ [*here $\delta(v) = 0$*]
 14. **else** [*Case 2: Choose a vertex v to eliminate from the graph*]
 15. $v = \operatorname{argmax}_{u \in U} \left\{ \frac{k(u)(k(u)+1)}{\delta(u)(\delta(u)+1)} \right\}$
 16. **for** each $u \in N(v)$ **do** $\{\delta(u) = \delta(u) - 1; N(u) = N(u) - \{v\}\}$
 17. $U = U - \{v\}$
-

TPI Results

- Correctness

- **Theorem**

For any graph G and threshold function t , the algorithm $\text{TPI}(G)$ outputs a target vector for G .

- Efficiency

- **Theorem**

For any $G = (V, E)$, the algorithm $\text{TPI}(G)$ 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

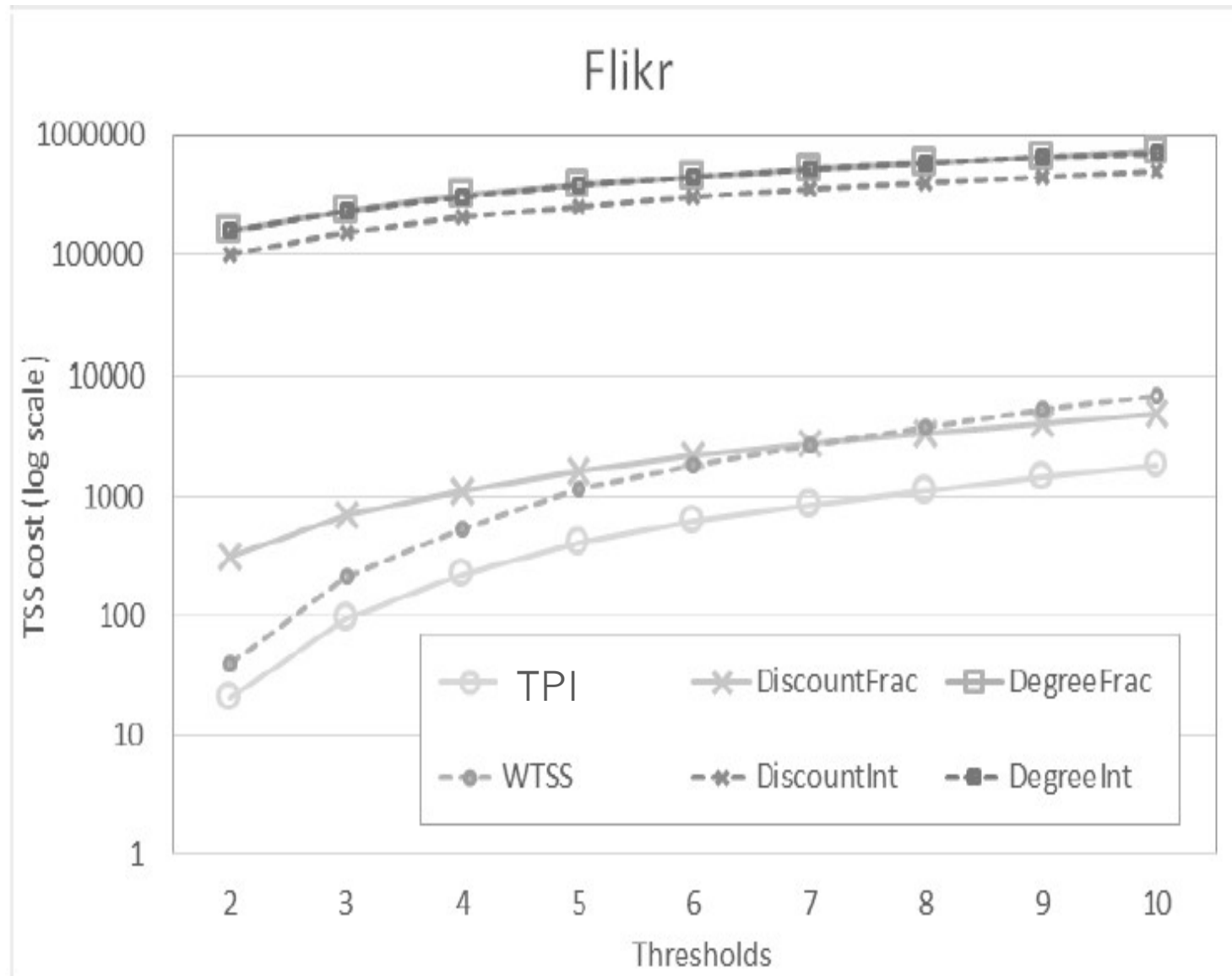
Experiments

- 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%)
Flickr	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: Flickr, Constant Thresholds



Results: Flickr, Degree Proportional Thresholds

