Jointly Complementary&Competitive Influence Maximization with Concurrent Ally-Boosting and Rival-Preventing

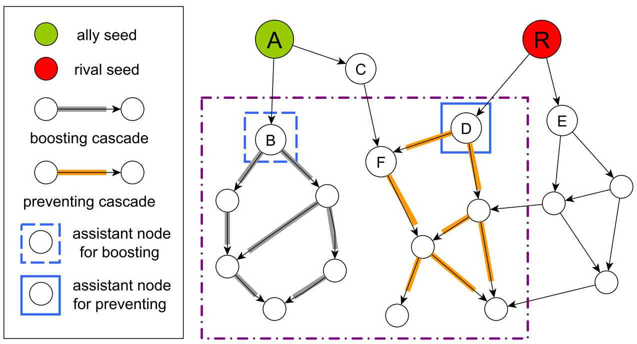

Fig. 1. An example of C^{2}IC problem.

***Abstract*—**

I. Introduction

For example, node B may be an influential user in social network. If B attaches some positive opinions to the piece of ally information that he receives, then a boosting cascade of this information may start. As shown in Fig.1 by the grey bold arrows, the boosted cascade may have higher probability to activate nodes, compared to the original cascade.

Meanwhile, we may also asks another influential user, i.e., node D in the example graph, to attach some negative opinions of the rival information that he receives, then a preventing cascade may start. As shown in Fig.1 by the orange bold arrows, the preventing cascade can protect users from being activated by the rival cascade.

In addition, in competitive environment, the negative opinions against rival cascade are very likely to support the competitive ally cascade. We may assume that the node activated by the preventing cascade still would like to accept the ally cascade.

Thus in the example shown in Fig.1, the expected activation content of the nodes circled by dot-dash lines are considered as the factor that affects the gain of C^{2}IC problem.

II. Influence Spread Model

A. C^{2}IC Model

We define the Complementary&Competitive IC model (C^{2}IC), which combines the influence boosting model proposed in [1], the campaign-oblivious IC model [2] and IC-N (independent cascade model with negative opinions) model [3], while extends to a more complicated scenario.

Specifically, in the C^{2}IC model, there are two cascades, i.e., ally cascade C_a and rival cascade C_r. Our action is to select assistant seed nodes to boost the ally cascade and prevent the rival cascade concurrently.

The two cascades spread in the social network following the classic IC model themselves, if the activation target node is not activated by any other cascade. When both the two cascades try to activate the same node, we assume the rival cascade dominates. That is, when C_r and C_a successfully activate a node at the same time, we assume this node becomes C_r-active. This is the property inherited from the campaign-oblivious IC model.

Given a set of assistant node A, the C^{2}IC model is different from IC model when the cascade spread process meets an assistant node in A.

Ally boosting. If cascade C_a reach an assistant node b, then we denote the subsequent cascade originated from b as C_a^+. Then if cascade C_a^+ reach an un-passed edge (u, v), it successfully spread through this edge with a probability of $p_{uv}^+ \geq p_{uv}$. This is the property extended from the influence boosting model, where the assistant nodes will start a boosted cascade that is enhanced from C_a-cascade by attaching additional positive information.

Rival preventing. If cascade C_r reach an assistant node b, then we denote the subsequent cascade originated from b as C_r^-. Then if cascade C_r^- reach an un-passed edge (u, v), it successfully spread through this edge still with probability p_{uv}. The difference is that, if a node becomes C_r^--active, it would never become C_r-active. This is the property inherited IC-N model, where the assistant nodes will start an anti-spread with negative opinions against C_r cascade, after meeting C_r-cascade.

When a node choose to become X-active, it will not change its state until the spread process ends.

C$^+$IC model: only exist cascade C_a and ally boosting.

It is an out-neighbor version of influence boosting model.

C$^-$IC model: only exist cascade C_r and rival preventing

It is an special case of COIC model.

B. Augmented Realization

realization of IC model is defined as follows.

For each edge $(u, v) \in E$, flip a coin and denote it is "live" with probability of p_{uv}. After all coins flipped, take $g = (V, E')$ as a realization where V is the node set containing all nodes and E' is the edge set containing all "live" edges.

Given a seed set S_a, we know the expected influence spread of S_a, i.e., the expected number of nodes that S_a can activate, is equal to the expected number of nodes that S_a can reach in a random generated realization g. That is, $\sigma(S_a) = \mathbb{E}[H_g(S_a)]$, where $\sigma(S_a)$ is the expected influence spread of S_a and $H_g(S_a)$ is the number of nodes that S_a can reach in g.

now, we define *augmented realization (Aug-realization)* for C^{2}IC model.

Given a realization g, for each edge $(u, v) \in E\backslash E'$, flip a coin and denote it is "live-upon-boost" with probability $\frac{p^+_{uv} - p_{uv}}{1-p_{uv}}$. After all coins flipped, take $g^{aug} = (V, E'')$ as a realization where V is the node set containing all nodes and E'' is the edge set containing all "live" and "live-upon-boost" edges.

Then we can show the expected influence spread of cascades C_a, C_r, C^+_a and C^-_r.

Denote $d_g(u, v)$ as the length of shortest path from u to v in realization g.

Since the ally seeds and rival seeds are given, we only use assistant nodes as input variable when defining objective function. We define $\sigma_a(S)$, $\sigma_r(S)$ as the expected number of nodes activated by cascade C_a and C_r respectively. Note C^+_a-active nodes are also regarded as activated by C_a.

Now we define five types of directed acyclic paths for end node v in an aug-realization.

C_a path ρ_a: starting at an ally seed and end at v with all edges in path ρ_a all being live edges.

C_r path ρ_r: starting at a rival seed, passing no assistant node and end at v with all edges in path ρ_r all being live edges.

C_a-boost-potential path $\rho^+_a p$: starting at an ally node, passing as least one assistant and end at v, with all edges in the path being live-upon-boost or live.

C_r-prevention path ρ^-_r: starting at an assistant node, passing as least one assistant and end at v, with all edges in the path being live.

Define $d^\rho_g(u, v)$ as the distance of the path ρ from u to v on aug-realization g. Then we can show the reachability.

C_a-boost path ρ^+_a: a C_a-boost-potential path $\rho^+_a p$ that satisfies for the ith node in the path u_i, $i-1 <= d^{\rho_a}_g(u_a, u_i)$, $\forall u_a \in S_a$ and $i-1 < d^{\rho_r}_g(u_r, u_i)$, $\forall u_r \in S_r$.

