# Stop-and-Stare: Optimal Sampling Algorithms for Viral Marketing in Billion-scale Networks

Hung T. Nguyen
CS Department
Virginia Commonwealth Univ.
Richmond, VA, USA
hungnt@vcu.edu

My T. Thai
CISE Department
University of Florida
Gainesville, Florida, USA
mythai@cise.ufl.edu

Thang N. Dinh
CS Department
Virginia Commonwealth Univ.
Richmond, VA, USA
tndinh@vcu.edu

## ABSTRACT

*Influence Maximization* (IM), that seeks a small set of key users who spread the influence widely into the network, is a core problem in multiple domains. It finds applications in viral marketing, epidemic control, and assessing cascading failures within complex systems. Despite the huge amount of effort, IM in billion-scale networks such as Facebook, Twitter, and World Wide Web has not been satisfactorily solved. Even the state-of-the-art methods such as TIM+ and IMM may take days on those networks.

In this paper, we propose SSA and D-SSA, two novel sampling frameworks for IM-based viral marketing problems. SSA and D-SSA are up to 1200 times faster than the SIGMOD'15 best method, IMM, while providing the same $(1-1/e-\epsilon)$ approximation guarantee. Underlying our frameworks is an innovative *Stop-and-Stare* strategy in which they *stop at exponential check points* to verify (*stare*) if there is adequate statistical evidence on the solution quality. Theoretically, we prove that SSA and D-SSA are the first approximation algorithms that use (asymptotically) minimum numbers of samples, meeting strict theoretical thresholds characterized for IM. The absolute superiority of SSA and D-SSA are confirmed through extensive experiments on real network data for IM and another topic-aware viral marketing problem, named TVM.

## Keywords

Influence Maximization; Stop-and-Stare; Sampling

## 1. INTRODUCTION

Viral Marketing, in which brand-awareness information is widely spread via the word-of-mouth effect, has emerged as one of the most effective marketing channels. It is becoming even more attractive with the explosion of social networking services such as Facebook[1] with 1.5 billion monthly active users or Instagram[2] with more than 3.5 billion daily like connections. To create a successful viral marketing campaign, one needs to seed the content with a set of individuals with high social networking influence. Finding such a set of users is known as the *Influence Maximization* problem.

Given a network and a budget $k$, Influence Maximization (IM) asks for $k$ influential users who can spread the influence widely into the network. Kempe et al. [1] were the first to formulate IM as a combinatorial optimization problem on the two pioneering diffusion models, namely, *Independent Cascade* (IC) and *Linear Threshold* (LT). They prove IM to be NP-hard and provide a natural greedy algorithm that yields $(1-1/e-\epsilon)$-approximate solutions for any $\epsilon > 0$. This celebrated work has motivated a vast amount of work on IM in the past decade [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. However, most of the existing methods either too slow for billion-scale networks [1, 2, 4, 5, 6, 7] or ad-hoc heuristics without performance guarantees [13, 3, 14, 15].

The most scalable methods with performance guarantee for IM are TIM/TIM+[8] and latter IMM[16]. They utilize a novel RIS sampling technique introduced by Borgs et al. in [17]. All these methods attempt to generate a $(1-1/e-\epsilon)$ approximate solution with minimal numbers of RIS samples. They use highly sophisticated estimating methods to make the number of RIS samples close to some theoretical thresholds $\theta$ [8, 16]. However, they all share two shortcomings: 1) the number of generated samples can be arbitrarily larger than $\theta$, and 2) the thresholds $\theta$ are not shown to be the minimum among their kinds.

In this paper, we 1) unify the approaches in [17, 8, 16] to characterize the necessary number of RIS samples to achieve $(1-1/e-\epsilon)$-approximation guarantee; 2) design two novel sampling algorithms SSA and D-SSA aiming towards achieving minimum number of RIS samples. In the first part, we begin with defining RIS framework which consists of two necessary conditions to achieve the $(1-1/e-\epsilon)$ factor and classes of *RIS thresholds* on the sufficient numbers of RIS samples, generalizing $\theta$ thresholds in [8, 16]. The minimum threshold in each class is then termed *type-1 minimum threshold*, and the minimum among all type-1 minimum thresholds is termed *type-2 minimum threshold.*

In the second part, we develop the *Stop-and-Stare Algorithm* (SSA) and its dynamic version D-SSA that guarantee to achieve, within constant factors, the two minimum thresholds, respectively. Both SSA and D-SSA follow the *stop-and-stare* strategy which can be efficiently applied to many optimization problems over the samples and guaran-

