Triangular Stability Maximization by Influence Spread over Social Networks (Technical Report)

Zheng Hu
Fudan University
Shanghai, China
huz22@m.fudan.edu.cn

Weiguo Zheng
Fudan University
Shanghai, China
zhengweiguo@fudan.edu.cn

Xiang Lian
Kent State University
Kent, United States
xlian@kent.edu

ABSTRACT

In many real-world applications such as social network analysis and online advertising/marketing, one of the most important and popular problems is called *influence maximization* (IM), which finds a set of k seed users that maximize the expected number of influenced user nodes. In practice, however, maximizing the number of influenced nodes may be far from satisfactory for real applications such as opinion promotion and collective buying. In this paper, we explore the importance of *stability* and *triangles* in social networks, and formulate a novel problem in the influence spread scenario, named *triangular stability maximization*, over social networks, and generalize it to a *general triangle influence maximization* problem, which is proved to be NP-hard. We develop an efficient *reverse influence sampling* (RIS) based framework for the triangle IM with theoretical guarantees. To enable unbiased estimators, it demands probabilistic sampling of triangles, that is, sampling triangles according to their probabilities. We propose an *edge-based triple sampling* approach, which is exactly equivalent to probabilistic sampling and avoids costly triangle enumeration and materialization.

To further improve the time efficiency, we also design several pruning and reduction techniques, as well as a cost-model-guided heuristic algorithm. Extensive experiments and a case study over real-world graphs confirm the effectiveness of our proposed algorithms and the superiority of our proposed *triangular stability maximization* and triangle influence maximization.

PVLDB Reference Format:
Zheng Hu, Weiguo Zheng, and Xiang Lian. Triangular Stability Maximization by Influence Spread over Social Networks (Technical Report). PVLDB, 14(1): XXX-XXX, 2020. doi:XX.XX/XXX.XX

PVLDB Artifact Availability:
The source code, data, and/or other artifacts have been made available at https://github.com/triangleim/triangleim.

1 INTRODUCTION

With the booming development of IP/mobile networks and the impact of the post-epidemic era, social networks have played an increasingly important role in individuals' daily lives, as well as strategic advertising/marketing plans by companies and organizations, which are highly similar to real-world socializing. One of the most important and popular topics in social networks is the "Influence Maximization" (IM) problem [27], which finds a set of k seed users such that the expected number of users influenced by these seed users is maximized through the propagation process. As social networks have become progressively more functional, the need for IM is no longer just to obtain more nodes under the influence. Many IM variants such as competitive IM [5], time-aware IM [28], topic-aware IM [21], and location-aware IM [30] have attracted extensive attention. Although these variants have addressed different aspects of the IM problem, most of the variants are all seeking to simply maximize the number of influenced *nodes*, via some diffusion model, *without realizing that the influence propagation process also yields a sub-network structure induced by all influenced nodes*. Moreover, the number of influenced nodes is not the only quality metric for analyzing a network. [56] is a pioneer work that relocates the objective of influence maximization from nodes to edges and sought to maximize the so-called interaction strength with Sandwich Approximation [35]. A pioneer work [56] relocates the objective of influence maximization from nodes to edges and maximizes the so-called interaction strength in the subgraph induced by influenced nodes with Sandwich Approximation [35]. But it is still not aware that the influenced network creates opportunities for maximizing some *properties* that are related to particular *subgraph structures* (e.g., triangles [57]). For applications like collective buying and opinion promotion, the influenced users (corresponding to nodes) are expected to have a strong loyalty to the products or opinions. Thus, the influenced network is better to be as stable as possible. Let us consider the following examples.

Table 1: Statistics of Twitch users

Nodes	View	Lifetime (days)	Dead account rate
w/ Triangles	203,074	1,560.89	0.023
w/o Triangles	8,906	1,312.49	0.114

1.1 Motivating Examples

Opinion Promotion. The practical application of influence maximization is not limited to promoting products; it is also used to spread ideas, opinions, and even ideologies or religions. For such purposes, some topological properties, such as *stability*, of the network formed by influence propagation will be more important than mere quantity (i.e., the number of influenced nodes). Suppose an investor holds a certain opinion and she wants to spread this opinion firmly in the network. We might call the nodes that are influenced by this opinion "believers". When possible, the investor certainly wants as many believers as possible, and that is the goal of conventional IM. But if we consider this point, i.e., *if some believers*

This work is licensed under the Creative Commons BY-NC-ND 4.0 International License. Visit https://creativecommons.org/licenses/by-nc-nd/4.0/ to view a copy of this license. For any use beyond those covered by this license, obtain permission by emailing info@vldb.org. Copyright is held by the owner/author(s). Publication rights licensed to the VLDB Endowment.
Proceedings of the VLDB Endowment, Vol. 14, No. 1 ISSN 2150-8097.
doi:XX.XX/XXX.XX

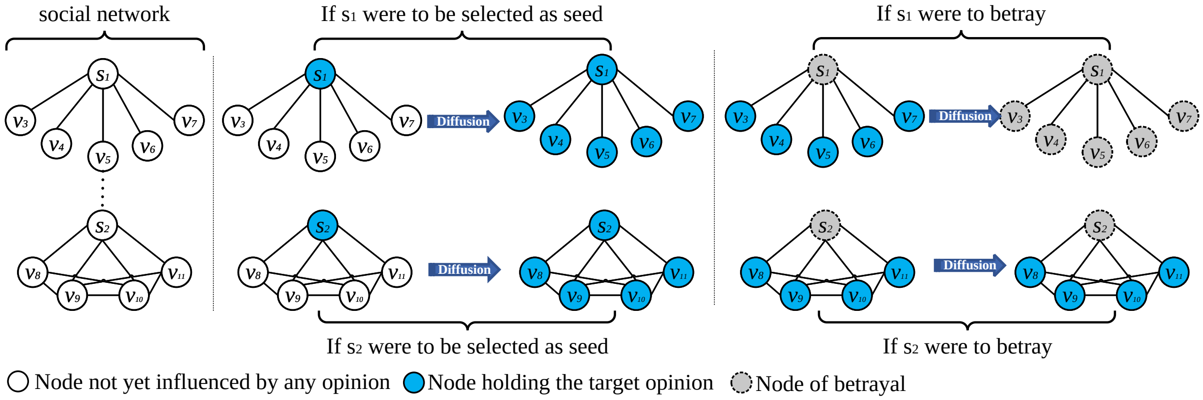


Figure 1: The left part shows a social network, where each node represents a user. Initially, these users do not hold any opinion. The middle and right parts illustrate the propagation process if s_1 or s_2 were to be selected as the seed and then betray.

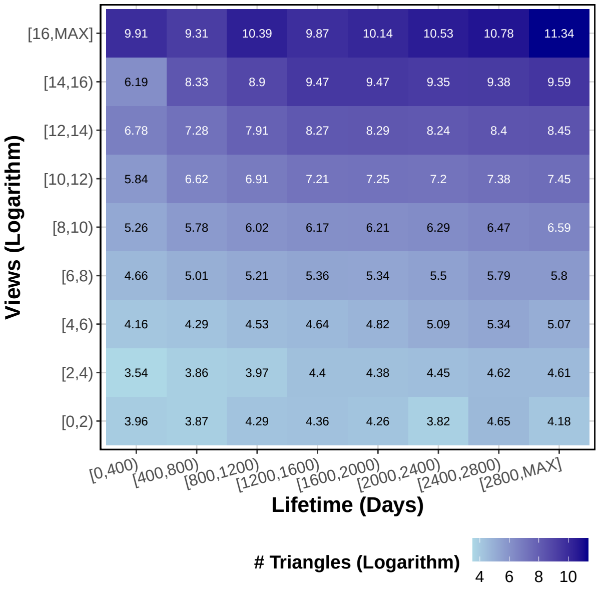
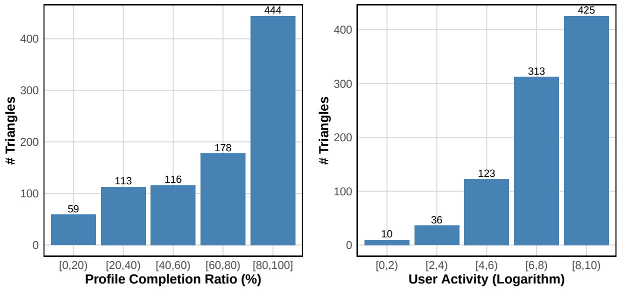


Figure 2: Distribution of #triangles on the Twitch dataset with respect to *Lifetime* and *Views*.



(a) Pokec

(b) LibraryThing

Figure 3: The correlation between the #triangles and user quality (i.e., profile completion ratio and user activity).

betray this opinion, will the influenced network consisting of believers become vulnerable or even collapse? This question is not considered in the previous IM task. It is well known that triangles are stable and highly correlated with important properties such as clustering coefficients, connectivity, etc. Li and Yu [32] argue that reducing the number of triangles effectively makes a network vulnerable from the attacker's perspective, which is also consistent with our motivation above. Thus, a natural idea to ensure the stability in an influence network is to increase the number of triangles involved in the influence propagation.

The left part of Figure 1 shows two components of a social network that are sufficiently distant from each other. For ease of presentation, we assume that in the initial state, opinions can freely spread in both parts of the network. Due to budget constraints, only one seed node is allowed. The conventional IM solution would suggest choosing s_1. However, this choice is risky if s_1 were to betray the network because a study in Nature [44] observed that players (referred to as "nodes") are inclined towards adopting the opinion of the majority that they witness in election-related influence networks. This means that the influenced network of s_1 would collapse as shown in the middle part of Figure 1, making it difficult to maintain the target opinion. In contrast, selecting s_2 may result in a smaller number of believers, but the sub-network of believers is more stable and less likely to collapse if s_2 were to betray the network, as shown in the right part of Figure 1 since the majority of the neighbors of the nodes influenced by s_2 hold the target opinion after the first round of the propagation. Our objective is to propose a tailored Influence Maximization in order to identify the seed nodes like s_2, which maximize the stability contributed by the influenced triangular structures. This concept the stability also applies to group buying in social commerce [60] which requires group members to be acquainted with each other. An opt-out by one participant in a fragile network might preclude the rest from organizing another group purchase.

Quality User Screening. Many recently emerging UGC (i.e., user-generated content) based platforms such as Tiktok and Twitch are embedded with functions such as entertainment content production and social networking. The nodes (users) on these platforms can serve as both producers (uploaders) and content consumers (viewers). That is, they are both influencers and influencees in information spread activities such as online marketing. It is a natural idea that users who contribute more to stability tend to have better real *attributes*. We still use the triangle as an indicator of stability. Table 1 depicts some statistics of users (nodes) involved or not involved in triangles of the Twitch network [43], including the number of content views, account lifetime, and the rate of dead accounts. These statistics indicate that a user node in triangular relationships with other users tends to be more "active" (in other words, more actively influence or be influenced by other users). Figure 2 reports the distribution of the number of triangles with respect to viewing and lifetime, it shows that the average number of triangles is positively correlated with views and lifetime, further supporting the above statement. Hence, to promote products or opinions on such UGC platforms, it is more important to identify targeting users who can induce an influenced network with abundant triangles composed of active users (rather than arbitrary, possibly inactive or dead, user accounts). Beyond the live-streaming platform, Twitch, we have identified a significant correlation between the number of triangles and the quality of users on other types of UGC platforms. Figure 3 illustrates how the user quality is affected by the number of triangles in two more datasets, a social network, *Pokec* [45], and a book review site, *LibraryThing* [9, 62]. This suggests that the number of triangles highly positively correlates with the presence of high-quality users. It becomes more important to leverage such triangular relationships, especially for third-party advertising agencies that do not have access to detailed user logs to obtain user activity directly.

Inspired by the examples above, the *triangle* can be considered one of the most important structures in social-network graphs. Beyond that, it is the basis for forming more complex structures like k-truss [14] and (k, d)-truss [26]. The number of triangles is also closely related to some important properties of the network, such as clustering coefficients [58] and transitivity [36], etc. As indicated in Table 4, most real-world (directed and undirected) graphs contain a large number of triangles, which we may leverage to enhance the stability of the influenced network. Influenced networks of this triangular nature will offer longer-term benefits, for example, building friendships or purchasing habits for users, and increasing the loyalty of users to the products.

Therefore, in this paper, we propose a novel problem, namely *triangular stability maximization (TSM)* by influence spread, which obtains a set of k seed users such that the expected *triangular structural stability score* in social networks is maximized after an influence propagation process. In particular, we consider a generalized problem, *general triangle influence maximization* (denoted as GΔIM), where we count the weights of triangles. To accommodate the algorithm design, we propose two upper and lower bound problems, *component and homologous triangle influence maximization* (denoted as CΔIM and HΔIM, respectively).

1.2 Challenges and Our Contributions

Since TSM and the triangle IMs suffer from intractable computational cost (i.e., NP-hardness as proved in Theorem 1), it is required to develop efficient algorithms while ensuring the quality of solutions. *Reverse influence sampling* (RIS) has become one of the widely used and promising approaches to the IM problem [40, 47, 48]. It keeps generating random *reverse reachable* (RR) sets until the total number of edges examined during the generation process reaches a pre-defined threshold. Nevertheless, the influenced targets in our triangle IM problems are *triangles* rather than *nodes*. To enable an unbiased estimator, it demands sampling triangles according to their probabilities in the graph. However, listing and materializing all triangles of a large graph may be infeasible, since the number of triangles may be much more than that of nodes (e.g., hundreds of times more), as studied in [52]. Another challenging problem is "empty intersection" arising from constructing homologous triangles based on RR sets, where homologous triangles are the triangles whose nodes are "activated" by the same seeds. If the RR sets of the three nodes those form a triangle do not share any node, no homologous triangles will be activated, leading to invalid samples. Hence, it urges the algorithm to generate more samples, resulting in an expensive time overhead. At the same time, the objective function of GΔIM is not submodular (as presented in Lemma 1), increasing the difficulty of employing the reverse influence sampling.

To address the challenges above, we propose a Joint Baking Algorithmic Framework for GΔIM, with efficient use of samples. Under the widely used diffusion models such as *independent cascade* (IC) [18] and *linear threshold* (LT) [19], we prove that GΔIM is monotonic, but not submodular; HΔIM is monotonic and submodular. In order to avoid the triangle materialization, we design an *edge-based triple sampling* approach which is exactly equivalent to sampling triangles according to their probabilities. To relieve the problem of empty intersection, we develop several techniques, including early pruning, dominance reduction, descendant reduction, and DFS-interval reduction.

In summary, we make the following contributions in this paper.

- To our best knowledge, we are the first to formulate *triangular stability maximization* by influence spread and *triangle influence maximization* problems, which are proved to be NP-hard. We also propose two submodular variants HΔIM and CΔIM as the lower and upper bound, respectively.
- We develop an efficient Joint Baking Algorithmic Framework for the triangle problem with theoretical guarantees. A novel *edge-based triple sampling* approach is proposed to avoid costly triangle enumeration and materialization.
- We design several reduction techniques to relieve the empty intersection problem and propose a cost-model-guided heuristic algorithm to improve the time efficiency.
- We evaluate our proposed algorithms through extensive experiments. The experimental results show that our algorithms produce seed sets of higher quality than the baseline. We also present a case study to illustrate the superiority of triangular stability maximization and triangle influence maximization.