Priority Setting.

The reachability is highly related to the priority settings.

$C_a \succ C_r$ means C_a has a higher priority than C_r, which means when both C_a and C_r successfully reach an inactive node at the same time, this node will choose to become C_a-active.

Different priority settings will affect the influence spread process heavily. We will analyze different properties of the objective functions under different priority settings in Section 8.

In this paper, we consider 4 cases of the objective function:

M-S: monotone and submodular

M-nS: monotone and not submodular

nM-S: not monotone and submodular

nM-nS: neither monotone nor submodular

A case of M-S. By setting $C^+_a \succ C_a \succ C^-_r \succ C_r$, we can have the following results.

$H^r_g(S)$: the set of nodes C_r-active in g. Each node v in $H^r_g(S)$ satisfies all the inequalities Hr1, Hr2 and Hr3:

Hr1: $\exists u_r \in S_r$, $d^{\rho_r}_g(u_r, v) < d^{\rho_a}_g(u_a, v)$, $\forall u_a \in S_a$

Hr2: $\exists u_r \in S_r$, $d^{\rho_r}_g(u_r, v) < d^{\rho^-_r}_g(u^-_r, v)$, $\forall u^-_r \in S_r$

Hr3: $\exists u_r \in S_r$, $d^{\rho_r}_g(u_r, v) < d^{\rho^+_a}_g(u^+_a, v)$, $\forall u^+_a \in S_a$

$H^a_g(S)$: the set of nodes C_a-active or C^+_a-active in g. Each node v in $H^a_g(S)$ satisfies either the inequalities Ha1 and Ha2:

Ha1: $\exists u_a \in S_a$, $d^{\rho_a}_g(u_a, v) < d^{\rho_r}_g(u_r, v)$, $\forall u_r \in S_r$

Ha2: $\exists u_a \in S_a$, $d^{\rho_a}_g(u_a, v) < d^{\rho^-_r}_g(u^-_r, v)$, $\forall u^-_r \in S_r$

or the inequalities Ha+1 and Ha+2:

Ha+1: $\exists u^+_a \in S_a$, $d^{\rho_a}_g(u^+_a, v) < d^{\rho_r}_g(u_r, v)$, $\forall u_r \in S_r$

Ha+2: $\exists u^+_a \in S_a$, $d^{\rho_a}_g(u^+_a, v) < d^{\rho^-_r}_g(u^-_r, v)$, $\forall u^-_r \in S_r$

Then we have $\sigma_r(S) = \mathbb{E}[H^r_g(S)]$ and $\sigma_a(S) = \mathbb{E}[H^a_g(S)]$, where the expectations are taken from the randomness of g.

Under other cases of priority, we just need to change the distance compare conditions of whether to take equal sign.

III. Problem Definition

Jointly complementary and competitive influence maximization problem.

$f(S) = \sigma_a(S) - \sigma_a(\emptyset)$

$g(S) = \sigma_r(\emptyset) - \sigma_r(S)$

Let $h(S) = \lambda f(S) + (1-\lambda) g(S)$

Problem 1 (C^{2}IM): Given an ally seed set S_a and a rival seed set S_r, the C^{2}IM problem asks to select an assistant seed node set A, such that the weighted summation of the expected increased number of ally-adopted nodes and the expected reduced number of rival-adopted nodes, is maximized, i.e.,

$$S^* = \arg \max_{S \subseteq V \backslash S_a \backslash S_r} h(S).$$

Naturally, we can obtain two simplified versions of C^{2}IM problem by considering C$^+$IC model and C$^-$IC model as follows.

Problem 2 (C$^+$IM): Given an ally seed set S_a, the C$^+$IM problem asks to select an assistant seed node set S, such that the expected increased number of ally-adopted nodes is maximized, i.e.,

$$S^* = \arg \max_{S \subseteq V \backslash S_a} f(S).$$

Problem 3 (C$^-$IM): Given a rival seed set S_r, the C$^-$IM problem asks to select an assistant seed node set S, such that the expected reduced number of rival-adopted nodes, is maximized, i.e.,

$$S^* = \arg \max_{S \subseteq V \backslash S_r} g(S).$$

Theorem 1: All the three problems defined above are NP-hard..

Proof: Consider a well-known NP-hard problem of Maximum k-Coverage [4]. A Maximum k-Coverage instance consists of an integer k and a collection of m non-empty sets $C = \{c_1, c_2, ..., c_m\}$ with each C_i contains a set of elements

$\{e_j\}$. It aims to find a subset $C' \subseteq C$ such that $|C'| \leq k$ and $|\cup_{c_i \in C'} c_i|$ is maximized. In the following, we will show that any Maximum k-Coverage instance can be reduced to a C$^+$IM/C$^-$IM/C^{2}IM instance.

Let $\cup_{c_i \in C} c_i$ be the ground set. For each element $e_j \in G$, we identify it as a node and add it into node set V. Then for each set $c_i \in C$ that contains e_j, we also identify c_i as a node add it into node set V. Meanwhile, we add an edge (c_i, e_j) into edge set E. Now we obtain a graph $G = (V, E)$. Then we further add a node u, and connect u with each node c_i. Let E' be the set containing all edges out-going from u. Now we obtain a graph $G' = (\{u\} \cup V, E \cup E')$. Obviously, the above process can be performed in polynomial time.

For C$^+$IM problem, let all edges in E have a probability of 0, but boosted probability of 1. Let all edges in E' have a probability of 1. Then let node u be the ally seed, we can see the optimal solution of C$^+$IM problem in this instance, must be the optimal solution for the corresponding Maximum k-Coverage problem.

For C$^-$IM problem, let all edges have a probability of 1. Then let node u be the rival seed, we can see the optimal solution of C$^-$IM problem in this instance, must be the optimal solution for the corresponding Maximum k-Coverage problem.

For C^{2}IM problem, we further add a node v, and connect v with each node c_i. Let E'' be the set containing all edges out-going from v. Now we obtain a graph $G'' = (\{u\} \cup \{v\} \cup V, E \cup E' \cup E'')$. Let u be the ally seed and v be the rival seed. Then we can show two special cases of the reduction when $\lambda = 1$ and $\lambda = 0$ respectively.

First, let all edges in E have a probability of 0, but boosted probability of 1. Let all edges in E' and E'' have a probability of 1. Then by setting $\lambda = 1$, we can see the optimal solution of C^{2}IM problem in this instance, must be the optimal solution for the corresponding Maximum k-Coverage problem.