[1] http://newsroom.fb.com/company-info/

[2] https://instagram.com/press/



*SIGMOD'16, June 26-July 01, 2016, San Francisco, CA, USA*
© 2016 ACM. ISBN 978-1-4503-3531-7/16/06...$15.00
DOI: http://dx.doi.org/10.1145/2882903.2915207

tee some constant times the minimum number of samples required. In short, the algorithms keep generating samples and *stop* at *exponential check points* to verify (*stare*) if there is adequate statistical evidence on the solution quality for termination. This strategy will be shown to address both of the shortcomings in [8, 16]: 1) guarantee to be close to the theoretical thresholds and 2) the thresholds are minimal by definitions. The dynamic algorithm, D-SSA, improves over SSA by automatically and dynamically selecting the best parameters for the RIS framework. We note that the Stop-and-Stare strategy combined with RIS framework enables SSA and D-SSA to meet the minimum thresholds without explicitly computing/looking for these thresholds. That is in contrast to previous approaches [17, 8, 16] which all find some explicit unreachable thresholds and then probe for them with unbounded or huge gaps.

Our experiments show that both SSA and D-SSA outperform the best existing methods up to several orders of magnitudes w.r.t running time while returning comparable seed set quality. More specifically, on Friendster network with roughly 65.6 million nodes and 1.8 billion edges, SSA and D-SSA, taking 3.5 seconds when $k = 500$, are up to 1200 times faster than IMM. We also run CELF++ (the fastest greedy algorithm for IM with guarantees) on Twitter network with $k = 1000$ and observe that D-SSA is $2 \cdot 10^9$ times faster. Our contributions are summarized as follows.

- We generalize the RIS sampling methods in [17, 8, 16] into a general framework which characterizes the necessary conditions to guarantee the $(1 - 1/e - \epsilon)$-approximation factor. Based on the framework, we define classes of RIS thresholds and two types of minimum thresholds, namely, type-1 and type-2.
- We propose the Stop-and-Stare Algorithm (SSA) and its dynamic version, D-SSA, which both guarantee a $(1-1/e-\epsilon)$-approximate solution and are the first algorithms to achieve, within constant factors, the type-1 and type-2 minimum thresholds, respectively. Our proposed methods are not limited to solve influence maximization problem but also can be generalized for an important class of hard optimization problems over samples/sketches.
- Our framework and approaches are generic and can be applied in principle to sample-based optimization problems to design high-confidence approximation algorithm using (asymptotically) *minimum number of samples*.
- We carry extensive experiments on various real networks with up to several billion edges to show the superiority in performance and comparable solution quality. To test the applicability of the proposed algorithms, we apply our methods on an IM-application, namely, Targeted Viral Marketing (TVM). The results show that our algorithms are up to 1200 times faster than the current best method on IM problem and, for TVM, the speedup is up to 500 times.

Note that this paper does not focus on distributed/parallel computation, however our algorithms are amenable to a distributed implementation which is one of our future works.

**Related works.** Kempe et al. [1] formulated the influence maximization problem as an optimization problem. They show the problem to be NP-complete and devise an $(1 - 1/e - \epsilon)$ greedy algorithm. Later, computing the exact influence is shown to be #P-hard [3]. Leskovec et al. [2] study the influence propagation in a different perspective in which they aim to find a set of nodes in networks to detect the spread of virus as soon as possible. They improve the simple greedy method with the lazy-forward heuristic (CELF), which is originally proposed to optimize submodular functions in [18], obtaining an (up to) 700-fold speedup.

Several heuristics are developed to find solutions in large networks. While those heuristics are often faster in practice, they fail to retain the $(1-1/e-\epsilon)$-approximation guarantee and produce lower quality seed sets. Chen et al. [19] obtain a speedup by using an influence estimation for the IC model. For the LT model, Chen et al. [3] propose to use local directed acyclic graphs (LDAG) to approximate the influence regions of nodes. In a complement direction, there are recent works on learning the parameters of influence propagation models [20, 21]. The influence maximization is also studied in other diffusion models including the majority threshold model [22] or when both positive and negative influences are considered [23] and when the propagation terminates after a predefined time [22, 24]. Recently, IM across multiple OSNs have been studied in [11] and [25] studies the IM problem on continuous-time diffusion models.