2 PROBLEM DEFINITION

2.1 Preliminaries

We model the social network as a directed graph $G = (V, E)$, where V is the set of nodes and E is the set of directed edges. $\langle u, v, w \rangle$ is called a *triple*, where $u, v, w \in V$. The triple $\langle u, v, w \rangle$ forms triangles if there are edges between each pair of u, v, w.

2.1.1 Diffusion Models. Diffusion models describe the information diffusion process in a social-network graph.

Definition 1. *(Diffusion Model [33]) Given a social-network graph $G = (V, E)$ and a user set $S \subseteq V$, a diffusion model M captures the stochastic process for S spreading the information on G.*

Each edge $e(v, w) \in E$ is assigned a weight $p(v, w)$ representing the probability of the information propagation from v to w. Classical propagation (diffusion) models are *progressive*, which means that after a node has been activated, it cannot be extinguished or activated again. The most common models include *independent cascade* (IC) [18, 27], *linear threshold* (LT) [19, 27], and *triggering* (TR) [27] models.

Independent Cascade (IC) [27]. For each neighbor w of a node $v \in V$, there exists an edge $e(v, w)$ with the weight $p(v, w)$ representing the probability of the information spreading from v to w. If v is active and w is inactive at time t, then v will try to activate w with probability $p(v, w)$. Assuming that the activation is successful, w will become active at time $(t + 1)$. However, if all the attempts to activate w by its already active neighbors fail, then w will stay inactive. Regardless of the result, v will stop trying to activate w through the edge $e(v, w)$ in the future.

Linear Threshold (LT) [27]. Each node v is assigned a threshold θ_v. If there is no prior knowledge available about the node, the threshold will be selected randomly from the range $[0, 1]$. Let $N(v)$ denote the neighbors of v. For each neighbor $w \in N(v)$, there is a probability $p(v, w)$ corresponding to the edge $e(v, w)$, where $\sum_{w \in N(v)} p(v, w) \leq 1$. If v is inactive and it holds that $\sum_{w \in N_a(v)} p(v, w) \geq \theta_v$ at time t, v will be activated at the next time, where $N_a(v)$ is the set of active nodes in $N(v)$.

The TR model is also referred to as the Live Edge (LE) model. In [27], triggering sets are denoted by "live" and "blocked" edges. An edge $e(u, v)$ is "live" if node u belongs to the triggering set of v, otherwise, it is "blocked". A node u ends up active if and only if there is a path from some node in the seed set S to u consisting entirely of live edges. Such a path is called a *live-edge path.* **Both IC and LT are special cases of the TR model. [27]** The TR model expressed in the form of the LE model *does not* focus on the temporal sequence of activation actions (Claim 2.3 in [27]), so we follow it and say a node u is "activated" by a seed means that there is a live-edge path starting from this seed and ending at u. Then we say that a node u is influenced by the seed set S if and only if there exists a seed node in S that activates the node u.

2.1.2 Influence Maximization. Given a diffusion model M and a set, S, of nodes in V, we can compute the influence spread of S.

Definition 2. *(Influence Spread [33]) An influence spread (a.k.a. the influence function) of S, denoted as $\sigma_{G,M}(S)$, is given by the expected number of users influenced by S, where $\sigma_{G,M}(\cdot)$ is a function on a subset of users, i.e., $\sigma_{G,M} : 2^V \to \mathbb{R}_{\geq 0}$.*

Definition 3. *(Influence Maximization (IM) [27, 33]) Given a social-network graph G, a diffusion model M, and a positive integer k, an influence maximization (IM) problem selects a set, S^*, of k seed users from V that maximize the influence spread S^*, i.e., $\sigma_{G,M}(S^*) = \arg\max_{S \subseteq V \wedge |S| = k} \sigma_{G,M}(S)$.*

Table 2: Abbreviations and Symbols

Symbol	Meaning
n, m, nt	#nodes, #edges, #directed triangles of a graph
G/H/CΔIM	General/Homologous/Component Triangle IM
ω_{uvw}	The weight of triple $\langle u, v, w \rangle$
RR_u	A random RR set for IM
RR_{uvw}	A random RR sequence for GΔIM
RRI_{uvw}	A random RRI set for HΔIM
$\mathcal{R}$	The collection of samples
$\chi(S)$	The set of triangles influenced by the seed set S
$I(S)$	The set of nodes influenced by the seed set S
$Cov_{\mathcal{R}}(S)$	The number of samples covered by S in $\mathcal{R}$
$\mathcal{S}_3$	Triangular Structural Stability Score
σ	The objective function of the original problem
μ	The objective function of the lower-bound problem
ν	The objective function of the upper-bound problem

2.1.3 RR Sets and RIS. Let g be a subgraph obtained by removing each edge e in G with a certain probability. A *reverse reachable* (RR) set is defined as a set of nodes in g that can reach v, where v is selected uniformly at random from g. Borgs et al. [6] proposed a *reverse influence sampling* (RIS) approach (named by Tang et al. [48]) to solve the IM problem by using the RR sets.

2.1.4 Sandwich Approximation Strategy. Lu et al. [35] proposed an algorithmic strategy for solving non-submodular influence maximization problems, which is called "Sandwich Approximation". This strategy can give a solution with data-dependent approximation guarantees. Let $\sigma : 2^V \longrightarrow \mathbb{R}_{\geq 0}$ be the objective function (filling of a sandwich) of a non-submodular variant of IM. Let μ and ν be submodular functions (breads of a sandwich) such that $\mu(S) \leq \sigma(S) \leq \nu(S)$ for all $S \subseteq V$. Denote S^*_σ as the optimal solution of σ. Sandwich Approximation first runs the greedy algorithms on all three functions and produces S_μ, S_σ, and S_ν, respectively, where S_μ and S_ν are approximate solutions for μ and ν, and S_σ is a heuristic solution for σ. The solution of Sandwich Approximation is

$$S_{\text{sand}} = \arg\max_{S \in \{S_\mu, S_\sigma, S_\nu\}} \sigma(S)$$

Lu et al. proved that when S_μ and S_ν are $(1 - 1/e)$-approximation solutions, it hold that

$$\sigma\left(S_{\text{sand}}\right) \geq \max\left\{\frac{\sigma\left(S_\nu\right)}{\nu\left(S_\nu\right)}, \frac{\mu\left(S^*_\sigma\right)}{\sigma\left(S^*_\sigma\right)}\right\} \cdot (1 - 1/e) \cdot \sigma\left(S^*_\sigma\right).$$

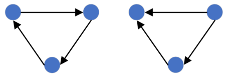
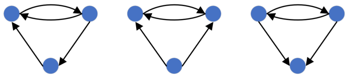
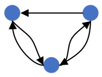
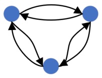
The equation above is further specified as Equation (1) by [56], when the algorithm uses a (γ, δ) estimator of σ, and both S_ν and S_μ are solved by RIS.

$$\sigma\left(S_{\text{sand}}\right) \geq \max\left\{\frac{\sigma\left(S_\nu\right)}{\nu\left(S_\nu\right)}, \frac{\mu\left(S^*_\sigma\right)}{\sigma\left(S^*_\sigma\right)}\right\} \cdot (1 - 1/e - \epsilon) \frac{1 - \gamma}{1 + \gamma} \cdot \sigma\left(S^*_\sigma\right). \quad (1)$$

2.2 Problem Statement

We follow the directed triangle patterns defined in Structural Stability Level [61]. For a given node u, it can participate in four categories of directed triangles as listed in Table 3, where triangles in each

Table 3: Triangular Structural Stability Score

Pattern	Score
	$\frac{1}{8}$
	$\frac{1}{4}$
	$\frac{1}{2}$
	1

row correspond to a category. We propose the triangular structural stability score ($\mathcal{S}_3$) according to the abundance of triangles.

Definition 4. *(Triangular Structural Stability Score.) Given a triple $\langle u, v, w\rangle$, we define its triangular structural stability score, denoted by $\mathcal{S}_3(\langle u, v, w\rangle)$, as the ratio of the number of directed triangles formed by $\langle u, v, w\rangle$ over the maximum number of directed triangles that can be produced by a triple. The triangular structural stability score of a graph G is defined as $\mathcal{S}_3(G) = \sum_{\langle u,v,w\rangle \subset G} \mathcal{S}_3(\langle u, v, w\rangle)$.*

As shown in the **last row** of Table 3, there are 8 different directed triangles at most for a triple $\langle u, v, w\rangle$. Table 3 presents the detailed score of each triple pattern, and the score of other patterns is 0.

2.2.1 Triangular Stability Maximization. We first introduce the concept of the influenced subgraph.

Definition 5. *(Influenced Subgraph). The activated nodes induce an influenced subgraph $G' = (V', E')$, where $V' = \{v_i : v_i \in V \wedge v_i$ is activated$\}$ and $E' = \{e(v_i, v_j) : e(v_i, v_j) \in E \wedge v_i, v_j \in V'\}$.*

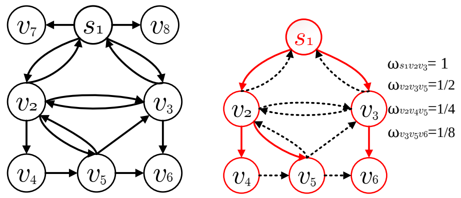
The subgraph composed of live-edge paths is essentially a subgraph of the influenced subgraph. We opt for "influenced subgraph" since every edge in it definitely exists within the social network and would contribute to stability, regardless of its propagation weight.

Problem Statement 1. (Triangular Stability Maximization *by Influence Spread, shorted as* **TSM.)** *Given a social-network graph G, a diffusion model M and a positive integer k, TSM returns a set, S^*, of k seed users from V to maximize the expectation of the triangular structural stability Score of the influenced subgraph $E_S[\mathcal{S}_3(G')]$, i.e., $S^* = \arg\max_{S\subseteq V\wedge|S|=k} E_S[\mathcal{S}_3(G')]$.*

TSM in essence aims to maximize the summed weights of triples that form triangles in the influenced subgraphs. Next, we generalize TSM to the general triangle influence maximization problem.

2.2.2 General Triangle IM. Assume each triple $\langle u, v, w\rangle$ that produces triangles has a weight $\omega_{uvw} \geq 0$. Let the summed weights of all triples that form triangles in G be $\Omega(G) = \sum \omega_{uvw}$.

Definition 6. *(General Triangle Influence Spread). A general triangle influence spread (a.k.a. triangle influence function) of S, denoted as $\Gamma_{G,M}(S)$, is defined as $E_S[\Omega(G')]$, the expected summed weights of the triples which are forming triangles in the influenced subgraph G'.*



(a) The original graph. (b) Influenced subgraph.

Figure 4: Example of an influenced subgraph.

Example 1. *Figure 4(b) shows an influenced subgraph and the weights of influenced triples that form triangles for the graph in Figure 4(a). The red edges represent being activated during propagation and the black dashed edges represent not being activated, but they are still part of the influenced subgraph. It can be seen that v_7 and v_8 are not activated, so their adjacent edges are not contained in the influenced subgraph. Each number on the right part represents the weight of these three nodes in the case of triangular stability maximization. For example, the weight of $\langle s_1, v_2, v_3\rangle$ is 1.*

For the sake of brevity, we use "triangles" to refer to the triples that form the triangles when the context is without ambiguity. We say that a triangle is influenced, only when all its three nodes have been influenced. With Definitions 5 and 6, we are ready to formulate the problem of *general triangle influence maximization.*

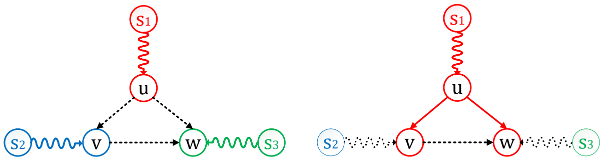
Problem Statement 2. (General Triangle Influence Maximization, GΔIM). *Given a social-network graph G, a diffusion model M and a positive integer k, GΔIM returns a set, S^*, of k seed users from V to maximize the general triangle influence spread $\Gamma_{G,M}(S)$, i.e., $S^* = \arg\max_{S\subseteq V\wedge|S|=k} \Gamma_{G,M}(S)$.*

It is clear that the objective function $E_S[\mathcal{S}_3(G')]$ of triangular stability maximization is a special case of the objective function $E_S[\Omega(G')]$ of general triangle influence maximization. They are equal when $\omega_{uvw} = \mathcal{S}_3(\langle u, v, w\rangle)$. Let the set function $\chi(S)$ be the set of triangles influenced by the set S. To make the expression easy to understand, we use $\Omega(\chi(S))$ to refer to $\Omega(G')$, where G' is an influenced subgraph by S. In the GΔIM problem, we can use any widely used diffusion model M, such as IC [27] or LT [27].

Lemma 1. *Under the IC or LT model, the objective function of GΔIM is monotonic, but not submodular.*

Proof. We prove the monotonicity as follows. Since both IC and LT models are progressive models, a node is not extinguished after it is activated. So the triangle will not be extinguished after all three nodes of the triangle are activated. Adding new seeds to the seed set does not reduce the number of activated nodes, so the summed weights of activated triangles will not be reduced either.

Non-submodularity can be proved by counterexamples. Assume that there are only three nodes $\{u, v, w\}$, forming a full triangle, in G. We also assume that it's in the case of Triangular Stability Maximization and the probabilities on all three nodes are 0. Then, there are two seed sets $S = \emptyset$ and $S' = \{u, v\}$, both of which have objective function $\Gamma^{\mathcal{H}}_{G,M}(\cdot)$=0 at this time. After adding w, we have: $\Gamma^{\mathcal{H}}_{G,M}(S \cup \{w\}) - \Gamma^{\mathcal{H}}_{G,M}(S) = 0 < 1 = \Gamma^{\mathcal{H}}_{G,M}(S' \cup \{w\}) - \Gamma^{\mathcal{H}}_{G,M}(S')$, which violates the submodularity. □



(a) Non-homologous triangle (b) Homologous triangle

Figure 5: An example of homologous triangles.

Next, we will define two submodular problems for sandwich approximation and to enhance time efficiency.

2.2.3 Lower Bound (Homologous Triangle IM). Different from GΔIM that considers triangles whose nodes can be influenced by any seed users, we next define its variant, *homologous triangle influence maximization* (HΔIM), that takes into account *homologous triangles* whose nodes are able to be all influenced by the same seed users.

Definition 7. *(Homologous Node & Homologous Triangle). In an influenced subgraph, those endpoints of live-edge paths from a same seed are mutually called homologous nodes. A triangle consisting of three homologous nodes is called a homologous triangle.*

Example 2. *Figure 5 shows an example of homologous triangles. We establish the seed set $S = \{s_1, s_2, s_3\}$ and assign them with distinct colors. The arrows connect the nodes u, v, and w are used to represent an edge, whereas a curve arrow directed from s_1, s_2, and s_3 to u, v, and w represents a path. The live paths, activated edges are solid, while a black dashed arrow signifies that the edge or path exists in the influenced subgraph but is not activated. In one propagation process, the nodes in triple $\langle u, v, w\rangle$ are activated by different seeds, forming a non-homologous triangle, as shown in Figure 5(a). In another propagation process, the triple $\langle u, v, w\rangle$ is activated by the same seed s_1, which constitutes a homologous triangle as shown in Figure 5(b).*

Note that, the term "activate" here is consistent with its definition in Section 2.1.1. We refer to a graph as a *"graph instance"* [56] in both IC and LT models if the graph is obtained by: marking each edge (u, v) as "live" with a probability $p(u, v)$ independently for the IC model, and marking at most one incoming edge (u, v) of each node v as "live" with a probability $1 - \sum_{u\in N(v)} p(u, v)$ for the LT model. Similar to Definition 6, we can define *homologous triangle influence spread*, $\Gamma^{\mathcal{H}}_{G,M}(S)$, as the expected summed weights of homologous triangles in the influenced subgraph.