Second, let all edges in E' have a probability of 0, while all edges in E and E'' have a probability of 1. Then by setting $\lambda = 0$, we can see the optimal solution of C^{2}IM problem in this instance, must also be the optimal solution for the corresponding Maximum k-Coverage problem. ■

***Theorem** 2:* Computing $f(\cdot)$ under C$^+$IC model is #P-hard, so is computing $g(\cdot)$ under C$^-$IC model.

Proof: We can reduce the computing of $f(\cdot)$ and $g(\cdot)$ to the computing of influence spread, which is proved to be to be #P-hard. Given a graph G and a node s, we know computing the influence spread of s is #P-hard.

Now we add a node u and a node v into the graph. Meanwhile, we add two edges (u, v) and (v, s) into the graph. Let the probability of (u, v) and (v, s) be p_{uv} and p_{vs} respectively. Further, let the boosted probability of all edges except (v, s) be the same as the spread probability. Further let the boosted probability of (v, s) be $p^+_{vs} > p_{vs}$. Then we can see, if u is the ally seed, the value of $f(\{v\})$, is equal to the influence spread of s multiplied by $p^+_{vs} - p_{vs}$.

Still consider the graph adding node u, v and edge (u, v), (v, s). Let edge (u, v) and (v, s) both have a probability of 1. Let u be the rival seed. Then the value of $g(\{v\})$, is equal to the influence spread of s added by 1. ■

By similar reduction, we can see the objective function computation of C^{2}IM problem is also #P-hard when $\lambda = 1$ or $\lambda = 0$. Thus we may conclude that the computation of $f(\cdot)$ and $g(\cdot)$ under C^{2}IC model at least suffer the same hardness.

IV. Problem Solution for M-S Case

Use the case $C^+_a \succ C_a \succ C^-_r \succ C_r$ as an example. As for other cases, we just need to change the distance compare conditions of whether to take equal sign in the algorithm.

A. Partial Realization Sketch

***Definition** 1 (PR-sketch):* Given an ally set S_a, a rival set S_r, a root node v and a random generated augmented realization g^{aug}, a PR-sketch for v is subgraph $R = (V^R, E^R)$, where V^R contains all nodes that can reach v in g^{aug} and E^R is the union of all the edges in the pathes connecting nodes in V^R and v.

Given an ally set S_a, a rival set S_r and a root node v, a PR-sketch for v is generated by a BFS process starting from v, with edge edge (u, v) visited and remarked as "live" with probability p_{uv}, or "live-upon-boost" with probability of $p^+_{uv} - p_{uv}$. With probability $1 - p^+_{uv}$, we do not visit this edge. When no new node is visited, we end the BFS process and take all the nodes and edges visited to form a subgraph, which is the PR-sketch for v.

We can compute the gain by enumerating the following cases:

(1) Meet no ally or rival in the graph, selecting any node appeared in this PR-sketch does not incur any gain.

(2) In the "live“ and "live-upon-boost“ graph, meet an ally through a C_a-boost path ρ^+_a before any rival In the "live“ graph: selecting any node appeared in this PR-sketch does not incur any gain.

(3) In the "live“ graph, meet an ally before any rival or C_a-boost path ρ^+_a: selecting any node appeared in this PR-sketch does not incur any gain.

(4) In the "live“ graph, meet an rival through an assistant node before any ally or rival without any assistant node or C_a-boost path ρ^+_a: selecting the nodes that can make an ally reach v through "live“ and "live-upon-boost" graph, gain λ.

(5) In the "live“ graph, meet an rival without any assistant node before any ally or rival through assistant node or C_a-boost path ρ^+_a: selecting the nodes that can make an rival reach v through an assistant node, gain $1 - \lambda$; or selecting the nodes that can make an ally reach v through "live“ and "live-upon-boost" graph, gain 1;

Suppose we randomly generate a PR-sketch R for a root node randomly and uniformly selected from V. Define $gain(S, R)$ as the gain of selecting assistant node set S on PR-sketch R. We have that

$$\mathbb{E}[gain(S, R)] = \lambda f(S) + (1 - \lambda) g(S).$$

Thus we can estimate the value of objective function for any set S by sampling PR-sketches.

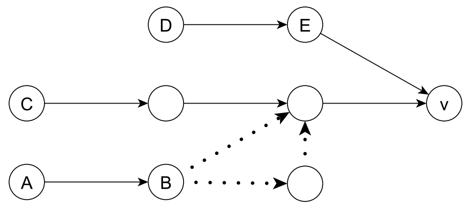

Fig. 2. An example of PR-sketch

We can list the gain of different cases in example shown in Fig.2.

1. A is ally seed, no rival seed, select B, gain is λ.
2. D is ally seed, no rival seed, select any node, no gain.
3. D is rival seed, no ally seed or C is ally seed, select E, gain is $1-\lambda$.
4. C is rival seed, A is ally seed, select B, gain is 1.

When executing the greedy selection, case 6 only appears after the appearance case 1 or case 3. Thus we need to update the gain of nodes in the PR-sketch when case 1 or case 3 appear.

In Algorithm 1 and Algorithm 2, we show the generation process of a random PR-sketch for root node v, and the gain computation for each node appear in the generated PR-sketch.

Algorithm 1: PR-sketch-Generation(G, S_a, S_r, v)

Initialize a queue Q with only v in Q;
while *Q is not empty* **do**
  $t = Q.front()$;
  $Q.pop\_front()$;
  for *each incoming neighbour s of t* **do**
    $p = random(0, 1)$;
    if $p > p_{st}^{+}$ **then**
      continue;
    else
      if $p \le p_{st}$ **then**
        Remark (s, t) as "live";
      else if $p_{st} < p \le p_{st}^{+}$ **then**
        Remark (s, t) as "live-upon-boost";
      if $s \notin S_r \cup S_a$ **then**
        $Q.push\_back(s)$;
return $R = (V^R, E^R)$

Given a PR-sketch R, we define to distances notion as follows:

$d_1^R(u, v)$: the path length from u to v only through live edges in R.

$d_2^R(u, v)$: the path length from u to v only through live and live-upon-boost edges that satisfies C_a-boost path in R.

B. PR-IMM Algorithm

Suppose we have already sampled a large number of PR-sketches $\mathcal{R} = \{R_i\}, i \in [\theta]$. We then can select assistant nodes greedily. It actually follows the approximation solution framework of maximum coverage problem.