Recently, Borgs et al. [17] make a theoretical breakthrough and present an $O(kl^2(m+n)\log^2 n/\epsilon^3)$ time algorithm for IM under IC model. Their algorithm (RIS) returns a $(1-1/e-\epsilon)$-approximate solution with probability at least $1 - n^{-l}$. In practice, the proposed algorithm is, however, less than satisfactory due to the rather large hidden constants. In sequential works, Tang et al. [8, 16] reduce the running time to $O((k + l)(m + n)\log n/\epsilon^2)$ and show that their algorithm is also very efficient in large networks with billions of edges. Nevertheless, Tang's algorithms have two weaknesses: 1) intractable estimation of maximum influence and 2) taking union bounds over all possible seed sets in order to guarantee a single returned set.

**Organization.** The rest of the paper is organized as follows: In Section 2, we introduce two fundamental models, i.e., LT and IC, and the IM problem definition. We, subsequently, devise the unified RIS framework, RIS threshold and two types of RIS minimum thresholds in Section 3. Section 4 and 5 will present the SSA algorithm and prove the approximation factor as well as the achievement of type-1 minimum threshold. In Section 6, we propose the dynamic algorithm, D-SSA and prove the approximation together with type-2 minimum threshold property. Finally, we show experimental results in Section 7 and draw some conclusion in Section 8.

## 2. MODELS AND PROBLEM DEFINITION

This section will formally define two most essential propagation models, e.i., *Linear Threshold* (LT) and *Independent Cascade* (IC), that we consider in this work and followed by the problem statement of the Influence Maximization (IM).

We abstract a network using a weighted graph $G = (V, E, w)$ with $|V| = n$ nodes and $|E| = m$ directed edges. Each edge $(u, v) \in E$ is associated with a weight $w(u, v) \in [0, 1]$ which indicates the probability that $u$ influences $v$.

### 2.1 Propagation Models

In this paper, we study two fundamental diffusion models, namely, Linear Threshold (LT) and Independent Cascade

(IC). Assume that we have a set of seed nodes $S$, the propagation processes under these two models happen in rounds. At round 0, all nodes in $S$ are activated and the others are not activated. In the subsequent rounds, the newly activated nodes will try to activate their neighbors. Once a node $v$ becomes active, it will remain active till the end. The process stops when no more nodes get activated. The distinctions of the two models are described as follows:

*Linear Threshold (LT) model.* The edge weights in LT model must satisfy the condition $\sum_{u \in V} w(u, v) \leq 1$. At the beginning of the propagation process, each node $v$ selects a random threshold $\lambda_v$ uniformly at random in range $[0, 1]$. In round $t \geq 1$, an inactivated node $v$ becomes activated if $\sum_{\textbf{activated neighbors } u} w(u, v) \geq \lambda_v$. Let $\mathbb{I}(S)$ denote the expected number of activated nodes given the seed set $S$, where the expectation is taken over all $\lambda_v$ values from their uniform distribution. We call $\mathbb{I}(S)$ the *influence spread* of $S$ under the LT model.

*Independent Cascade (IC) model.* In IC, when a node $u$ gets activated, initially or by another node, it has a single chance to activate each inactive neighbor $v$ with the probability proportional to the edge weight $w(u, v)$. After that moment, the activated nodes remain its active state but they have no contribution in later activations.

Table 1: Table of Symbols

| Notation | Description |
|---|---|
| $n, m$ | #nodes, #edges of graph $\mathcal{G} = (V, E, w)$. |
| $\mathbb{I}(S)$ | Influence Spread of seed set $S \subseteq V$. |
| $OPT_k$ | The maximum $\mathbb{I}(S)$ for any size-$k$ seed set $S$. |
| $\hat{S}_k$ | The returned size-$k$ seed set of SSA/D-SSA. |
| $S_k^*$ | An optimal size-$k$ seed set, i.e., $\mathbb{I}(S_k^*) = OPT_k$. |
| $R_j$ | A random RR set. |
| $\mathcal{R}$ | A set of random RR sets. |
| $\mathrm{Cov}_{\mathcal{R}}(S)$, $S \subseteq V$ | #RR sets in $\mathcal{R}$ incident at some node in $S$. |
| $c$ | $c = 2(e-2) \approx \sqrt{2}$. |
| $\Upsilon(\epsilon, \delta)$ | $\Upsilon(\epsilon, \delta) = 2c \ln \frac{1}{\delta} \frac{1}{\epsilon^2}$. |
| $\Lambda_1$ | $\Lambda_1 = (1 + \epsilon_1)(1 + \epsilon_2)\Upsilon(\epsilon_3, \delta/3)$. |
| $\Lambda_2$ | $\Lambda_2 = (1 + \epsilon_2)\Upsilon(\epsilon_2, \delta_2)$. |