Lemma 2. *$\Gamma^{\mathcal{H}}_{G,M}(S)$ is a lower bound of $\Gamma_{G,M}(S)$.*

Proof. Let $\Gamma_{G,M}(S|r)$ and $\Gamma^{\mathcal{H}}_{G,M}(S|r)$ denote the summed weights of the triangles and homologous triangles of the influenced subgraphs in a graph instance r, respectively. This means $\Gamma_{G,M}(S) = E[\Gamma_{G,M}(S|r)]$ and $\Gamma^{\mathcal{H}}_{G,M}(S) = E[\Gamma^{\mathcal{H}}_{G,M}(S|r)]$. For this graph instance r, the following inequality holds.

$$\begin{aligned}\Gamma^{\mathcal{H}}_{G,M}(S|r) \le \Gamma^{\mathcal{H}}_{G,M}(S|r) + \Omega(\{&\text{Influenced triangles that cannot be}\\ &\text{activated by the same seed.}\})\\ = \Gamma_{G,M}(S|r).&\end{aligned}$$

Next, we take the expectation on both sides of the inequality above to obtain the conclusion $\Gamma^{\mathcal{H}}_{G,M}(S) \le \Gamma_{G,M}(S)$. □

Now we are ready to formulate the problem of *homologous triangle influence maximization.*

Problem Statement 3. (Homologous Triangle Influence Maximization, HΔIM). *Given a social-network graph* G*, a diffusion model M, and a positive integer k, HΔIM obtains a set, S^*, of k seed users from V that maximize homologous triangle influence spread $\Gamma^{\mathcal{H}}_{G,M}(S)$, i.e., $S^* = \arg\max_{S\subseteq V\wedge|S|=k} \Gamma^{\mathcal{H}}_{G,M}(S)$.*

Lemma 3. *Under the IC or LT model, the objective function of HΔIM is monotonic and submodular.*

Proof. Monotonicity: Since both IC and LT are progressive models, a node is not extinguished after it is activated. A homologous triangle being activated means that all three nodes that make it up are activated and that these three nodes are the endpoints of the live-paths of the same seed. Since the nodes are not extinguished and the live-paths do not disappear when new seeds are added, the summed weights of homologous triangles is not reduced either.

Let $\gamma^{\mathcal{H}}(S)$ be the set of homologous triangles influenced by set S. $\gamma^{\mathcal{H}}(\{u\})$ is the set of triangles influenced by u. Suppose that $S \subseteq S'$ and $\gamma^{\mathcal{H}}(S) \subseteq \gamma^{\mathcal{H}}(S')$. Add a new node u. In a graph instance, consider the homologous triangles added by the addition of u to S, i.e., $\gamma^{\mathcal{H}}(S+\{u\}) \setminus \gamma^{\mathcal{H}}(S)$. If $u \in S$, $\gamma^{\mathcal{H}}(S+\{u\}) \setminus \gamma^{\mathcal{H}}(S) = \emptyset$. Therefore, $\Omega(\gamma^{\mathcal{H}}(S + \{u\})) - \Omega(\gamma^{\mathcal{H}}(S)) = 0$. If $u \notin S$, the added homologous triangles must belong to triangles composed of homologous nodes starting from u, and such triangles must be activated by $S + \{u\}$ as long as they have not been activated by S, i.e., $\gamma^{\mathcal{H}}(S + \{u\}) \setminus \gamma^{\mathcal{H}}(S) = \gamma^{\mathcal{H}}(u) \setminus \gamma^{\mathcal{H}}(S)$. Since $\gamma^{\mathcal{H}}(S) \subseteq \gamma^{\mathcal{H}}(S')$, we have $\{\gamma^{\mathcal{H}}(u) \setminus \gamma^{\mathcal{H}}(S')\} \subseteq \{\gamma^{\mathcal{H}}(u) \setminus \gamma^{\mathcal{H}}(S)\}$. Therefore, $\Omega[\gamma^{\mathcal{H}}(S'+\{u\})]-\Omega[\gamma^{\mathcal{H}}(S')] \le \Omega[\gamma^{\mathcal{H}}(S+\{u\})]-\Omega[\gamma^{\mathcal{H}}(S)]$ holds. □

Both GΔIM and HΔIM suffer from intractable computational cost as revealed in the following theorem.

Theorem 1. *The triangle IM problems are NP-hard.*

Proof. The triangle IM can be reduced from the IM problem which is proven to be NP-hard [27]. Let $G = (V, E)$ be an arbitrary graph instance of IM. For each node $u_i \in V$, we add $2|V|^3$ new nodes $v_{i_1}, v_{i_2}, \ldots, v_{i_{|V|^3}}$ and $w_{i_1}, w_{i_2}, \ldots, w_{i_{|V|^3}}$. Then we build $|V|^3$ undirected triangles of equal weight $\langle u_i, v_{i_1}, w_{i_1}\rangle$, $\langle u_i, v_{i_2}, w_{i_2}\rangle$, ..., $\langle u_i, v_{i_{|V|^3}}, w_{i_{|V|^3}}\rangle$ for u_i, where the influence probability of each edge is 1. Thus, we get a new graph $G' = (V', E')$. Suppose S is the answer to the triangle IM problem over the graph G'. S is the answer to the IM problem over G, otherwise there is a set S' which would produce more influence triangles than S. If the triangle IM problem can be solved in polynomial time, so will the IM problem as the reduction process is achieved in polynomial time, which contradicts the NP-hardness of the IM problem. □

2.2.4 Upper Bound (Component Triangle IM). A natural idea for designing the upper bound problem for the general triangle IM is to assign the weight of a triangle to each of the nodes in the triangle. This allows the nodes to contribute to the objective function when a "component" of the triangle is activated.

Let the component weight $\omega^{\mathcal{C}}_u$ of node u be $\sum \frac{\omega_{u\cdot\cdot}}{3}$, where $\langle u, \cdot, \cdot\rangle$ is a triple that contains u and forms triangles. Similar to Definition

6, we can define *component triangle influence spread*, $\Gamma^C_{G,M}(S)$, as the expected sum of component weights of the nodes in the influenced subgraph.

Lemma 4. *$\Gamma^C_{G,M}(S)$ is an upper bound of $\Gamma_{G,M}(S)$.*

Proof. Similar to Lemma 2, in a graph instance r, let $\Gamma_{G,M}(S|r)$ and $\Gamma^C_{G,M}(S|r)$ denote the summed weights of the triangles and the component weights of the nodes in the influenced subgraph, respectively. That is, $\Gamma_{G,M}(S) = E[\Gamma_{G,M}(S|r)]$ and $\Gamma^C_{G,M}(S) = E[\Gamma^C_{G,M}(S|r)]$. For this graph instance r, the following inequality holds.

$$\Gamma^C_{G,M}(S|r) = \Gamma_{G,M}(S|r) + \sum_{u\in V'} \sum_{\langle u,\cdot,\cdot\rangle \textbf{ is not activated.}} \frac{\omega_{u\cdot\cdot}}{3}$$
$$\geq \Gamma_{G,M}(S|r).$$

Next, we take the expectation on both sides of the inequality above to obtain the conclusion that $\Gamma^C_{G,M}(S) \geq \Gamma_{G,M}(S)$. □

Problem Statement 4. (Component Triangle Influence Maximization, CΔIM). *Given a social-network graph* G*, a diffusion model M, and a positive integer k, CΔIM obtains a set, S^*, of k seed users from V that maximize component triangle influence spread $\Gamma^C_{G,M}(S)$, i.e., $S^* = \arg\max_{S\subseteq V\wedge|S|=k} \Gamma^C_{G,M}(S)$.*

CΔIM is essentially a weighted conventional IM [55], so it also possesses monotonicity, submodularity, and NP-hardness.

3 ALGORITHMIC FRAMEWORK

3.1 Joint Baking Algorithmic Framework

In this section, we present a Joint Baking Algorithmic Framework (JBAF) to tackle the triangle IM problems, as illustrated in Algorithm 1. JBAF is a RIS-based variant of the sandwich approximation. The idea is to use RIS to solve (bake) the upper and lower bound problems (breads) by generate samples (reverse reachable structures) jointly in the RIS process to reduce the overhead of sample generation. Specifically, we estimate the objective function by continuously generating random RR sets. Then, it is transformed into a "Max-Coverage" problem, i.e., finding a set of k nodes such that this set intersects with as many RR structures as possible.

We call the process of randomly obtaining a node (or triple) and generating its corresponding random *reverse reachable* (RR) structure one *sampling* process. Each node (or triple) and its RR structure form a *sample*. The final seed set is returned when enough samples have been generated to guarantee the quality of the solution.

In Algorithm 1, it first sets the initial sample sizes, Λ_{L0} and Λ_{U0}, for lower and upper bound problems, respectively. Since the generated samples can be used for both problems, we select the larger sample size, Λ, for the first round. After that, the algorithm randomly generates Λ triples that form triangles and generates a corresponding RR structure for each triple. The subsequent process is similar to a typical RIS algorithm, wherein we double the sample count and perform the Max-Coverage procedure until the current seed sets meet the desired approximation ratio or the number of samples reaches the pre-determined maximum ($\Lambda_{L\max}$ for the lower-bound problem and $\Lambda_{U\max}$ for the upper-bound problem). Since the lower-bound and upper-bound problems may need to vary the number of samples, we can terminate the problem early once it reaches a sufficient sample count first. Since the original problem is non-submodular, we may have to use other tactics to solve it. After the solutions to the lower/upper bound problems and the original problem are finalized, we can return the optimal solution generated by the algorithm for each problem, for each problem.

If RIS is also used to solve S_σ, the original problem can likewise be involved in the process in lines 6-12. The data-dependent approximation guarantees for the sandwich approximation are independent of the correlation between the samples used in the upper and lower bound algorithms, so JBAF still maintains the same approximation guarantee as shown in Equation (1).

3.2 Theoretical Foundation

We first need to clarify some basic concepts. Let $\mathcal{R}$ denote a collection of samples and $Cov_{\mathcal{R}}(S)$ denote the number of samples covered by S in $\mathcal{R}$. Since our problems consider the triangles instead of nodes, the estimate $\frac{Cov_{\mathcal{R}}(S)}{|\mathcal{R}|}$ of the coverage of the RR sets also changes. In the classic IM problem, the estimate is an unbiased estimator of $\mathbb{E}\left[\frac{|I(S)|}{n}\right]$, where $I(S)$ is the set of nodes influenced by the seed set S and n is the number of nodes of a graph G. In the GΔIM problem, we design an RR set generation approach such that $\frac{Cov_{\mathcal{R}}(S)}{|\mathcal{R}|}$ is an unbiased estimator of $\mathbb{E}\left[\frac{\Omega(\chi(S))}{\Omega(G)}\right]$, where $\chi(S)$ is the set of triangles influenced by the seed set S and $\Omega(G)$ is the summed weights of the triples that form triangles in the whole graph.

Definition 8. *(Reverse Reachable Set & Reverse Reachable Sequence) Let $\langle u,v,w\rangle$ be a triple in graph $G = (V,E)$ (denoted as $\langle u,v,w\rangle \in V^3$ for simplicity), and a reduced subgraph g be a graph obtained by removing each edge e in G with the probability determined by the edge weight $p(e)$ and diffusion model M. A reverse reachable (RR) set for v (RR_v) in g is a set of nodes in g that can reach v. A reverse reachable (RR) sequence for $\langle u,v,w\rangle$, RR_{uvw}, is the sequence of the RR sets of u, v, w, i.e., $RR_{uvw} = \{RR_u, RR_v, RR_w\}$. In addition, we define the intersection of an RR sequence with a set S that is not empty as $RR_{uvw}\cap S \neq \emptyset \equiv (RR_u\cap S\neq\emptyset)\wedge(RR_v\cap S\neq\emptyset)\wedge(RR_w\cap S\neq\emptyset)$.*

In order to make our RIS estimator remain unbiased, the triple $\langle u,v,w\rangle$ then needs to be chosen with probability $\frac{\omega_{uvw}}{\Omega(G)}$. Since $\Omega(G)$ represents the total summed weights in the original graph, $\frac{\sum_{\langle u,v,w\rangle\in V^3}\omega_{uvw}}{\Omega(G)} = 1$. We can prove that under the above definition, $\frac{Cov_{\mathcal{R}}(S)}{|\mathcal{R}|}$ is an unbiased estimator of $\mathbb{E}\left[\frac{\Omega(\chi(S))}{\Omega(G)}\right]$ under GΔIM.

Example 3. *Figure 6 shows the subgraph of a reduced graph where (1) nodes can be reached by u, v, or w, and (2) all edges in the subgraph are activated reverse edges. Each RR set of u, v, and w consists of its descendants in the reduced graph and the node itself. The RR sequence of $\langle u,v,w\rangle$ is $RR_{uvw} = \{\{u,2,3,4\},\{v,1,2\},\{w,u,2,3,4,5\}\}$.*

Lemma 5. $\mathbb{E}\left[\frac{\Omega(\chi(S))}{\Omega(G)}\right] = \mathbb{E}\left[\frac{Cov_{\mathcal{R}}(S)}{|\mathcal{R}|}\right]$ *under GΔIM.*

Proof. In conventional IM, the probability that the RR set of a node v is covered by the seed set S is the probability that v is influenced. Similarly, the probability that the RR sequence of a triple $\langle u,v,w\rangle$ is covered by S is the probability that nodes u, v, and w are

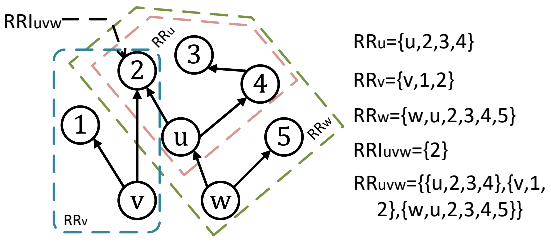


Figure 6: An example of an RR sequence and an RRI set.