Algorithm 2: GainComputation(S_a, S_r, R)

Initialize $\psi(u, R) := 0, \forall u \in R$;
Let v be the root node of R;
Compute $d_1^R(\cdot)$ and $d_2^R(\cdot)$ for each node pair in R;
$d_r = \min\{|d_1^R(r, v)| : r \in S_r\}$;
$d_a = \min\{|d_1^R(a, v)| : a \in S_a\}$;
$d_a^+ = \min\{|d_2^R(a, v)| : a \in S_a\}$;
if *($d_a < d_r$ and $d_a < \infty$) or ($d_r = \infty$ and $d_a = \infty$ and $d_a^+ = \infty$)* **then**
  Return $\{\psi(u, R)\}_{u \in V}$;
for *each node $u \in R$* **do**
  if $d_r = \infty$ *and* $d_a = \infty$ *and* $d_a^+ < \infty$ **then**
    if $\exists a \in S_a, u \in d_2^R(a, v)$ **then**
      $\psi(u, R) := \psi(u, R) + \lambda$;
  else if $d_r < \infty$ *and* $d_r \le d^a$ *and* $d_a^+ < d_r$ **then**
    if $\exists a \in s_a, u \in d_2^R(a, v) and |d_2^R(a, v)| < d_r$ **then**
      $\psi(u, R) := \psi(u, R) + 1$;
  else if $d_r < \infty$ *and* $d_r \le d^a$ **then**
    if $\exists r \in s_r, |d_1^R(r, u)| + |d_1^R(u, v)| = d_r$ **then**
      $\psi(u, R) := \psi(u, R) + 1 - \lambda$;
return $\{\psi(u, R)\}_{u \in V}$

That is, select the node that brings largest marginal gain of $\Phi(S) = \sum_{R \in \mathcal{R}} gain(A \cup \{u\}, R)$. Note adding new node, may reduce the gain of unselected nodes, which is so-called submodular, of diminishing return property. We can show that the monotonicity and submodularity of function $\Phi(S)$ is consistent to function $h(S)$. In this section, we are considering the case that $h(S)$, accordingly $\Phi(S)$, is monotone and submodular. Hence a $(1-1/e)$-approximation ratio can be achieved by the greedy algorithm.

We utilize the sampling scheme proposed in the IMM method [5], which can ensure a $1 - 1/e - \varepsilon$ approximate solution with at least $1 - \frac{1}{n^\ell}$ probability.

Generally, in the PR-IMM algorithm (Algorithm 3), the SampleGeneration in Line 2 (Algorithm 4) is to sample a large enough PR-sketch set $\mathcal{R}$ and the GreedySelection in Line 3 (Algorithm 5) is to select a set S using $\mathcal{R}$. Let OPT be the maximum value of the objective function obtained by optimal solution S^*. The parameters in Algorithm 4 are defined as

$$\alpha = \delta \sqrt{\ell \log n + \log 2}$$

$$\beta = \sqrt{\delta \left(\ell \log n + \log \binom{n}{k} + \log 2\right)}$$

$$\theta_i = \left(1 + \frac{\sqrt{2}}{3}\varepsilon\right) \cdot \left(\log \binom{n}{k} + \ell \log n + \log \log_2 n\right) \cdot \frac{2^i}{\varepsilon^2}$$

By the analysis of the IMM method [5], we have the following results.

Algorithm 3: PR-IMM$(G, k, S_a, S_r, \ell, \varepsilon)$

Initial $\mathcal{R} = \emptyset$ and let $\ell' = \ell \cdot (1 + \log 2/\log n)$;
$\langle \mathcal{R}, \{\Psi(u)\}_{u\in V} \rangle$ =SampleGeneration$(G, k, \ell', \varepsilon)$;
S =GreedySelection$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u\in V})$;
return S

Algorithm 4: SampleGeneration$(G, k, S_a, S_r, \ell, \varepsilon)$

Initialize a set $\mathcal{R} = \emptyset$ and an integer $LB = 1$;
Initialize $\Psi(u) = 0$ for each $u \in V$;
for $i = 1$ *to* $\log_2 n - 1$ **do**
  for $j = |\mathcal{R}|$ *to* θ_i **do**
    R =PR-sketch-Generation(G, S_a, S_r, v);
    $\{\psi(u, R)\}_{u\in V}$ =GainComputation(S_a, S_r, R);
    $\forall u \in V, \Psi(u) = \Psi(u) + \psi(u, R)$;
    Insert PR-sketch R into $\mathcal{R}$;
  S = GreedySelection$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u\in V})$;
  if $n \cdot \Phi(S, \mathcal{R}) \geq (1 + \sqrt{2} \cdot \varepsilon) \cdot n/2^i$ **then**
    $LB = n \cdot \Phi(S, \mathcal{R})/(1 + \sqrt{2} \cdot \varepsilon)$;
    break;
Compute $\theta = \frac{2n(\alpha+\beta)^2}{LB\cdot\varepsilon^2}$;
for $j = |\mathcal{R}|$ *to* θ **do**
  R =PR-sketch-Generation(G, S_a, S_r, v);
  $\{\psi(u, R)\}_{u\in V}$ =GainComputation(S_a, S_r, R);
  $\forall u \in V, \Psi(u) = \Psi(u) + \psi(u, R)$;
  Insert PR-sketch R into $\mathcal{R}$;
return $\langle \mathcal{R}, \{\Psi(u)\}_{u\in V} \rangle$

***Lemma** 1:* If the GreedySelection returns an approximation for maximizing $\Phi(\cdot)$ with a ratio of δ, then with at least $1-n^{-\ell}$ probability, the set $\mathcal{I}$ returned by RILgeneration satisfies

$$|\mathcal{R}| \geq 2n_v(\alpha + \beta)^2\varepsilon^{-2}/OPT \quad \text{(Theorem.2 in [5])}.$$

Meanwhile, given the above equation holds, with a probability of $1 - n^{-\ell}$, GreedySelection returns a solution S satisfying

$$\lambda f(S) + (1 - \lambda)g(S) \geq (\delta - \varepsilon) \cdot OPT \quad \text{(Theorem.1 in [5])}.$$

Combining Lemma 2 and Lemma 3, Algorithm 3 ensures a $1 - 1/e - \varepsilon$ approximate ratio.