Algorithm 1 Joint Baking Algorithmic Framework

```
Input: A graph $G$, a budget $k$, and an estimator of the objective
    function $\hat{\sigma}$
Output: A set, $S$, of seed nodes
 $\Lambda_{L0}$ ← the number of samples for the lower-bound problem
 $\Lambda_{U0}$ ← the number of samples for the upper-bound problem
 $\Lambda \leftarrow \max(\Lambda_{L0}, \Lambda_{U0})$
 $\{\langle u, v, w\rangle\}$ ← sample $\Lambda$ triples that form triangles
 $\mathcal{R}$ ←generate $\Lambda$ samples of $\{\langle u, v, w\rangle\}$
 repeat
     generate triples and their samples to double the size of $\mathcal{R}$
     if the generated samples for $S_\mu$ are not sufficient then
         $S_\mu$ ←Max-Coverage($\mathcal{R}, k$)    // Algorithm 2
     if the generated samples for $S_\nu$ are not sufficient then
         $S_\nu$ ←Max-Coverage($\mathcal{R}, k$)
 until a sufficient number of samples are generated for both $S_\mu$
     and $S_\nu$.
 $S_\sigma$ ← a solution of any strategy for the original problem
 $S \leftarrow \arg\max_{S \in \{S_\mu, S_\sigma, S_\nu\}} \hat{\sigma}(S)$
 return $S$
```

influenced by S, and also the probability that the triangles formed by $\langle u, v, w\rangle$ are influenced, denoted as $Pr_{\text{influenced}}(\langle u, v, w\rangle|S)$. We sample the triple by probability $\frac{\omega_{uvw}}{\Omega(G)}$ and get the derivation.

$$\mathbb{E}\left[\frac{Cov_{\mathcal{R}}(S)}{|\mathcal{R}|}\right] = \sum_{\langle u,v,w\rangle \in V^3} \frac{\omega_{uvw}}{\Omega(G)} Pr(RR_{uvw} \cap S \neq \emptyset)$$

$$= \sum_{\langle u,v,w\rangle \in V^3} \frac{\omega_{uvw}}{\Omega(G)} Pr_{\text{influenced}}(\langle u, v, w\rangle|S)$$

$$= \frac{\mathbb{E}_{\langle u,v,w\rangle \in \chi(S)} \omega_{uvw}}{\Omega(G)} = \frac{\mathbb{E}\Omega(\chi(S))}{\Omega(G)} = \mathbb{E}\left[\frac{\Omega(\chi(S))}{\Omega(G)}\right].$$

□

Similarly, we define the corresponding reverse reachable conception for HΔIM. This problem requires that the three nodes in a triangle can be influenced by the same seed in the seed set, and correspondingly, this actually requires that the intersection of the RR sets of these three nodes has an intersection with the seed set.

Algorithm 2 Max-Coverage

```
Input: A collection, $\mathcal{R}$, of samples and a budget $k$
Output: A set, $S$, of seed nodes
 $S \leftarrow \emptyset$
 for $i = 1$ to $k$ do
     $\hat{v} \leftarrow \arg\max_{v \in V} (Cov_{\mathcal{R}}(S \cup \{v\}) - Cov_{\mathcal{R}}(S))$
     insert $\hat{v}$ to $S$
 return $S$
```

Definition 9. *(Reverse Reachable Intersection Set) Let $\langle u, v, w\rangle$ be a triple in Graph $G = (V, E)$, and a reduced subgraph g be a graph obtained by removing each edge in G with a certain probability. The reverse reachable intersection (RRI) set for $\langle u, v, w\rangle$, RRI_{uvw}, is the intersection of the RR sets of u, v, w, i.e., $RRI_{uvw} = RR_u \cap RR_v \cap RR_w$.*

Example 4. *As shown in Figure 6, the RRI set of $\langle u, v, w\rangle$ is $RRI_{uvw} = RR_u \cap RR_v \cap RR_w = \{2\}$.*

Lemma 6. $\mathbb{E}\left[\frac{\Omega(\gamma^{\mathcal{H}}(S))}{\Omega(G)}\right] = \mathbb{E}\left[\frac{Cov_{\mathcal{R}^{\mathcal{H}}}(S)}{|\mathcal{R}^{\mathcal{H}}|}\right]$ *under HΔIM.*

We use the superscript $\mathcal{H}$ to denote the corresponding symbols under HΔIM. The proof procedure is similar to that of Lemma 5. According to Lemma 3 and [39], the approximation ratio of the solution of HΔIM is guaranteed to be $1 - 1/e - \epsilon$.

CΔIM is essentially a weighted conventional IM problem [55], and we use the superscript C to denote the corresponding symbols under CΔIM, therefore, $\mathbb{E}\left[\frac{\sum_{u \in I(S)} \omega_u^C}{\Omega(G)}\right] = \mathbb{E}\left[\frac{Cov_{\mathcal{R}^C}(S)}{|\mathcal{R}^C|}\right]$ holds under CΔIM and the greedy approximation algorithmic framework can also be guaranteed to get a $1 - 1/e - \epsilon$-approximate solution.

Specifically, although CΔIM is essentially a weighted conventional IM, it can still share samples with HΔIM. It is only necessary to select an RR set of the nodes with equal probability in a sampled triple $\langle u, v, w\rangle$. We can prove that because

$$\sum_{\langle u,v,w\rangle : u \in \langle u,v,w\rangle} \frac{1}{3} \frac{\omega_{uvw}}{\Omega(G)} = \frac{1}{\Omega(G)} \sum \frac{\omega_{u\cdot\cdot}}{3}$$

where $\frac{\omega_{uvw}}{\Omega(G)}$ is the sampling probability of $\langle u, v, w\rangle$ and the right hand side is proportional to $\sum \frac{\omega_{u\cdot\cdot}}{3}$ (i.e., the component weight of node u as defined in Section 2.2.4). This is exactly the sampling probability required by the RIS algorithm that is applicable to the weighted conventional IM [55].

4 ALGORITHM DETAILS

4.1 Edge-based Triple Sampling

To sample the triples of nodes according to their probabilities, a naive method is to compute probabilities for all triples by enumerating and materializing all the triangles in TSM. However, it is not practical as the space cost would be $O(|V|^3)$. A better way is storing the summed weights of triangles that each edge participates in and then performing edge-based samplings, reducing the storage overhead to $O(|E|)$. It is worth re-emphasizing that the edge-based triple sampling approach is NOT an approximate triangle counting method; it serves to sample triples with exact probability for use while taking up less space.

Edge-based triple sampling approach. Let ω_{uv} denote the summed weights of triangles containing an edge $e(u,v)$. When the algorithm is running, we sample an edge $e(u,v)$ with probability $\frac{\omega_{uv}}{\sum_{e(u,v)\in E}\omega_{uv}}$. Then we compute the common neighbors of u and v and sample the third node w in it based on the number of occurrences, which is proportional to $\frac{\omega_{uvw}}{\omega_{uv}}$. We can show that such a sampling approach is equivalent to sampling directly according to the triple probability.

Lemma 7. The edge-based triple sampling approach above *is equivalent to sampling directly according to the triple probability, i.e.,* $Pr(\langle u,v,w\rangle \textit{ is selected}) = \frac{\omega_{uvw}}{\Omega(G)}$.

Proof. To make it easy to understand, we can consider $e(u,v)$ as the existence of an arbitrarily oriented edge between nodes u,v and ω_{uv} as the summed weights of triangles consisting of arbitrarily oriented edges between nodes u,v. Notice that for the weight of each triangle consisting of triple $\langle u,v,w\rangle$, it's actually counted 3 times in $\omega_{uv},\omega_{vw},\omega_{uw}$. So it holds that $\Omega(G)=\frac{1}{3}\sum_{e(u,v)\in E}\omega_{uv}$. A triple $\langle u,v,w\rangle$ is sampled in only three cases: (1) edge $e(u,v)$ is sampled and afterward w is selected; (2) edge $e(v,w)$ is sampled and afterward u is selected; (3) edge $e(u,w)$ is sampled and afterward v is selected. Then we can obtain the following derivation.

$$
\begin{aligned}
&Pr(\langle u,v,w\rangle \text{ is selected})\\
&=\frac{\omega_{uv}}{\sum_{e(u,v)\in E}\omega_{uv}}\frac{\omega_{uvw}}{\omega_{uv}}+\frac{\omega_{vw}}{\sum_{e(u,v)\in E}\omega_{uv}}\frac{\omega_{uvw}}{\omega_{vw}}\\
&\quad+\frac{\omega_{uw}}{\sum_{e(u,v)\in E}\omega_{uv}}\frac{\omega_{uvw}}{\omega_{uw}}\\
&=\frac{\omega_{uv}}{3\Omega(G)}\frac{\omega_{uvw}}{\omega_{uv}}+\frac{\omega_{vw}}{3\Omega(G)}\frac{\omega_{uvw}}{\omega_{vw}}+\frac{\omega_{uw}}{3\Omega(G)}\frac{\omega_{uvw}}{\omega_{uw}}\\
&=\frac{\omega_{uvw}+\omega_{uvw}+\omega_{uvw}}{3\Omega(G)}=\frac{\omega_{uvw}}{\Omega(G)}.
\end{aligned}
$$

□

4.2 Generate RR Sequences & Intersection Sets

According to Definition 8, it is known that in GΔIM, the generation of RR sets for each node in a triple $\langle u,v,w\rangle$ must not be independent of each other. The process of generating an RR set is essentially the process of constructing an influenced subgraph. We perform the depth-first search (DFS) with them in the order of u,v,w respectively. When we finish generating RR_u, we also partially generate the reduced subgraph g (but the directions of all edges are *reserve*). When generating RR_v, if it meets a node in g, the expansion terminates at that node and connects to g at that node. Perform a similar operation when performing RR_w generation. On completion, the entire reduced subgraph related to u,v, and w is created. The reduced subgraph here is not composed of search trees. It is composed of the influenced nodes and all the edges that are activated, so it is likely to be a DAG and may even have cycles.

It is worth noting that during the DFS process, we do not know the ancestral relationship between nodes. So we need to do BFS or DFS on g after g is generated, and then do BFS or DFS on g with u,v, and w as the starting node to get their descendants, which are their RR sets, respectively. After generating RR sets, we generate RR sequences ($RR_{uvw}=\{RR_u,RR_v,RR_w\}$) for GΔIM or RR Intersection sets ($RRI_{uvw}=RR_u\cap RR_v\cap RR_w$) for HΔIM.

Example 5. *Figure 7 illustrates a process to generate RR sequences, where all edges are* reverse *edges activated during the generation process. First, blue nodes form the RR set of node u (i.e., RR_u). Then, we conduct a DFS from node v. When encountering a node previously expanded by u, we directly add this node and its descendants (nodes in Area (a)) to the RR set of node v (i.e., RR_v). Similarly, the nodes in Area (b) are directly added to the RR set of node w (i.e., RR_w). Only the RR set of the first node u is completely enumerated.*

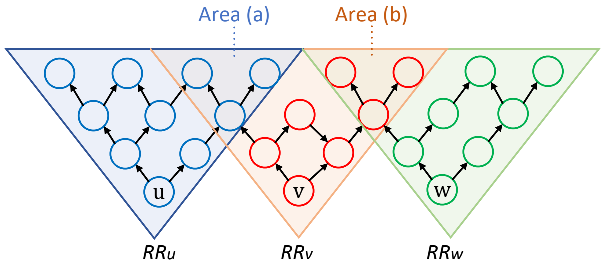


Figure 7: An illustration of generating the RR sequence.

RIS Time Complexity Analysis for GΔIM and HΔIM. Following the proof for conventional IM [48], we can guarantee the complexity $O(\frac{m}{n}\mathbb{E}[\sigma(v^*)])$ for generating a random RR set in the process of generating RR sequences for GΔIM and intersection sets for HΔIM, where v^* is sampled from a distribution where the probability of a node being selected is proportional to its in-degree. It is worth noting that this distribution differs from the ones employed in RIS for conventional IM or TSM, as it is primarily used to facilitate the time complexity analysis. In fact, due to the generation strategy above, it is not necessary to produce the complete RR set for every node. The actual time cost would be lower than $O(\frac{m}{n}\mathbb{E}[\sigma(v^*)])$.
The time complexity of generating an RR sequence or an RRI set is proportional to the size of an RR set (the merging of 3 RR sets). The complexity of building seed sets in "Max-Coverage" is linear for HΔIM. As for GΔIM, due to the loss of submodularity, the worst case requires recalculating the marginal benefits and ranking all nodes for each added seed, leading to the time complexity $O(kn\,(|\mathcal{R}|+\log(n)))$ by omitting the size of a single sample.
Theoretical Analysis of the sample size $|\mathcal{R}|$. Generally, the sample size $|\mathcal{R}|$ is the number of samples required to guarantee the approximation ratio. Specifically, it is determined by factors the maximum number of samples $\Lambda_{L\max}$, the initial sample size Λ_{L0}, and the approximation ratio as shown in the following theorem.

Theorem 2. *The expected number of sampled RRI sets for HΔIM is* $O\left((k\log n+\log(1/\delta))\Omega(G)\epsilon^{-2}/\Gamma^{\mathcal{H}}_{G,M}(S^o)\right)$ *to guarantee the approximation ratio* $1-1/e-\epsilon$ *when the following conditions are satisfied:* $\delta\le 1/2$, *the maximum number of samples* $\Lambda_{L\max}=\frac{2nt\left((1-1/e)\sqrt{\ln\frac{2}{\delta}}+\sqrt{(1-1/e)\left(\ln\binom{n}{k}+\ln\frac{2}{\delta}\right)}\right)^2}{\varepsilon^2(k/3)}$, *the initial sample size* $\Lambda_{L0}=\frac{\epsilon^2 k\Lambda_{L\max}}{3nt}$, *and the termination condition satisfies* $\frac{{\Gamma^{\mathcal{H}}_{G,M}}^{l}(S')}{{\hat{\Gamma}^{\mathcal{H}}_{G,M}}^{u}(S^o)}\ge 1-1/e-\epsilon$ *or the maximum number of samples* $\Lambda_{L\max}$ *is reached, where*

$$
{\Gamma^{\mathcal{H}}_{G,M}}^{l}(S')=\left(\left(\sqrt{\Phi_2(S')+\frac{2\log\left(\frac{3i_{\max}}{\delta}\right)}{9}}-\sqrt{\frac{\log\left(\frac{3i_{\max}}{\delta}\right)}{2}}\right)^2-\frac{\log\left(\frac{3i_{\max}}{\delta}\right)}{18}\right).
$$

$\frac{\Omega(G)}{|\mathcal{R}|/2}$ and $\hat{\Gamma}^{\mathcal{H}}_{G,M}{}^{u}(S^o) = \left(\sqrt{\Phi_1^u(S^o) + \frac{\log\left(\frac{3i_{\max}}{\delta}\right)}{2}} + \sqrt{\frac{\log\left(\frac{3i_{\max}}{\delta}\right)}{2}}\right)^2 \cdot \frac{\Omega(G)}{|\mathcal{R}|/2}$, $i_{\max} = \log\lceil\frac{\Lambda_{L\max}}{\Lambda_{L0}}\rceil$, S' and S^o denote current and optimal solutions to HΔIM, respectively. In the setup above, we have a sample collection $\mathcal{R}_1$ for constructing the seed set and another sample collection $\mathcal{R}_2$ of size $|\mathcal{R}_1|$ for estimating the approximation ratio. $\Phi_1^u(S)$ is an upper bound on the coverage $Cov_{\mathcal{R}_1}(S)$ of a solution S in $\mathcal{R}_1$, which should not be greater than $1/(1-1/e)$ times the true value. $\Phi_2(S)$ is the coverage $Cov_{\mathcal{R}_2}(S)$ of a solution S in $\mathcal{R}_2$.