Now we analyze the time complexity. Denote EPT as the expected complexity of generating one random PR-sketch. According to [5], the complexity of SampleGeneration is

$$O(\mathbb{E}[|\mathcal{R}|] \cdot EPT) = O\left(\frac{EPT}{OPT}(k + \ell)(n + m)\varepsilon^{-2}\log n\right).$$

Meanwhile, as a variation of the maximum coverage algorithm, Algorithm 5 (GreedySelection Algorithm) runs with a time complexity of $O\left(\sum_{R\in\mathcal{R}} |R|\right)$ [4], i.e., linear to the input size. Combining Lemma 2, Lemma 3 and the above analysis, Algorithm 3 returns a $(1-1/e-\varepsilon)$-approximate solution with at least $1 - 2n^{-\ell'} = 1 - n^{-\ell}$ probability. Together with the complexity analysis, we have the following theorem.

Algorithm 5: GreedySelection$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u\in V})$

Initial $S = \emptyset$;
for $i = 1$ *to* k **do**
  $v = \arg\max_{v\in V\setminus S} \Psi(v)$;
  $S = S \cup \{v\}$;
  for *each PR-sketch* $R \in \mathcal{R}$ *that contains* v **do**
    $\{\phi(u, R)\}_{u\in V}$ = GainUpdate(S_a, S_r, S, R);
    $\forall u \in V, \Psi(u) = \Psi(u) - \psi(u, R) + \phi(u, R)$;
return S

Algorithm 6: GainUpdate(S_a, S_r, S, R)

Initialize $\phi(u, R) = 0, \forall u \in V$;
Let v be the root node of R;
for *all node* $u \in R$ **do**
  if $gain(S, R) = 1$ *or* $gain(S, R) = \lambda$ **then**
    $\phi(u, R) = 0$;
  else if $gain(S, R) = 1 - \lambda$ **then**
    if $\exists a \in S_a, d_1^R(a, u) + d_2^R(u, v) \leq d_r$ **then**
      $\phi(u, R) = \lambda$;
return $\{\phi(u, R)\}_{u\in V}$

***Theorem** 3:* With at least $1 - 1/n^\ell$ probability, The PR-IMM algorithm (Algorithm 3) returns a $(1 - 1/e - \varepsilon)$-approximate solution with a time complexity of $O\left(\frac{EPT}{OPT}(k + \ell)(n + m)\varepsilon^{-2}\log n\right)$.

V. Problem Solution for M-nS Case

We can see if $C_a \succ C_a^+$ or $C_r \succ C_r^-$, the objective function is not submodular. In Section VIII, we will analyze in which cases the objective function is not submodular. Here, we provide a data-dependent approximation algorithm.

A. Sandwich Approximation Strategy

If the objective function is monotone but not submodular, we can utilize the Sandwich Approximation (SA) Strategy [6] to find a data-dependent solution. Specifically, suppose we can find an upper bound function $U(S)$ and a lower bound function $L(S)$ of the objective function $h(S)$, where both $U(S)$ and $L(S)$ are submodular. Then we select three solution S_h, S_U and S_L greedily by maximizing $h(S)$, $U(S)$ and $L(S)$ respectively. Suppose S_U and S_L are $1 - 1/e - \varepsilon$ approximation solutions for maximizing $U(S)$ and $L(S)$. Let $S_{sa} = \arg\max_{S\in\{S_h,S_U,S_L\}} h(S)$. By SA strategy [6], we have

$$h(S_{sa}) \geq \max\{\frac{L(S^*)}{h(S^*)}, \frac{h(S_U)}{U(S_U)}\} \cdot (1 - 1/e - \varepsilon) \cdot OPT,$$

See the effectiveness of SA strategy depends on how close $U(S)$ and $L(S)$ are close to the objective function $h(S)$. Thus the problem becomes how to derive $U(S)$ and $L(S)$ that are as close to $h(S)$ as possible while maintaining submodularity.

Here we can see the upper bound function $U(S)$ is easy to derive. Just compute the gain under the priority setting of $C_a^+ \succ C_a \succ C_r^- \succ C_r$. Such a function $U(S)$ is ???

depends on λ. If $\lambda > 0.5$, then $C_a^+ \succ C_a \succ C_r^- \succ C_r$ yields an upper bound function of all other priority settings. If $\lambda < 0.5$, the upper bound function becomes $C_r^- \succ C_r \succ C_a^+ \succ C_a$.

To derive the submodular lower bound function, we first rewrite the objective function from a node-wise gain perspective. Let μ_{sv} denote the expected gain of node v if s is selected as assistant node. Then by the independency of influence spread process [7], [8], we know that $h(\{s\}) = \sum_v \mu_{sv}$.

Thus we can derive a lower bound function: $\underline{h}(S) = \sum_{v \in V} \max_{s \in S}\{\mu_{sv}\}$

***Lemma** 2:* The function $\underline{h}(S)$ is a lower bound function of $h(S)$ and it is monotone and submodular.

Proof: Let $h(S, v)$ denote the expected gain of node v with assistant node set S. Obviously, we know that $h(S, v) \geq \mu_{sv}$ for any $s \in S$. Thus we have $h(S) = \sum_{v \in V} h(S, v) \geq \sum_{v \in V} \max_{s \in S} \mu_{sv} = \underline{h}(S)$. This proves that $\underline{h}(S)$ is a lower bound function of $h(S)$.

Next, we can see the function $\underline{h}(S)$ is a simple summation of gain computation for each node once. For a specific node v, suppose the current maximal gain is achieved by assistant node s_1 in the current assistant node set S. Adding a new node s_2 into S either achieve a larger gain (i.e., $\mu_{s_2 v} > \mu_{s_1 v}$), or keep the same gain (i.e., $\mu_{s_2 v} \leq \mu_{s_1 v}$). Thus the monotonicity follows. Note in the former case, the marginal gain of adding s_2 is $\mu_{s_2 v} - \mu_{s_1 v}$, at least not larger than μ_{sv}. This holds true for any s_1, s_2 and S, which implies that $\underline{h}(S \cup \{s_2\}) - \underline{h}(S) \geq \underline{h}(S \cup \{s_1, s_2\}) - \underline{h}(S \cup \{s_1\})$. Thus the submodularity follows. ■

B. SA-IMM Algorithm

Lines 1-3 exactly follows the same process of PR-IMM algorithm (Algorithm 3). Here GreedySelectionUB function in line 3 is the GreedySelection algorithm (Algorithm 5). In line 4, we greedily select the solution S_h on the same PR-sketch sampled in line 2, for maximizing the objective function under the given priority setting. Then in lines 5-12 in Algorithm 7, we give the process of selecting an approximate lower bound solution S_L.

for a fixed node v, sample a set of PR-sketch $\mathcal{R}_v$.

We do not know which s yield the maximum gain, thus we need to modify the sampling scheme so as to guarantee an error bound for each s.

For any node s, we know $\mu_{sv} = \mathbb{E}[gain(s, R)]$

Let $F_v(s, \mathcal{R}_v) = \frac{\sum_{R \in \mathcal{R}_v} gain(s,R)}{|\mathcal{R}_v|}$

By similar martingale analysis of RR sets [5], the PR-sketch sampling possesses the same martingale tail bounds. That is, given a sequence of randomly generated PR-sketches $\mathcal{R}_v$ with $\theta = |\mathcal{R}_v|$, for any node $v \in S$ and $\mu_{sv} = \mathbb{E}[gain(s, R)]$, we have

$$\Pr[\sum_{R \in \mathcal{R}_v} gain(s, R) - \theta\mu_{sv} \geq \delta\theta\mu_{sv}] \leq \exp(-\frac{\delta^2}{2 + \frac{2}{3}\delta}\theta\mu_{sv}),$$

Algorithm 7: SA-IMM$(G, k, S_a, S_r, \ell, \varepsilon_1, \varepsilon_2)$

1 Initial $\mathcal{R} = \emptyset$ and let $\ell' = \ell \cdot (1 + \log 2/\log n)$;
2 $\langle \mathcal{R}, \{\Psi(u)\}_{u \in V}\rangle$ =SampleGeneration$(G, k, \ell', \varepsilon_1)$;
3 S_U =GreedySelectionUB$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u \in V})$;
4 S_h =GreedySelection$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u \in V})$;
5 $\kappa = \frac{\varepsilon_2}{2 - \varepsilon_2}$;
6 $\theta = (2 + \frac{2}{3}\kappa)(2 - \frac{1}{e} + \kappa)\frac{(\ell+1)\log n + \log 2}{\kappa^3(3 - \frac{1}{e})}$;
7 Initialize a set $\mathcal{R}_v = \emptyset$ for each node v;
8 **for** *each* $v \in V$ **do**
9   **for** $j = 0$ *to* θ **do**
10   R =PR-sketch-Generation(G, S_a, S_r, v);
11   Insert PR-sketch R into $\mathcal{R}_v$;
12 S_L =GreedySelectionLB$(S_a, S_r, \{\mathcal{R}_v\}_{v \in V}, k)$;
13 **return** $\langle S_U, S_h, S_L\rangle$

holds for any $\delta > 0$ and

$$\Pr[\sum_{R \in \mathcal{R}_v} gain(s, R) - \theta\mu_{sv} \leq -\delta\theta\mu_{sv}] \leq \exp(-\frac{\delta^2}{2}\theta\mu_{sv}),$$

holds for any $0 < \delta < 1$.

Now, we show the error bound of estimating μ_{sv} for any s and v using PR-sketch set $\mathcal{R}_v$.

***Lemma** 3:* Given $\mathcal{R}_v$ that

$$\theta = |\mathcal{R}_v| \geq (2 + \frac{2}{3}\kappa)\frac{(\ell + 1)\log n + \log 2}{\gamma \cdot \kappa^2},$$

it holds for all $s \in V$ simultaneously with a probability at least $1 - \frac{1}{n^\ell}$ that

$$\begin{aligned} &\text{if} \quad \mu_{sv} \geq \gamma, |\mu_{sv} - F_v(s, \mathcal{R}_v)| < \kappa\mu_{sv}, \\ &\text{if} \quad \mu_{sv} < \gamma, |\mu_{sv} - F_v(s, \mathcal{R}_v)| < \kappa\gamma. \end{aligned}$$

Proof: Fix a node s. We know that $\mu_{sv} = \mathbb{E}[gain(s, R)]$. Then $|\mathcal{R}_v| \cdot F_v(\mathcal{R}_v)$ can be regarded as $|\mathcal{R}_v|$ times of i.i.d. Bernoulli variables with a mean of μ_{sv}. If $\mu_{sv} < \gamma$, we have

$$\begin{aligned} &\Pr[|F_v(s, \mathcal{R}_v) - \mu_{sv}| \geq \kappa\gamma] \\ &\quad = \Pr[\sum_{R \in \mathcal{R}_v} gain(s, R) - \theta\mu_{sv} \geq \theta\frac{\kappa\gamma}{\mu_{sv}}\mu_{sv}] \\ &\quad \leq 2\exp(-\frac{\frac{\kappa^2\gamma^2}{\mu_{sv}}}{2 + \frac{2\kappa\gamma}{3\mu_{sv}}}|\mathcal{R}_v|) \\ &\quad = 2\exp(-\frac{3\kappa^2\gamma^2}{6\mu_{sv} + 2\kappa\gamma}\theta) \\ &\quad < 2\exp(-\frac{3\kappa^2\gamma^2}{6\gamma + 2\kappa\gamma}\theta) = \frac{1}{n^{\ell+1}}. \end{aligned}$$

Meanwhile, if $\mu_{sv} \geq \gamma$, then we have

$$\begin{aligned} &\Pr[|F_v(s, \mathcal{R}_v) - \mu_{sv}| \geq \kappa\mu_{sv}] \\ &\quad = \Pr[\sum_{R \in \mathcal{R}_v} gain(s, R) - \theta\mu_{sv} \geq \theta\kappa\mu_{sv}] \\ &\quad \leq 2\exp(-\frac{\kappa^2\mu_{sv}}{2 + \frac{2}{3}\kappa}\theta) \leq \frac{1}{n^{\ell+1}}. \end{aligned}$$

Taking the union bound, the lemma is proved. ■

Given Lemma 4 holds, then we have

$$
\begin{aligned}
\underline{h}(S) &= \sum_{v\in V} \max_{s\in S}\{\mu_{sv}\} \\
&= \sum_{v\in V_\gamma} h(s,v) + \sum_{v\in V\setminus V_\gamma} \mu_{sv} \\
&\geq \sum_{v\in V_\gamma} \frac{F_v(S,\mathcal{R}_v)}{1+\kappa} + \sum_{v\in V\setminus V_\gamma} (F_v(S,\mathcal{R}_v) - \kappa\gamma) \\
&\geq \sum_{v\in V} \frac{F(S,\mathcal{R}_v)}{1+\kappa} - n\kappa\gamma \\
&\geq \frac{1-1/e}{1+\kappa} \sum_{v\in V} F(S^*,\mathcal{R}_v) - n\kappa\gamma \\
&\geq \frac{1-1/e}{1+\kappa} (\sum_{v\in V_\gamma} (1-\kappa)h(S^*) + \sum_{v\in V\setminus V_\gamma} (h(S^*) - \kappa\gamma)) - n\kappa\gamma \\
&= (1-\frac{1}{e})\frac{1-\kappa}{1+\kappa} OPT - (\frac{1-1/e}{1+\kappa}+1)n\kappa\gamma
\end{aligned}
$$

See the approximation guarantee and the sampling number are both controlled by parameters κ and γ. By setting $\gamma = \frac{\gamma'}{n}OPT$, and setting κ to the value that satisfies

$$(1+\kappa)\varepsilon_2 = 2\kappa + (2 - \frac{1}{e} + \kappa)\gamma'\kappa.$$

We can make the approximation guarantee become a nice formulation, i.e., $\underline{h}(S) \geq (1-\frac{1}{e}-\varepsilon_2)OPT$.