Proof. *Let $a_1 = c\log(3i_{\max}/\delta)$ for any $c \geq 1$, and $\Lambda' = \max\left\{\frac{2\Omega(G)\log\frac{6}{\delta}}{\varepsilon_1^2\Gamma^{\mathcal{H}}_{G,M}(S^o)}, \frac{(2+2\tilde{\varepsilon}_1/3)\Omega(G)\log\frac{6\binom{n}{k}}{\delta}}{\tilde{\varepsilon}_1^2\Gamma^{\mathcal{H}}_{G,M}(S^o)}, \frac{27\Omega(G)\log\frac{3i_{\max}}{\delta}}{(1-1/e-\varepsilon)\varepsilon_1^2\Gamma^{\mathcal{H}}_{G,M}(S^o)}\right\}$, where $\varepsilon_1 = \varepsilon$, $\tilde{\varepsilon}_1 = \varepsilon-(1-1/e)\varepsilon_1 = \varepsilon/e$, $\hat{\varepsilon}_1 = \sqrt{\frac{2a_1\Omega(G)}{\Gamma^{\mathcal{H}}_{G,M}(S^o)\Lambda_1}}$, $\varepsilon_2 = \sqrt{\frac{2a_1\Omega(G)}{\Gamma^{\mathcal{H}}_{G,M}(S')\Lambda_2}}$, and $\tilde{\varepsilon}_2 = \left(\sqrt{\frac{2a_1\Gamma^{\mathcal{H}}_{G,M}(S')\Lambda_2}{\Omega(G)} + \frac{a_1^2}{9}} + \frac{a_1}{3}\right) \cdot \frac{\Omega(G)}{\Gamma^{\mathcal{H}}_{G,M}(S')\Lambda_2}$. We can verify that $\Lambda' \sim O\left((k\log n + \log(1/\delta))\Omega(G)\epsilon^{-2}/\Gamma^{\mathcal{H}}_{G,M}(S^o)\right)$. Let $\Lambda_1 = \Lambda_2 = c\Lambda'$, where $\Lambda_1 = |\mathcal{R}_1|$ is the number of samples $\mathcal{R}_1$ used to compute S' during Max Coverage and $\Lambda_2 = |\mathcal{R}_2|$ is the number of samples $\mathcal{R}_2$ used to verify the approximation guarantee of S'.*

From the perspective of martingale [47], we have the following inequalities. Given a fixed number of Λ random samples $\mathcal{R}$ and a seed set S, let $Cov_{\mathcal{R}}(S)$ be the coverage of S in $\mathcal{R}$. For any $\lambda > 0$,

$$\Pr\left[Cov_{\mathcal{R}}(S) - \Gamma^{\mathcal{H}}_{G,M}(S)\cdot\frac{\Lambda}{\Omega(G)} \geq \lambda\right] \leq \exp\left(\frac{-\lambda^2}{2\Gamma^{\mathcal{H}}_{G,M}(S)\cdot\frac{\Lambda}{\Omega(G)} + \frac{2}{3}\lambda}\right),$$

$$\Pr\left[Cov_{\mathcal{R}}(S) - \Gamma^{\mathcal{H}}_{G,M}(S)\cdot\frac{\Lambda}{\Omega(G)} \leq -\lambda\right] \leq \exp\left(\frac{-\lambda^2}{2\Gamma^{\mathcal{H}}_{G,M}(S)\cdot\frac{\Lambda}{\Omega(G)}}\right). \tag{2}$$

Then according to the inequalities above, the fact that $\Gamma^{\mathcal{H}}_{G,M}(S') \leq \Gamma^{\mathcal{H}}_{G,M}(S^o)$, and the settings in the theorem, we have:

$$\Pr\left[Cov_{\mathcal{R}_1}(S^o)\cdot\frac{\Omega(G)}{\Lambda_1} < (1-\varepsilon_1)\cdot\Gamma^{\mathcal{H}}_{G,M}(S^o)\right] \leq \left(\frac{\delta}{6}\right)^c, \tag{3}$$

$$\Pr\left[Cov_{\mathcal{R}_1}(S')\cdot\frac{\Omega(G)}{\Lambda_1} > \Gamma^{\mathcal{H}}_{G,M}(S') + \tilde{\varepsilon}_1\cdot\Gamma^{\mathcal{H}}_{G,M}(S^o)\right] \leq \left(\frac{\delta}{6\binom{n}{k}}\right)^c, \tag{4}$$

$$\Pr\left[Cov_{\mathcal{R}_1}(S^o)\cdot\frac{\Omega(G)}{\Lambda_1} < (1-\hat{\varepsilon}_1)\cdot\Gamma^{\mathcal{H}}_{G,M}(S^o)\right] \leq \left(\frac{\delta}{3i_{\max}}\right)^c, \tag{5}$$

$$\Pr\left[Cov_{\mathcal{R}_2}(S')\cdot\frac{\Omega(G)}{\Lambda_2} < (1-\varepsilon_2)\cdot\Gamma^{\mathcal{H}}_{G,M}(S')\right] \leq \left(\frac{\delta}{3i_{\max}}\right)^c, \tag{6}$$

$$\Pr\left[Cov_{\mathcal{R}_2}(S')\cdot\frac{\Omega(G)}{\Lambda_2} > (1+\tilde{\varepsilon}_2)\cdot\Gamma^{\mathcal{H}}_{G,M}(S')\right] \leq \left(\frac{\delta}{3i_{\max}}\right)^c. \tag{7}$$

Now, we know the probability that none of the events in Equations (3)-(7) occurs is at least

$$1 - \left(\left(\frac{\delta}{6}\right)^c + \left(\frac{\delta}{6\binom{n}{k}}\right)^c \cdot \binom{n}{k} + 3\cdot\left(\frac{\delta}{3i_{\max}}\right)^c\right) \geq 1 - \delta^c.$$

It is worth noting that the coefficient $\binom{n}{k}$ is due to the fact that S' and $Cov_{\mathcal{R}_1}$ in Equation (5) are not independent. So, it is necessary to include all possible k-node sets to satisfy inequalities in Equation (2), according to [46].

Next, the approximation guarantee of HΔIM can be proved and further used to derive the sample size. First, we have:

$$\hat{\varepsilon}_1 \leq \sqrt{\frac{2(1-1/e-\varepsilon)\varepsilon_1^2}{27}} < \varepsilon_1/3.$$

Then, the approximation guarantee can be derived as follows.

$$\begin{aligned}
\Gamma^{\mathcal{H}}_{G,M}(S') &\geq Cov_{\mathcal{R}_1}(S')\cdot\frac{\Omega(G)}{\Lambda_1} - \tilde{\varepsilon}_1\cdot\Gamma^{\mathcal{H}}_{G,M}(S^o) \\
&\geq (1-1/e)Cov_{\mathcal{R}_1}(S^o)\cdot\frac{\Omega(G)}{\Lambda_1} - \tilde{\varepsilon}_1\cdot\Gamma^{\mathcal{H}}_{G,M}(S^o) \\
&\geq (1-1/e)(1-\varepsilon_1)\cdot\Gamma^{\mathcal{H}}_{G,M}(S^o) - \tilde{\varepsilon}_1\cdot\Gamma^{\mathcal{H}}_{G,M}(S^o) \\
&= (1-1/e-\varepsilon)\cdot\Gamma^{\mathcal{H}}_{G,M}(S^o).
\end{aligned}$$

This approximate guarantee holds if the events of Equations (3) and (4) do not occur.

Based on the definitions of ϵ_2 and $\tilde{\varepsilon}_2$, the following inequality relation can be introduced.

$$\varepsilon_2 = \sqrt{\frac{2(1-1/e-\varepsilon)\Gamma^{\mathcal{H}}_{G,M}(S^o)\varepsilon_1^2}{27\Gamma^{\mathcal{H}}_{G,M}(S')}} < \varepsilon_1/3,$$

$$\text{and}\quad \tilde{\varepsilon}_2 = \sqrt{\frac{a_1(2+2\tilde{\varepsilon}_2/3)\Omega(G)}{\Gamma^{\mathcal{H}}_{G,M}(S')\Lambda_2}} \leq \sqrt{\frac{(2+2\tilde{\varepsilon}_2/3)\varepsilon_1^2}{27}} < \varepsilon_1/3.$$

When the event in Equation (5) does not occur, it holds that:

$$\left(\sqrt{Cov_{\mathcal{R}_1}(S^o) + \frac{a_1}{2}} + \sqrt{\frac{a_1}{2}}\right)^2 \cdot \frac{\Omega(G)}{\Lambda_1} \geq \sigma(S^o)$$

Next, the following inequality can be derived.

$$\begin{aligned}
1-\hat{\varepsilon}_1 = 1 - \sqrt{\frac{2a_1\Omega(G)}{\Gamma^{\mathcal{H}}_{G,M}(S^o)\Lambda_1}} &\leq 1 - \frac{\sqrt{2a_1}}{\sqrt{Cov_{\mathcal{R}_1}(S^o) + \frac{a_1}{2}} + \sqrt{\frac{a_1}{2}}} \\
&\leq \frac{Cov^u_{\mathcal{R}_1}(S^o)}{\left(\sqrt{Cov^u_{\mathcal{R}_1}(S^o) + \frac{a_1}{2}} + \sqrt{\frac{a_1}{2}}\right)^2}.
\end{aligned}$$

As $\log(3i_{\max}/\delta) \leq a_1$ due to $c \geq 1$. Then,

$$\hat{\Gamma}^{\mathcal{H}}_{G,M}{}^{u}(S^o) \leq \left(\sqrt{Cov^u_{\mathcal{R}_1}(S^o) + \frac{a_1}{2}} + \sqrt{\frac{a_1}{2}}\right)^2 \cdot \frac{\Omega(G)}{\Lambda_1} \leq \frac{Cov^u_{\mathcal{R}_1}(S^o)}{1-\hat{\varepsilon}_1}\cdot\frac{\Omega(G)}{\Lambda_1}. \tag{8}$$

where $\hat{\Gamma}^{\mathcal{H}}_{G,M}{}^{u} = \left(\sqrt{Cov^u_{\mathcal{R}_1}(S^o) + \frac{\log(1/\delta_1)}{2}} + \sqrt{\frac{\log(1/\delta_1)}{2}}\right)^2 \cdot \frac{\Omega(G)}{\Lambda_1}$ is an upper bound of $\Gamma^{\mathcal{H}}_{G,M}$ and can be derived in a similar way to the upper bound derivation used in OPIM-C [46].

When the event in Equation (7) does not occur, the following inequality relations can be derived.

$$\left(\sqrt{Cov_{\mathcal{R}_2}(S') + \frac{2a_1}{9}} - \sqrt{\frac{a_1}{2}}\right)^2 - \frac{a_1}{18} \leq \Gamma^{\mathcal{H}}_{G,M}(S')\cdot\frac{\Lambda_2}{\Omega(G)},$$

and

$$
\begin{aligned}
&Cov_{\mathcal{R}_2}\left(S'\right) - \tilde{\varepsilon}_2 \Gamma^{\mathcal{H}}_{G,M}\left(S'\right) \cdot \frac{\Lambda_2}{\Omega(G)} \\
&= Cov_{\mathcal{R}_2}\left(S'\right) - \left(\sqrt{2a_1\sigma\left(S'\right) \cdot \frac{\Lambda_2}{\Omega(G)} + \frac{a_1^2}{9}} + \frac{a_1}{3}\right) \\
&\leq Cov_{\mathcal{R}_2}\left(S'\right) - \left(\sqrt{2a_1 Cov_{\mathcal{R}_2}\left(S'\right) + \frac{4a_1^2}{9}} - a_1 + \frac{a_1}{3}\right) \\
&= \left(\sqrt{Cov_{\mathcal{R}_2}\left(S'\right) + \frac{2a_1}{9}} - \sqrt{\frac{a_1}{2}}\right)^2 - \frac{a_1}{18}.
\end{aligned}
$$

Similarly, it follows that:

$$
\Gamma^{\mathcal{H}}_{G,M}{}^{l}\left(S'\right) \geq Cov_{\mathcal{R}_2}\left(S'\right) \cdot \frac{\Omega(G)}{\Lambda_2} - \tilde{\varepsilon}_2 \Gamma^{\mathcal{H}}_{G,M}\left(S'\right), \tag{9}
$$

where $\Gamma^{\mathcal{H}}_{G,M}{}^{l}(S') = \left(\left(\sqrt{Cov_{\mathcal{R}_2}\left(S'\right) + \frac{2\log(1/\delta_2)}{9}} - \sqrt{\frac{\log(1/\delta_2)}{2}}\right)^2 - \frac{\log(1/\delta_2)}{18}\right) \cdot \frac{\Omega(G)}{\Lambda_2}$ *is a lower bound of* $\Gamma^{\mathcal{H}}_{G,M}$ *and can be derived in a similar way to the lower bound derivation used in OPIM-C [46].*

With Equations (8) and (9), when none of the events in Equations (3)-(7) occur, the termination condition will be reached.

$$
\begin{aligned}
\frac{\Gamma^{\mathcal{H}}_{G,M}{}^{l}\left(S'\right)}{\hat{\Gamma}^{\mathcal{H}}_{G,M}{}^{u}\left(S^o\right)} &\geq \frac{Cov_{\mathcal{R}_2}\left(S'\right) - \tilde{\varepsilon}_2 \Gamma^{\mathcal{H}}_{G,M}\left(S'\right) \cdot \frac{\Lambda_2}{\Omega(G)}}{Cov^{u}_{\mathcal{R}_1}\left(S^o\right)/\left(1-\hat{\varepsilon}_1\right)} \\
&\geq \frac{\left(1-\hat{\varepsilon}_1\right)\left(1-\varepsilon_2-\tilde{\varepsilon}_2\right)\Gamma^{\mathcal{H}}_{G,M}\left(S'\right) \cdot \frac{\Lambda_2}{\Omega(G)}}{Cov^{u}_{\mathcal{R}_1}\left(S^o\right)} \\
&> \left(1-\hat{\varepsilon}_1-\varepsilon_2-\tilde{\varepsilon}_2\right)\frac{\Gamma^{\mathcal{H}}_{G,M}\left(S'\right) \cdot \frac{\Lambda_2}{\Omega(G)}}{Cov_{\mathcal{R}_1}\left(S'\right)} \frac{Cov_{\mathcal{R}_1}\left(S'\right)}{Cov^{u}_{\mathcal{R}_1}\left(S^o\right)} \\
&> \left(1-\varepsilon_1\right)\frac{Cov_{\mathcal{R}_1}\left(S'\right) - \tilde{\varepsilon}_1 \cdot \Gamma^{\mathcal{H}}_{G,M}\left(S^o\right) \cdot \frac{\Lambda_2}{\Omega(G)}}{Cov_{\mathcal{R}_1}\left(S'\right)}(1-1/e) \\
&\geq \left(1-\varepsilon_1\right)\frac{Cov_{\mathcal{R}_1}\left(S'\right) - \tilde{\varepsilon}_1 Cov_{\mathcal{R}_1}\left(S^o\right)/\left(1-\varepsilon_1\right)}{Cov_{\mathcal{R}_1}\left(S'\right)}(1-1/e) \\
&\geq \left(1-\varepsilon_1\right)\left(1-\frac{\tilde{\varepsilon}_1/\left(1-\varepsilon_1\right)}{1-1/e}\right)(1-1/e) \\
&= 1-1/e-\varepsilon.
\end{aligned}
$$

Hence, if $\Lambda_1 = \Lambda_2 = c\Lambda'$ *RRI sets are generated, Algorithm 1 will not terminate unless at least one of the events described in Equations (3)-(6) occurs. The maximum probability for this occurrence is* δ^c. *Let* j *denote the initial iteration at which the number of RRI sets generated by the algorithm reaches* Λ'. *After this iteration, the expected number of additional RRI sets generated is at most:*

$$
\begin{aligned}
2 \cdot \sum_{z\geq j} \Lambda_{L0} \cdot 2^z \cdot \delta^{2^{z-j}} &= 2 \cdot 2^j \cdot \Lambda_{L0} \sum_{z=0} 2^z \cdot \delta^{2^z} \\
&\leq 4\Lambda' \sum_{z=0} 2^{-2^z+z} \\
&\leq 4\Lambda' \sum_{z=0} 2^{-z} \\
&\leq 8\Lambda'.
\end{aligned}
$$

If the algorithm terminates prior to this iteration, the maximum number of RRI sets generated will not exceed $2\Lambda'$. *Consequently, the expected number of RRI sets generated will not be larger than* $10\Lambda'$, *i.e.,* $O\left((k\log n + \log(1/\delta))\Omega(G)\epsilon^{-2}/\Gamma^{\mathcal{H}}_{G,M}(S^o)\right)$. □

4.3 Reduction for RRI Set Generation

As justified Table 5, RRI sets are likely to be empty. These empty RRI sets do not contribute to the final seed set construction, yet they increase time overheads. Therefore, we propose several techniques to enhance the intersection operation for HΔIM.

Early Pruning. Since we are asking for the intersection of three RR sets, as long as the intersection of any 2 of them is empty, we can end the generation process and return the empty intersection. Considering the generation strategy in Section 4.2, g is partially built up after the generation of RR_u is completed. In the process of generating RR_v and RR_w, if no node in g is encountered, the empty intersection can be returned directly. Under many models such as the weight cascade model, the probabilities on each edge are lower.

Degree-Oriented & Dominance Reduction. Most social networks follow the power-law distribution, indicating that most nodes of the graph are of a low degree. Intuitively, the set formed by expanding outward from the low-degree nodes tends to be much smaller. We require that before generating RRI sets, the nodes in the sampled triples are sorted in ascending order by in-degrees to ensure that in-degree(u) ≤ in-degree(v) ≤ in-degree(w). This helps to find the empty intersections as early as possible in the early pruning session. Since the sampled are triples that can form triangles, this means that there are edges between u, v, w. In the process of generating RR_u, if v is encountered, one can stop generating RR_u and go directly to generating RR_v. It is because in this case $RR_u \cap RR_v = RR_v$. It is similar for other nodes as well. We call this strategy “dominance reduction”.

Descendant Reduction. When generating RR_u, RR_v, and RR_w, we need to finally search again on the reduced subgraph g with u, v, and w as the starting nodes to get the nodes in the RR sets. But for HΔIM, since we are asking for their intersection RRI_{uvw}, we can reduce the search space. Let g be updated at the end of each node's DFS. Let B_1 and B_2 denote the set of nodes that meet g in the DFS process of v and w, respectively. Then we have Descendant(B_1) = $RR_u \cap RR_v$ and Descendant$(B_2) = (RR_u \cup RR_v) \cap RR_w$. We can obtain Descendant$(B_1) \cap$ Descendant$(B_2) = RR_u \cap RR_v \cap RR_w = RRI_{uvw}$. Thus, we just need to conduct a search from B_1 and B_2 to get their descendants and make an intersection.

DFS-Interval Reduction. The interval formed by the combination of pre-order traversal order and post-order traversal order can be used to determine the ancestral relationship between nodes [**?**]. Let $\phi_r(x)$ and $\phi_t(x)$ denote the visit index in the pre-order and post-order traversal of node x. Assume $\phi_r(u) = 1$, $\phi_t(u) = 1$, $\phi_r(v) = 2$, $\phi_t(v) = 3$, we have $[2,3] \subseteq [1,4]$, indicating that u is an ancestor of v. For a node $b_1 \in B_1$ and its descendants to be present in RRI_{uvw}, it is both necessary and sufficient that $\exists b_2 \in B_2$ s.t. $b_1 \in$ Descendant(b_2). For a node $b_2 \in B_2$ and its descendants to be present in RRI_{uvw}, it is both necessary and sufficient that $\exists b_1 \in B_1$ s.t. $b_2 \in$ Descendant(b_1). The conditions above allow us to exclude

Table 4: Statistics of Datasets

Dataset	n	m	nt	Type
DBLP	317K	1.05M	17.8M	Undirected
Enron	36.7K	184K	5.81M	Undirected
Epinions	132K	841K	13.3M	Directed
Pokec	1.63M	30.6M	123M	Directed
LiveJournal	4.85M	69.0M	1.12B	Directed

those nodes in B_1 and B_2 that will not enter the RRI set, further narrowing the search space and avoiding the intersection process.

For an RR set generation process, a node may have more than one parent. In such a case, the ancestor relationship may be present even if the interval does not satisfy the above relationship. Then it is required to search from B_1 and B_2 to find the intersection.

4.4 A Cost-Model-Guided Heuristic for GΔIM

Due to the non-submodularity of GΔIM, applying RIS to GΔIM does not produce approximation guarantees while invoking the very costly max-coverage process. Thus, RIS is not cost-effective to be applied to GΔIM. Next, we design a lightweight heuristic algorithm to solve GΔIM and get S_σ in Algorithm 1, based on a formal cost model.

In order to maximize the summed weights of the influenced triangles, the seeds of GΔIM should have relatively high values at least locally on the following two factors: One is that the node itself should participate in as many triangles with higher weights as possible, and the other is that the edges it activates should also participate in as many triangles with higher weights as possible. Considering a graph where all edges have been marked as "live" or "blocked", we can design a cost-model function to evaluate the quality score, $h(u)$, of each node u as follows.

$$h(u) = \omega_u + \sum_{e(u,v)\in E \wedge e(u,v) \text{ is live}} \omega_{uv}, \tag{10}$$

where ω_u is the summed weights of triangles containing node u and ω_{uv} is the summed weights of triangles containing the edge $e(u,v)$. Based on the cost model, we first sample "live" status for edges and compute $h(u)$ for each node, then sort all nodes in descending order of $h(u)$, and finally pick top-k nodes as the seed set.

5 EXPERIMENTAL EVALUATION

We evaluate our proposed algorithms for triangular stability maximization and compare them with the state-of-the-art IM algorithms.

5.1 Experimental Settings

Objectives. Our goal is to solve the Triangle Stability Maximization by influence spread on the datasets, that is, we need to solve the general triangle influence maximization problem that satisfies the weights $\omega_{uvw} = \mathcal{S}_3(\langle u, v, w\rangle)$.

Datasets. We tested five graphs, DBLP, Enron, Epinions, Pokec, and LiveJournal, which were downloaded from SNAP [29]. Table 4 summarizes the statistics of these graphs.

Algorithms.

- InfMax. The algorithm used to solve the conventional IM problem, here we refer to the state-of-the-art algorithm OPIM-C [46].
- Sandwich. We extend the Sandwich Approximation [35, 56] to the case of triangle influence maximization problems. Following the settings in [56], Stop-and-Stare [40] RIS (also named "Polling" in [56]) is used to solve the upper-bound, lower-bound, and original problems. In particular, "Polling" for weighted conventional IM is used to solve CΔIM, while we extend "Polling" with the generation strategy of Sections 4.1 and 4.2 so that it can solve GΔIM and HΔIM.[1]
- Bounds. A variant of Sandwich Approximation method by disabling the solution to the original problem (like a sandwich without fillings). Specifically, its solution $S_{\text{sand'}} = \arg\max_{S\in\{S_\mu, S_\nu\}} \sigma(S)$, where μ and ν are denoted as objective functions of lower and upper bound problems, respectively in Section 2.1.4. This method is used to compare with Sandwich and assess the quality of upper and lower bound problems we define.
- JBAF. Joint Baking Algorithmic Framework with applying all the strategies described in Section 3 and 4 to Algorithm 1.

Parameter Settings. For the weight of each edge $p(u,v)$, we follow the convention [40, 48] and set it to $\frac{1}{\text{in-degree}(v)}$. We set other parameters of RIS-based algorithms to default values, such as $\epsilon = \gamma = 0.1, \delta = \frac{1}{n}$. The setup above maintains the theoretical correctness of the Stop-and-Stare and OPIM-C algorithms under the new problems. In addition, we set the maximum timeout to 10,000 seconds.

Effectiveness Evaluation. To evaluate the quality of the delivered seed sets for different algorithms, these seeds are used to initiate the influence propagation in the network, and the number of influenced directed triangles indicates the triangular structural stability score (the larger the better). Formally, we define the metric, *structural stability ratio*, as the percentage of the influenced directed triangles among all directed triangles, i.e., $\mathcal{S}_3(\chi(S))/\mathcal{S}_3(G) \times 100\%$. We use reverse influence sampling to simulate the process above and generate 100K samples for each seed set to measure the final quality.

All experiments are conducted in a single thread on a Ubuntu 16.04.7 machine with Intel Xeon CPU E5-2678 v3 Processor @ 2.50GHz and 200 GB main memory.

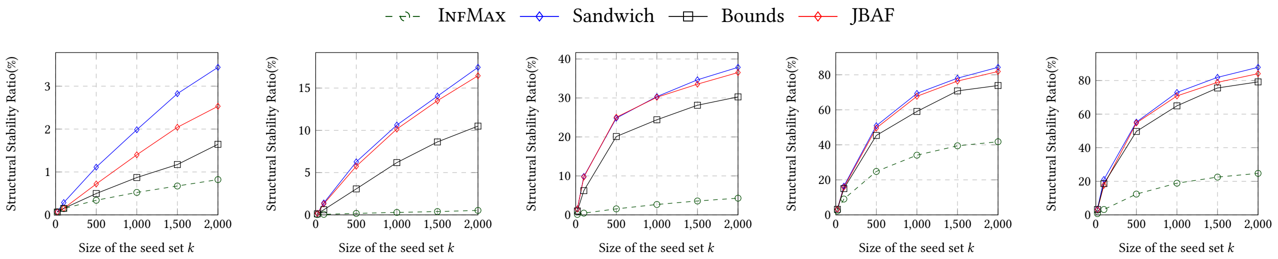
Source Codes. The datasets used in this experiment can be obtained at https://snap.stanford.edu/data/index.html. The source codes of our methods are available at the following repository https://github.com/triangleim/triangleim.

5.2 Results Under IC and LT Models

Figures 8-13 show the experimental results of the algorithms under the IC and LT models on the five datasets above. To show the trend of the characteristics of the five algorithms on our proposed problems (i.e., Triangular Stability Maximization) as well as on the different data sets, we set k = 20, 100, 500, 1, 000, 1, 500, 2, 000.

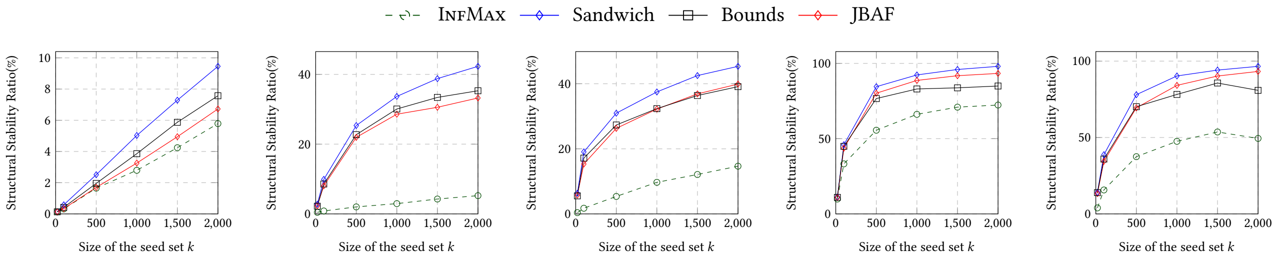
Solution Quality. As we observe in Figures 8-9, Sandwich can give the highest quality solutions. This means that the networks influenced by the seeds selected by Sandwich have the highest triangular structural stability score expectations. In the vast majority of cases, our JBAF is able to give solutions of almost the same quality as Sandwich. In particular, the quality of the JBAF solutions is highly consistent with Sandwich under the IC model

[1] [40] was noted to have some errors by [24], and we adopted its corrected version.



(a) Pokec (b) LiveJournal (c) DBLP (d) Enron (e) Epinions

Figure 8: Structural Stability Ratio under IC model.



(a) Pokec (b) LiveJournal (c) DBLP (d) Enron (e) Epinions

Figure 9: Structural Stability Ratio under LT model.

Table 5: Triangle Densities and Empty Intersection Rates

Dataset	nt/m	Empty Int. (IC)	Empty Int. (LT)
DBLP	17.0	88.4%	72.1%
Enron	31.6	97.5%	94.9%
Epinions	15.8	98.6%	93.8%
Pokec	4.0	99.2%	98.9%
LiveJournal	16.2	96.8%	76.4%

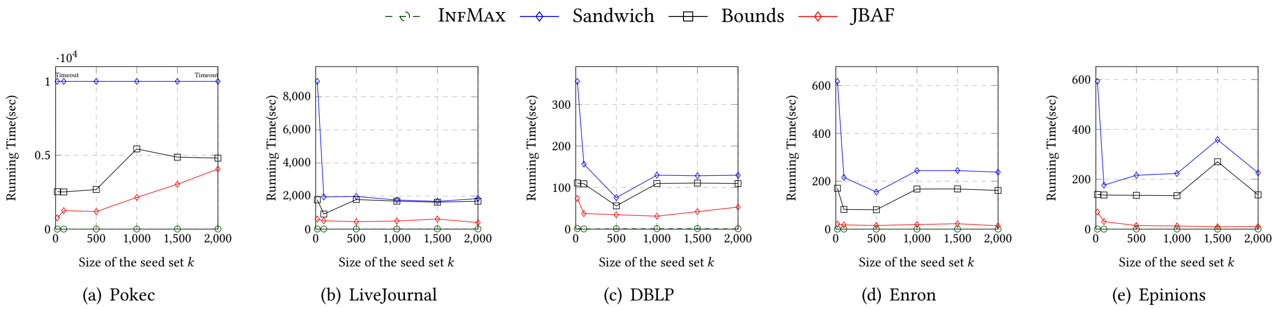
on data sets other than Pokec. The effectiveness of our algorithmic framework is verified by such experimental results. It also implies that our cost-model-guided heuristic also yields efficient solutions for GΔIM which are close to the results of the RIS algorithm that is used directly to solve GΔIM. The results of the conventional algorithm, InfMax, are significantly worse than our algorithms adapted to the corresponding problems in general. This point reflects the difference between the objectives of triangular stability maximization and conventional IM problems, and the existing algorithms for conventional IM are completely unable to solve the stability-related problems.

GΔIM conforms to this property on most data sets, that is, as k increases, the expected triangular structural stability score increases while the growth decreases. This is similar to the monotonicity and submodularity of the conventional IM and C/HΔIM. However, there are exceptions. As can be seen in Figure 8(a) and 9(a), the growth of the expected triangle influence spread on Pokec is almost linear, which is also consistent with Lemma 1.