By utilizing similar lower bound estimation method of IMM [5], we can get a lower bound of OPT to get a substitution value of γ that can still make the above guarantee holds. However, such setting may lead the sampling number dominated by a factor of n for each node v, which makes the total sampling number dominated by a factor of n^2. It is obviously time consuming with such a sampling number.

Therefore, we consider a more simple case by setting $\kappa = \frac{\varepsilon_2}{2(1-1/e))-\varepsilon_2}$. See with $\kappa \in (0,1]$, we have $\varepsilon \in (0, 1-\frac{1}{e}]$. Then we have

$$\underline{h}(S) \geq (1-\frac{1}{e}-\varepsilon_2)OPT - (\frac{1-1/e}{1+\kappa}+1)n\kappa\gamma.$$

We can further set $\gamma = \frac{\kappa(3-1/e)}{(2-1/e+\kappa)}$. See with $\kappa \in (0,1]$, we have $\gamma \in (0,1]$. Then we have

$$\underline{h}(S) \geq (1-\frac{1}{e}-\varepsilon_2)OPT - \frac{(3-1/e)\kappa^2}{1+\kappa}n.$$

other setting so more elegant formula?

The sampling number is then

$$\theta = |\mathcal{R}_v| = (2+\frac{2}{3}\kappa)(2-\frac{1}{e}+\kappa)\frac{(\ell+1)\log n + \log 2}{\kappa^3(3-1/e)},$$

Replace κ by $\frac{\varepsilon_2}{2-\varepsilon_2}$, the total time complexity is then $O(EPTn\ell\varepsilon_2^{-3}\log n)$ (ε_2^{-3}?) We omit the analytical formula here since it is too complex.

In this case, we see $\kappa\gamma = \frac{\kappa^2(3-1/e)}{(2-1/e+\kappa)}$. When $\kappa = 1$, $\gamma = 1$. Lemma 4 holds with the case that $h(s,v) \leq \gamma = 1$.

Algorithm 8: RG-IMM$(G, k, S_a, S_r, \ell, \varepsilon)$

Initial $\mathcal{R} = \emptyset$ and let $\ell' = \ell \cdot (1 + \log 2/\log n)$;
$\langle \mathcal{R}, \{\Psi(u)\}_{u\in V}\rangle$ =SampleGeneration$(G, k, \ell', \varepsilon)$;
S =RandomGreedy$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u\in V})$;
return S

Algorithm 9: RandomGreedy$(S_a, S_r, \mathcal{R}, k, \{\Psi(u)\}_{u\in V})$

Initial $S = \emptyset$ and $S_0 = \emptyset$;
for $i = 1$ *to* k **do**
  $\forall u \in V, \Psi'(u) = \Psi(u)$;
  for *each* $R \in \mathcal{R}$ *that contains a node in* S_{i-1} **do**
    $\{\phi(u,R)\}_{u\in V}$ = GainUpdate(S_a, S_r, S_{i-1}, R);
    $\forall u \in V, \Psi'(u) = \Psi'(u) - \psi(u,R) + \phi(u,R)$;
  Initial $S_i = \emptyset$;
  for $j = 1$ *to* k **do**
    $v = \arg\max_{v\in V\setminus\{S_i \cup S_{i-1}\}} \Psi'(v)$;
    $S_i = S_i \cup \{v\}$;
  Select a node u uniformly at random from S_i;
  $S = S \cup \{u\}$;
return S

When $\kappa = 0$, then $\gamma = 0$ and $\kappa\gamma = 0$. In this case, for all $h(s,v) \geq \gamma = 0$, we need to sample infinity PR-sketches to ensure zero error of all $h(s,v)$.

with ε being small, both the relative error factor $1-1/e-\varepsilon_2$ and the absolute error factor $\frac{\varepsilon_2^2(3-1/e)}{(2-\varepsilon_2)^2}$ correspondingly become small. We in experiments tune an appropriate value of ε_2 and observe the changing of performance.

***Theorem** 4:* For any $\varepsilon_1 > 0$, $\varepsilon_2 \in (0, 1-\frac{1}{e}]$ and $\kappa = \frac{\varepsilon_2}{2(1-1/e))-\varepsilon_2}$, with at least $1 - 2/n^\ell$ probability, the SA-IMM algorithm (Algorithm 7) returns a solution S_{sa} satisfies the following two data-dependent approximation guarantees:

$$
\begin{aligned}
h(S_{sa}) &\geq \frac{h(S_U)}{U(S_U)} \cdot (1-1/e-\varepsilon_1) \cdot OPT, \\
h(S_{sa}) &\geq \frac{L(S^*)}{h(S^*)} \cdot (1-1/e-\varepsilon_2) \cdot OPT - \frac{(3-1/e)\kappa^2}{1+\kappa}n.
\end{aligned}
$$

The time complexity is $O(\max\{T_1, T_2\})$, where $T_1 = \frac{EPT}{OPT}(k+\ell)(n+m)\varepsilon_1^{-2}\log n$ and $T_2 = EPTn\ell\varepsilon_2^{-3}\log n)$.

VI. Solution for nM-S and nM-nS Cases

We first consider the nM-S case. By replacing the GreedySelection function in PR-IMM algorithm (Algorithm 3) by the RandomGreedy Algorithm [9] (shown in Algorithm 9), we can achieve a $(\frac{1}{e}-\varepsilon)$-approximate solution. The corresponding RG-IMM algorithm is shown in Algorithm 8.

It is proved in [9] that the RandomGreedy algorithm can guarantee a $\frac{1}{e}$-approximate solution for maximization problem of non-monotone submodular function. Similar to Theorem 3, by replacing δ with $\frac{1}{e}$ in Lemma 2, we can derive the performance guarantee of RG-IMM Algorithm as follows.