The performance under the LT model is generally consistent with that under the IC model. However, we can see the following differences in terms of the quality of the solution: under the LT model (1) the various spreads are a bit higher. This suggests that there are more activations of triangles under the LT model than under the IC model. It is partly caused by the differences between the models themselves. (2) JBAF does not perform as well as under the IC model. This indicates that the tightness of the upper and lower bound problems is worse or the effectiveness of the heuristic algorithm is diminished under the LT model.

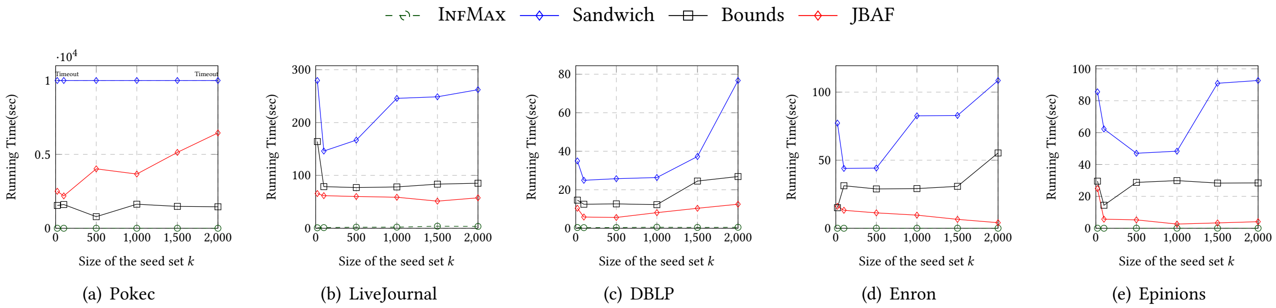
We observe that the algorithms behave differently on different data sets. For example, On LiveJournal, both Sandwich and JBAF are able to get significantly more triangular structural stability scores, while on Pokec this advantage is not so obvious and JBAF performs worse. It implies that there exists some property that may likewise affect the tightness of the upper and lower bound problems and the effectiveness of the heuristic. We have initially explored the factors that contribute to this situation. Table 5 shows the triangle density nt/m, where nt represents the number of directed triangles. It is worth noting that the edges of the undirected graph are transformed into two opposite directed edges. In most cases, on nt/m larger datasets, Sandwich and JBAF produce more triangles than InfMax and the gap is smaller on a nt/m smaller graph like Pokec, implying that the algorithms tailored for the TSM prefer structures with densely distributed triangles, otherwise their results fall back to those similar to the solutions of InfMax. The method Bounds also performs relatively well across data sets. This reflects the good approximation quality of our proposed upper and lower bound problems with respect to the original problem.

Overhead of the Algorithms. Figures 10-13 illustrate the time and number of samples generated for the algorithms. With the same settings, Sandwich spends significantly more time than JBAF. It is well known that for RIS-based algorithms, the running time depends heavily on the number of samples $|\mathcal{R}|$. In turn, $|\mathcal{R}|$ depends on the size and structure of the data set and the requirement of



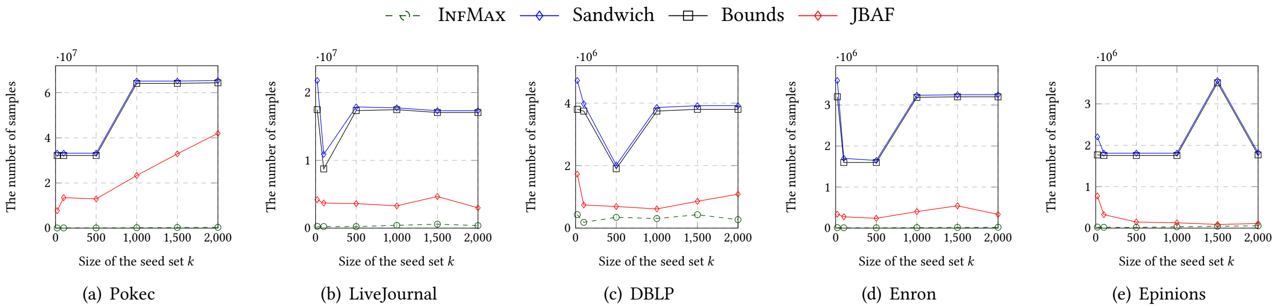
(a) Pokec (b) LiveJournal (c) DBLP (d) Enron (e) Epinions

Figure 10: Running time under IC model.



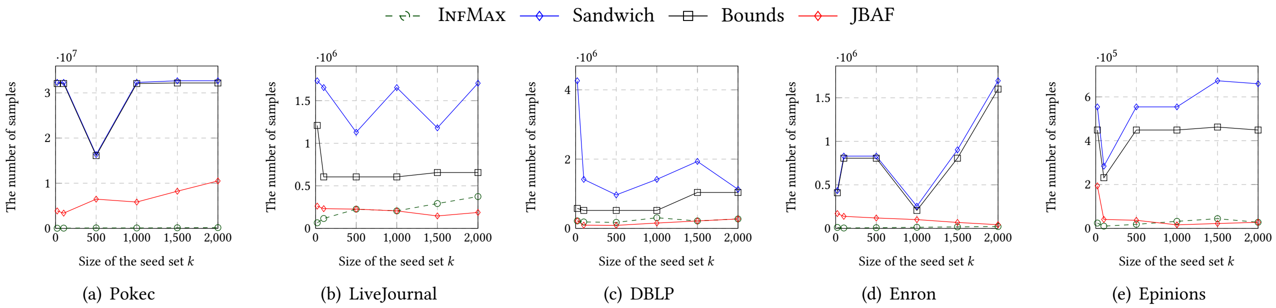
(a) Pokec (b) LiveJournal (c) DBLP (d) Enron (e) Epinions

Figure 11: Running time under LT model.



(a) Pokec (b) LiveJournal (c) DBLP (d) Enron (e) Epinions

Figure 12: The number of samples under IC model.



(a) Pokec (b) LiveJournal (c) DBLP (d) Enron (e) Epinions

Figure 13: The number of samples under LT model.

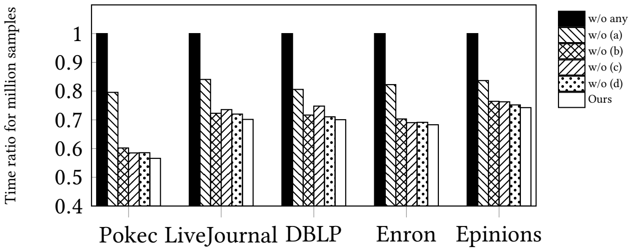


Figure 14: Effect of the pruning and reduction techniques.

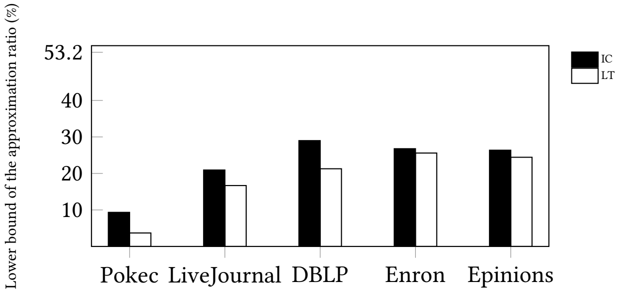


Figure 15: Average lower bound of the approximation ratio.

the algorithm to achieve approximate guarantees. Therefore, a direct reason for the efficiency of JBAF is that it is a framework for the efficient use of samples, avoiding a lot of duplicate sampling. In addition, the application of more advanced strategies and an effective and efficient heuristic algorithm also helps to avoid generating more samples. Figures 12-13 show that under both IC and LT models, JBAF generates much fewer samples than Sandwich. Another important reason is that Sandwich struggles with the issue that "Max-Coverage" for GΔIM cannot build the seed set in linear time, as mentioned in Section 4.2. In most cases, the number of samples generated by Bounds is close to that of Sandwich but still reduces the time to a large extent compared to Sandwich. This suggests that the RIS of the original problem may lead to uncontrolled time growth for just a small expansion of the sample size. This experimental phenomenon proves the importance of this reason. Using Pokec as an example, although the upper and lower bound problems are solved within the time limit, Sandwich still runs out of time.

JBAF is significantly more efficient compared to Sandwich, but we have to admit that there is still a long way to go to reduce the time overhead to the same level as solving conventional IM problems. The most intuitive reason is that all three nodes of a triangle require a reverse reachable operation, which leads to a higher overhead for a single sample. In addition, JBAF still requires the use of RIS to solve HΔIM. However, for HΔIM, there is a major problem that the generated RRI sets may be empty. These empty RRI sets urge the generation of a large number of samples and do not contribute to the construction of the seed set, but we cannot give up generating them because of the requirement of unbiasedness. This is one of the important reasons why we improve the efficiency for HΔIM in Section 4.3. The rate that an RRI set is empty at runtime on each data set is also reported in Table 5 and is referred to as empirical *empty intersection* rate for convenience. Granted the dataset size is smaller and often requires a smaller sample size. The effect of the empty intersection rate on the sample size due to the network structure of the dataset is also significant.

5.3 Efficiency of the RRI Set Generation

In this subsection, we evaluate how much the strategies proposed in Section 4.3 contribute to the efficiency improvement of generating RRI sets. We label the four strategies in Section 4.3 as (a), (b), (c), and (d) in order. Under the LT model, the generation of RR sets is essentially generating random walk paths. Since their intersection is easy to obtain, the experiments are conducted under the IC model. Figure 14 shows the relative time ratio required to generate 1 million samples by our method, the methods after removing one of the strategies, and the method after removing all strategies. It can be seen that the above strategies can reduce the RRI set generation time by about 30% to 45% on average. For a single strategy, the importance of (a), i.e., early pruning, is rather striking.

5.4 Quality of Approximation Guarantee

$\frac{(1-\gamma)^2}{(1+\gamma)^2} \cdot (1 - 1/e - \epsilon) \cdot \frac{\hat{\sigma}(S_v)}{\hat{v}(S_v)}$ is a computable lower bound of the approximation ratio for Equation (1), according to [56]. Figure 15 shows the average lower bound of the approximation ratio of JBAF on different data sets (averaged over k). The value of this lower bound ranges between 20% and 30% on the LiveJournal, DBLP, Enron, and Epinions datasets. As a data-dependent approximation guaranteed algorithm, JBAF performs well in terms of approximation as the lower bounds are close to the value of $1-1/e-\epsilon$ (=53.2%). The performance on the Pokec dataset is worse, especially under the LT model, which is also consistent with the performance of the quality of our solutions in Figures 8-9.

5.5 Case Study: Why Are Triangles Important

Twitch is an American video live streaming service that focuses on video game. Rozemberczki et al. [43] provide a social network dataset of Twitch gamers, which is an undirected network consisting of 168,114 users, 6,797,557 edges, and 54,148,895 undirected triangles. Each edge represents a mutually followed relationship.

We choose the rate of dead accounts, views, and lifetime as the criteria to evaluate the importance. The average global viewing is 188,162, the average lifetime is 1,542 days, and the average dead account rate is 0.031. Twitch dataset [43] also provides a variety of other useful information about these gamers such as when they created their accounts, when they last updated, and the language they use, however, it is more intuitive to use dead account indicators, views and lifetimes as measures of user importance and activity to emphasize the advantages of TSM. We present the expectation of the above properties for the nodes, our proposed homologous triangles (abbreviated as H-Triangles), and triangles in the influenced subgraph with the seed sets selected by InfMax, RIS for HΔIM, and Sandwich, respectively. The properties of a triangle are obtained by averaging the properties of its three nodes. Each score is reported by averaging 10 replicate experiments, generating 100K samples per experiment.

Table 6: Results of the Case Study (Twitch, IC)

	$k = 20$			$k = 100$			$k = 500$		
	View	Lifetime (days)	Dead rate	View	Lifetime (days)	Dead rate	View	Lifetime (days)	Dead rate
Nodes	246,879	1,528	0.030	260,565	1,541	0.033	288,867	1,550	0.034
H-Triangles	33,421,274	2,049	0.004	29,678,983	2,046	0.004	27,632,988	2,054	0.003
Triangles	53,275,466	2,142	0.003	38,800,349	2,168	0.003	23,996,320	2,184	0.003
L-Triangles	867,663	1,737	0.002	757,678	1,790	0.002	540,693	1,910	0.002

Table 7: Results of the Case Study (Twitch, LT)

	$k = 20$			$k = 100$			$k = 500$		
	View	Lifetime (days)	Dead rate	View	Lifetime (days)	Dead rate	View	Lifetime (days)	Dead rate
Nodes	187,473	1,544	0.030	194,350	1,546	0.031	198,821	1,547	0.031
H-Triangles	17,395,932	2,044	0.003	16,912,540	2,046	0.003	16,759,062	2,044	0.003
Triangles	19,479,234	2,053	0.003	16,980,948	2,060	0.003	15,538,536	2,057	0.003
L-Triangles	698,843	1,784	0.002	652,703	1,796	0.002	649,583	1,803	0.002

Table 8: Results of the Case Study (Pokec)

	Profile completion (%)					
	IC			LT		
k	20	100	500	20	100	500
Nodes	41.68	40.63	40.34	42.11	41.20	40.78
H-Triangles	52.14	49.70	50.67	49.57	49.24	50.67
Triangles	55.26	49.39	48.62	53.17	48.02	50.98

Tables 6 and 7 show the average scores of the relevant attributes of the corresponding structures of each algorithm under IC and LT models. The experimental results for each attribute score of the case study conducted under the LT model are similar to those under the IC model. From the tables, we can learn that triangles and homologous triangles tend to be more active. They have more views, lifetime, and lower dead account rates.

We report the average attribute scores of triangles, excluding those containing small-weight edges (denoted as "L-Triangles"). We refer to an edge having a *small* weight when its weight is less than 0.001. As presented in Tables 6 and 7, the views and lifetime of L-Triangles are significantly lower than those of Triangles and H-Triangles, accompanied only by a slight decrease in the dead rate. This validates the rationality of our problem using the influenced subgraph rather than the subgraph composed of living-edge paths.

We conduct similar experiments on the Pokec [45] dataset. The results shown in Table 8 demonstrate that there is a higher probability that triangles and homologous triangles will have their personal information completed on the social network, implying that they are more active and more likely to share more information to the platform.

Such results confirm the motivation of finding triangles, where users who frequently appear in the influenced triangles are more active and loyal to the platform, which may lead to more long-term benefits. At the same time, we do not require additional information to label the importance of nodes, allowing us to substitute weighted IM [55] in cases when attribute information is lacking.

6 RELATED WORK

6.1 Influence Maximization

Kempe et al. [27] provided several theoretical foundations of the IM problem, for example, the IM problem is proved to be NP-hard. They also formulated the *independent cascade* and *linear threshold* models, the IM problem in both of which models can be approximated to within a factor of $(1 - 1/e - \epsilon)$, where e is the base of the natural logarithm and ϵ is any positive real number less than $1 - 1/e$.

Many research studies [12, 23, 37, 40, 47] applied IM to viral marketing. In addition, making IM more contextual is a hot trend in the current research field, such as combining it with topics [21, 34, 38, 49], locations [10, 30, 53], time [28, 38], interaction strength [20, 56], and competitive information [8, 35, 50]. There are some studies [11, 13, 23, 54, 63, 64] that exploit the properties of communities or groups in social networks to address the IM problem. Recent works also consider properties of the influenced nodes that are related to the network structure, such as the diversity of communities [31]. Chandran et al. [11] and Zhang et al. [61] introduced triangle-related properties in the original network to heuristically solve the conventional IM. For more details please refer to the survey paper [33]. RIS-based algorithms [40, 47, 48] reduce the number of samples as much as possible while ensuring the quality of the solution. Online algorithms, e.g., OPIM [46], were derived to solve the IM problem. Guo et al. [22] further reduced the average size of random RR sets.

When constructing variants of the IM problem, researchers usually design the objective function to possess submodularity, which means that these variants are easy to handle in the framework of greedy algorithms. However, keeping submodularity of the objective function can be quite challenging and may not always be practical in certain situations. For example, the objective function of opinion-aware IM is non-submodular [16, 17]. The idea that nodes can switch their positive or negative opinions leads to such non-submodularity. There are also non-submodular objective functions in competitive IM that considers the simultaneous presence of multiple competitors within the network. Borodin et al. [7] extends the LT model to competitive scenarios and proves the non-submodularity of its objective function. Lu et al. propose the Sandwich Approximation strategy to solve the non-submodular IM problem and give a data-dependent approximation guarantee when studying the influence diffusion dynamics of products with arbitrary degrees of competition or complementarity [35]. Following the Sandwich Approximation strategy, non-submodular activity-related [20, 56] and community-related [23, 41] IM problems have been addressed. Huang et al. [25] recently study influence maximization over closed social networks, which is not submodular under the common IC model. Thus they develop a lower bound of the problem.

6.2 Triangle Counting

Triangle counting is a key computational task in network analysis. Researchers have invented many metrics related to the number of triangles, such as clustering coefficient [58] and transitivity ratio [36] to measure the quality of the network. Surveys like [2] have also shown that many efforts are using triangle counting to address tasks such as detecting web spam [3], revealing hidden topic structures [15], and performing community discovery [42]. As applications have increased, researchers also paid more attention to the actual time performance of triangle counting. It also drives the emergence of numerous approximation algorithms such as some sampling-based methods [1, 4, 51, 59].

7 CONCLUSION

In this paper, we propose *triangular stability maximization* by influence spread and *triangle influence maximization problems* which find a set of k seed users such that the expected summed weights of influenced triangles is maximized. We design an efficient RIS-based Sandwich variant framework for triangle IM problems with theoretical guarantees. To avoid enumerating and materializing all the triangles, a novel edge-based triple sampling approach is developed. We also present several pruning and reduction techniques to further improve time efficiency. Extensive experiments over real-world graphs demonstrate the effectiveness and efficiency of our proposed approaches.

REFERENCES

[1] Nesreen K. Ahmed, Nick G. Duffield, Jennifer Neville, and Ramana Rao Kompella. 2014. Graph sample and hold: a framework for big-graph analytics. In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1446–1455.

[2] Mohammad Al Hasan and Vachik S Dave. 2018. Triangle counting in large networks: a review. *WIREs Data Mining Knowl. Discov.* 8, 2 (2018), e1226.

[3] Luca Becchetti, Paolo Boldi, Carlos Castillo, and Aristides Gionis. 2008. Efficient semi-streaming algorithms for local triangle counting in massive graphs. In *SIGKDD*. 16–24.

[4] Suman K. Bera and C. Seshadhri. 2020. How to Count Triangles, without Seeing the Whole Graph. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 306–316.

[5] Shishir Bharathi, David Kempe, and Mahyar Salek. 2007. Competitive influence maximization in social networks. In *WINE*. 306–311.

[6] Christian Borgs, Michael Brautbar, Jennifer Chayes, and Brendan Lucier. 2014. Maximizing social influence in nearly optimal time. In *SODA*. 946–957.

[7] Allan Borodin, Yuval Filmus, and Joel Oren. 2010. Threshold models for competitive influence in social networks. In *Internet and Network Economics: 6th International Workshop, WINE 2010, Stanford, CA, USA, December 13-17, 2010. Proceedings 6*. Springer, 539–550.

[8] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. 2011. Limiting the spread of misinformation in social networks. In *WWW*. 665–674.

[9] Chenwei Cai, Ruining He, and Julian McAuley. 2017. SPMC: socially-aware personalized markov chains for sparse sequential recommendation. *arXiv preprint arXiv:1708.04497* (2017).

[10] Taotao Cai, Jianxin Li, Ajmal S Mian, Timos Sellis, Jeffrey Xu Yu, et al. 2020. Target-aware holistic influence maximization in spatial social networks. *IEEE Transactions on Knowledge and Data Engineering* (2020).

[11] Jyothimon Chandran et al. 2021. A Novel Triangle Count-Based Influence Maximization Method on Social Networks. *International Journal of Knowledge and Systems Science (IJKSS)* 12, 4 (2021), 1–17.

[12] Wei Chen, Chi Wang, and Yajun Wang. 2010. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *SIGKDD*. 1029–1038.

[13] Yi-Cheng Chen, Wen-Yuan Zhu, Wen-Chih Peng, Wang-Chien Lee, and Suh-Yin Lee. 2014. CIM: community-based influence maximization in social networks. *TIST* 5, 2 (2014), 1–31.

[14] Jonathan Cohen. 2008. Trusses: Cohesive subgraphs for social network analysis. *National security agency technical report* 16, 3.1 (2008).

[15] Jean-Pierre Eckmann and Elisha Moses. 2002. Curvature of co-links uncovers hidden thematic layers in the world wide web. *PNAS* 99, 9 (2002), 5825–5829.

[16] Sainyam Galhotra, Akhil Arora, and Shourya Roy. 2016. Holistic influence maximization: Combining scalability and efficiency with opinion-aware models. In *Proceedings of the 2016 international conference on management of data*. 743–758.

[17] Aristides Gionis, Evimaria Terzi, and Panayiotis Tsaparas. 2013. Opinion maximization in social networks. In *Proceedings of the 2013 SIAM International Conference on Data Mining*. SIAM, 387–395.

[18] Jacob Goldenberg and Libai Eitan Muller. 2001. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. *Marketing Letters* 12, 3 (2001), 211–223.

[19] Granovetter and Mark. 1978. Threshold Models of Collective Behavior. *Amer. J. Sociology* 83, 6 (1978), 1420–1443.

[20] Jianxiong Guo, Tiantian Chen, and Weili Wu. 2020. Continuous activity maximization in online social networks. *IEEE Transactions on Network Science and Engineering* 7, 4 (2020), 2775–2786.

[21] Jing Guo, Peng Zhang, Chuan Zhou, Yanan Cao, and Li Guo. 2013. Personalized influence maximization on social networks. In *CIKM*. 199–208.

[22] Qintian Guo, Sibo Wang, Zhewei Wei, and Ming Chen. 2020. Influence maximization revisited: Efficient reverse reachable set generation with bound tightened. In *SIGMOD*. 2167–2181.

[23] Huimin Huang, Hong Shen, Zaiqiao Meng, Huajian Chang, and Huaiwen He. 2019. Community-based influence maximization for viral marketing. *Applied Intelligence* 49, 6 (2019), 2137–2150.

[24] Keke Huang, Sibo Wang, Glenn Bevilacqua, Xiaokui Xiao, and Laks VS Lakshmanan. 2017. Revisiting the stop-and-stare algorithms for influence maximization. *Proceedings of the VLDB Endowment* 10, 9 (2017), 913–924.

[25] Shixun Huang, Wenqing Lin, Zhifeng Bao, and Jiachen Sun. 2022. Influence Maximization in Real-World Closed Social Networks. *Proc. VLDB Endow.* 16, 2 (oct 2022), 180–192. https://doi.org/10.14778/3565816.3565821

[26] Xin Huang and Laks VS Lakshmanan. 2017. Attribute-driven community search. *Proceedings of the VLDB Endowment* 10, 9 (2017), 949–960.

[27] David Kempe, Jon Kleinberg, and Éva Tardos. 2003. Maximizing the spread of influence through a social network. In *SIGKDD*. 137–146.

[28] Jinha Kim, Wonyeol Lee, and Hwanjo Yu. 2014. CT-IC: Continuously activated and time-restricted independent cascade model for viral marketing. *Knowledge-Based Systems* 62 (2014), 57–68.

[29] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data.

[30] Guoliang Li, Shuo Chen, Jianhua Feng, Kian-lee Tan, and Wen-syan Li. 2014. Efficient location-aware influence maximization. In *SIGMOD*. 87–98.

[31] Jianxin Li, Taotao Cai, Ke Deng, Xinjue Wang, Timos Sellis, and Feng Xia. 2020. Community-diversified influence maximization in social networks. *Information Systems* 92 (2020), 101522.

[32] Rong-Hua Li and Jeffrey Xu Yu. 2015. Triangle minimization in large networks. *Knowl. Inf. Syst.* 45, 3 (2015), 617–643.

[33] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. 2018. Influence maximization on social graphs: A survey. *TKDE* 30, 10 (2018), 1852–1872.

[34] Yuchen Li, Ju Fan, Dongxiang Zhang, and Kian-Lee Tan. 2017. Discovering your selling points: Personalized social influential tags exploration. In *SIGMOD*. 619–634.

[35] Wei Lu, Wei Chen, and Laks VS Lakshmanan. 2015. From Competition to Complementarity: Comparative Influence Diffusion and Maximization. *Proc. VLDB Endow.* 9, 2 (2015), 60–71.

[36] R Duncan Luce and Albert D Perry. 1949. A method of matrix analysis of group structure. *Psychometrika* 14, 2 (1949), 95–116.

[37] Yasir Mehmood, Francesco Bonchi, and David García-Soriano. 2016. Spheres of influence for more effective viral marketing. In *SIGMOD*. 711–726.

[38] Huiyu Min, Jiuxin Cao, Tangfei Yuan, and Bo Liu. 2020. Topic based time-sensitive influence maximization in online social networks. *World Wide Web* 23, 3 (2020), 1831–1859.

[39] George L Nemhauser and Laurence A Wolsey. 1981. Maximizing submodular set functions: formulations and analysis of algorithms. In *North-Holland Mathematics Studies*. Vol. 59. Elsevier, 279–301.

[40] Hung T Nguyen, My T Thai, and Thang N Dinh. 2016. Stop-and-stare: Optimal sampling algorithms for viral marketing in billion-scale networks. In *SIGMOD*. 695–710.

[41] Qiufen Ni, Jianxiong Guo, Weili Wu, and Huan Wang. 2022. Influence-based community partition with sandwich method for social networks. *IEEE Transactions on Computational Social Systems* (2022).

[42] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *nature* 435, 7043 (2005), 814–818.

[43] Benedek Rozemberczki and Rik Sarkar. 2021. Twitch Gamers: a Dataset for Evaluating Proximity Preserving and Structural Role-based Node Embeddings. arXiv:2101.03091 [cs.SI]

[44] Alexander J Stewart, Mohsen Mosleh, Marina Diakonova, Antonio A Arechar, David G Rand, and Joshua B Plotkin. 2019. Information gerrymandering and undemocratic decisions. *Nature* 573, 7772 (2019), 117–121.

[45] Lubos Takac and Michal Zabovsky. 2012. Data analysis in public social networks. In *International scientific conference and international workshop present day trends of innovations*, Vol. 1. Present Day Trends of Innovations Lamza Poland.

[46] Jing Tang, Xueyan Tang, Xiaokui Xiao, and Junsong Yuan. 2018. Online processing algorithms for influence maximization. In *SIGMOD*. 991–1005.

[47] Youze Tang, Yanchen Shi, and Xiaokui Xiao. 2015. Influence maximization in near-linear time: A martingale approach. In *SIGMOD*. 1539–1554.

[48] Youze Tang, Xiaokui Xiao, and Yanchen Shi. 2014. Influence maximization: Near-optimal time complexity meets practical efficiency. In *SIGMOD*. 75–86.

[49] Shan Tian, Songsong Mo, Liwei Wang, and Zhiyong Peng. 2020. Deep reinforcement learning-based approach to tackle topic-aware influence maximization. *Data Science and Engineering* 5, 1 (2020), 1–11.

[50] Dimitris Tsaras, George Trimponias, Lefteris Ntaflos, and Dimitris Papadias. 2021. Collective influence maximization for multiple competing products with an awareness-to-influence model. *Proc. VLDB Endow.* 14, 7 (2021), 1124–1136.

[51] Duru Türkoglu and Ata Turk. 2017. Edge-Based Wedge Sampling to Estimate Triangle Counts in Very Large Graphs. In *2017 IEEE International Conference on Data Mining*. IEEE Computer Society, 455–464.

[52] Pinghui Wang, Yiyan Qi, Yu Sun, Xiangliang Zhang, Jing Tao, and Xiaohong Guan. 2017. Approximately Counting Triangles in Large Graph Streams Including Edge Duplicates with a Fixed Memory Usage. *Proc. VLDB Endow.* 11, 2 (2017), 162–175.

[53] Xiaoyang Wang, Ying Zhang, Wenjie Zhang, and Xuemin Lin. 2016. Efficient distance-aware influence maximization in geo-social networks. *TKDE* 29, 3 (2016), 599–612.

[54] Yu Wang, Gao Cong, Guojie Song, and Kunqing Xie. 2010. Community-based greedy algorithm for mining top-K influential nodes in mobile social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 1039–1048.

[55] Yaxuan Wang, Hongzhi Wang, Jianzhong Li, and Hong Gao. 2016. Efficient Influence Maximization in Weighted Independent Cascade Model. In *Database Systems for Advanced Applications - 21st International Conference (Lecture Notes in Computer Science)*, Vol. 9643. Springer, 49–64.

[56] Zhefeng Wang, Yu Yang, Jian Pei, Lingyang Chu, and Enhong Chen. 2017. Activity maximization by effective information diffusion in social networks. *IEEE Transactions on Knowledge and Data Engineering* 29, 11 (2017), 2374–2387.

[57] Stanley Wasserman, Katherine Faust, et al. 1994. Social network analysis: Methods and applications. (1994).

[58] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world'networks. *nature* 393, 6684 (1998), 440–442.

[59] Xiaowei Ye, Rong-Hua Li, Qiangqiang Dai, Hongzhi Chen, and Guoren Wang. 2022. Lightning Fast and Space Efficient K-Clique Counting. In *Proceedings of the ACM Web Conference 2022*. Association for Computing Machinery, 1191–1202.

[60] Kem ZK Zhang and Morad Benyoucef. 2016. Consumer behavior in social commerce: A literature review. *Decision support systems* 86 (2016), 95–108.

[61] Xinxin Zhang, Li Xu, and Zhenyu Xu. 2022. Influence Maximization Based on Network Motifs in Mobile Social Networks. *IEEE Trans. Netw. Sci. Eng.* 9, 4 (2022), 2353–2363. https://doi.org/10.1109/TNSE.2022.3163203

[62] Tong Zhao, Julian McAuley, and Irwin King. 2015. Improving latent factor models via personalized feature projection for one class recommendation. In *Proceedings of the 24th ACM international on conference on information and knowledge management*. 821–830.

[63] Yuting Zhong and Longkun Guo. 2020. Group Influence Maximization in Social Networks. In *CSoNet*. 152–163.

[64] Jianming Zhu, Smita Ghosh, and Weili Wu. 2019. Group influence maximization problem in social networks. *TCSS* 6, 6 (2019), 1156–1164.