Theorem 5: With at least $1 - 1/n^\ell$ probability, The RG-IMM algorithm (Algorithm 8) returns a $(1/e - \varepsilon)$-approximate solution with a time complexity of $O\left(\frac{EPT}{OPT}(k+\ell)(n+m)\varepsilon^{-2}\log n\right)$.

SA-RG-IMM algorithm for the nM-nS case. Similarly, we can replace all the GreedySelection functions (including GreedySelectionUB and GreedySelectionLB) by RandomGreedy algorithm. With all other steps the same to SA-IMM algorithm (Algorithm 7), we can derive an algorithm for the nM-nS case, which is named as SA-RG-IMM algorithm. Similar to Theorem 4, we have the following results by setting $\gamma = \frac{\kappa(2+1/e)}{1+1/e+\kappa}$ and $\kappa = \frac{\varepsilon_2}{2/e-\varepsilon_2}$.

Theorem 6: For any $\varepsilon_1 > 0$, $\varepsilon_2 \in (0, \frac{1}{e}]$ and $\kappa = \frac{\varepsilon_2}{2/e-\varepsilon_2}$, with at least $1 - 2/n^\ell$ probability, The SA-RG-IMM algorithm returns a solution S_{sa} satisfies the following two data-dependent approximation guarantee:

$$h(S_{sa}) \geq \frac{h(S_U)}{U(S_U)} \cdot (1/e - \varepsilon_1) \cdot OPT,$$

$$h(S_{sa}) \geq \frac{L(S^*)}{h(S^*)} \cdot (1/e - \varepsilon_2) \cdot OPT - \frac{(2+1/e)\kappa^2}{1+\kappa} n.$$

The time complexity is $O(\max\{T_1, T_2\})$, where $T_1 = \frac{EPT}{OPT}(k+\ell)(n+m)\varepsilon_1^{-2}\log n$ and $T_2 = EPT n \ell \varepsilon_2^{-3} \log n)$.

VII. Affection of Priority Settings

A. M-S Cases

Under three cases as shown in Table 1, the objective function is both monotone and submodular.

Lemma 4: The functions $f(\cdot)$ and $g(\cdot)$ are both monotone and submodular under C^{2}IC model, if the priority is given as one of the above three cases.

Proof: to be completed. ∎

B. M-nS Cases

If $C_a \succ C_a^+$ or $C_r \succ C_r^-$, then the objective function is not submodular. Remove the cases that satisfy the not monotone conditions, we can find 11 cases.

C. nM-S Cases

If $C_r^- \succ C_a \succ C_r$ or $C_r^- \succ C_a^+ \succ C_r$, while $C_a^+ \succ C_a$, the objective function is not monotone but submodular.

a node is supposed to be C_a-active, but may now become C_r^--active, which reduce the gain by λ.

D. nM-nS Cases

In nM-S cases, if $C_a \succ C_a^+$, then is neither monotone nor submodular. The first three cases in Table 1 of nM-nS.

In addition, if $C_a \succ C_r \succ C_a^+$ or $C_a \succ C_r^- \succ C_a^+$, the objective function is neither monotone nor submodular. As the other 4 cases in Table 1 of nM-nS.

TABLE I
MS Properties under Different Priority Settings

M-S	nM-S
$C_a^+ \succ C_a \succ C_r^- \succ C_r$	$C_a^+ \succ C_r^- \succ C_a \succ C_r$
$C_a^+ \succ C_r^- \succ C_r \succ C_a$	$C_r^- \succ C_a^+ \succ C_a \succ C_r$
$C_r^- \succ C_r \succ C_a^+ \succ C_a$	$C_r^- \succ C_a^+ \succ C_r \succ C_a$
M-nS	**nM-nS**
$C_a^+ \succ C_a \succ C_r \succ C_r^-$	
$C_a \succ C_a^+ \succ C_r^- \succ C_r$	
$C_a \succ C_a^+ \succ C_r \succ C_r^-$	$C_r^- \succ C_a \succ C_a^+ \succ C_r$
$C_a^+ \succ C_r \succ C_r^- \succ C_a$	$C_r^- \succ C_a \succ C_r \succ C_a^+$
$C_r \succ C_r^- \succ C_a^+ \succ C_a$	$C_a \succ C_r^- \succ C_a^+ \succ C_r$
$C_r^- \succ C_r \succ C_a \succ C_a^+$	$C_a \succ C_r^- \succ C_r \succ C_a^+$
$C_r \succ C_r^- \succ C_a \succ C_a^+$	$C_a \succ C_r \succ C_r^- \succ C_a^+$
$C_a^+ \succ C_r \succ C_a \succ C_r^-$	$C_a \succ C_r \succ C_a^+ \succ C_r^-$
$C_r \succ C_a^+ \succ C_a \succ C_r^-$	$C_r \succ C_a \succ C_r^- \succ C_a^+$
$C_r \succ C_a \succ C_a^+ \succ C_r^-$	
$C_r \succ C_a^+ \succ C_r^- \succ C_a$	

VIII. Experiments

References

[1] Y. Lin, W. Chen, and J. C. Lui, "Boosting information spread: An algorithmic approach," in *Data Engineering (ICDE), 2017 IEEE 33rd International Conference on*. IEEE, 2017, pp. 883–894.

[2] C. Budak, D. Agrawal, and A. El Abbadi, "Limiting the spread of misinformation in social networks," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 665–674.

[3] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, and Y. Yuan, "Influence maximization in social networks when negative opinions may emerge and propagate," in *Proceedings of the 2011 siam international conference on data mining*. SIAM, 2011, pp. 379–390.

[4] V. V. Vazirani, *Approximation algorithms*. Springer Science & Business Media, 2013.

[5] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: a martingale approach," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1539–1554.

[6] W. Lu, W. Chen, and L. V. Lakshmanan, "From competition to complementarity: comparative influence diffusion and maximization," *Proceedings of the VLDB Endowment*, vol. 9, no. 2, pp. 60–71, 2015.

[7] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2003, pp. 137–146.

[8] R. Yan, Y. Li, W. Wu, D. Li, and Y. Wang, "Rumor blocking through online link deletion on social networks," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 13, no. 2, pp. 1–26, 2019.

[9] N. Buchbinder, M. Feldman, J. Naor, and R. Schwartz, "Submodular maximization with cardinality constraints," in *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*. SIAM, 2014, pp. 1433–1452.