## 2.2 Problem Definition

Given the propagation models defined previously, we formally state the Influence Maximization (IM) problem as in the following definition,

Definition 1 (Influence Maximization (IM)). *Given a graph $G = (V, E, w)$, $k \in Z^+$ and a propagation model, the Influence Maximization problem asks for a seed set $\hat{S}_k \subset V$ of $k$ nodes that maximizes its influence spread, $\mathbb{I}(\hat{S}_k)$.*

# 3. UNIFIED RIS FRAMEWORK

This section will present the unified RIS framework which generalizes the methods of using RIS sampling for IM problem. The unified framework characterizes the sufficient conditions to guarantee an $(1 - 1/e - \epsilon)$-approximation in the framework. Subsequently, we will introduce the concept of RIS threshold in terms of the number of necessary samples to guarantee the solution quality and two types of minimum RIS thresholds, i.e., type-1 and type-2.

## 3.1 Preliminaries

### 3.1.1 RIS sampling

The major bottle-neck in the traditional methods for IM [1, 2, 4, 26] is the inefficiency in estimating the influence spread. To address that, Borgs et al. [17] introduced a novel sampling approach for IM, called Reverse Influence Sampling (in short, RIS), which is the foundation for TIM/TIM+[8] and IMM[16], the state-of-the-art methods.

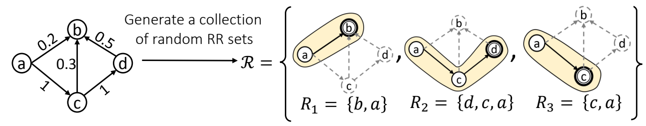



Figure 1: An example of generating random RR sets under the LT model. Three random RR sets $R_1, R_2$ and $R_3$ are generated. Node $a$ has the highest influence and is also the most frequent element across the RR sets.

Given a graph $G = (V, E, w)$, RIS captures the influence landscape of $G$ through generating a set $\mathcal{R}$ of random *Reverse Reachable (RR) sets*. The term 'RR set' is also used in TIM/TIM+ [8, 16] and referred to as 'hyperedge' in [17]. Each RR set $R_j$ is a subset of $V$ and constructed as follows,

Definition 2 (Reverse Reachable (RR) set). *Given $G = (V, E, w)$, a random RR set $R_j$ is generated from $G$ by 1) selecting a random node $v \in V$ 2) generating a sample graph $g$ from $G$ and 3) returning $R_j$ as the set of nodes that can reach $v$ in $g$.*

Node $v$ in the above definition is called the *source* of $R_j$. Observe that $R_j$ contains the nodes that can influence its source $v$. If we generate multiple random RR sets, influential nodes will likely appear frequently in the RR sets. Thus a seed set $S$ that *covers* most of the RR sets will likely maximize the influence spread $\mathbb{I}(S)$. Here a seed set $S$ covers an RR set $R_j$, if $S \cap R_j \neq \emptyset$. For convenience, we denote the coverage of set $S$ as $\mathrm{Cov}_{\mathcal{R}}(S) = \sum_{R_j \in \mathcal{R}} \min\{|S \cap R_j|, 1\}$. An illustration of this intuition and how to generate RR sets is given in Fig. 1. In the figure, three random RR sets are generated following the LT model with sources $b$, $d$ and $c$, respectively. The influence of node $a$ is the highest among all the nodes in the original graph and also is the most frequent node across the RR sets. This observation is rigorously captured in the following lemma in [17].

Lemma 1. *Given $G = (V, E, w)$ and a random RR set $R_j$ generated from $G$. For each seed set $S \subset V$,*

$$\mathbb{I}(S) = n \Pr[S \text{ covers } R_j]. \tag{1}$$

Lemma 1 says that the influence of a node set $S$ is proportional to the probability that $S$ intersects with a random RR set. Thus, to find $S$ that optimizes $\mathbb{I}(S)$, we can alternatively optimize over the proportional probability which can be approximated by sampling many $R_j$. This relation is the soul of the RIS-based approaches on IM problem.

### 3.1.2 $(\epsilon, \delta)$-approximation

We recall the $(\epsilon, \delta)$-approximation and two criteria of achieving such approximation in [27] that will be used in our framework and later proofs.

Definition 3 ($(\epsilon, \delta)$-approximation). *Let $Z_1, Z_2, \ldots$ be independently and identically distributed samples according* *to Z in the interval [0, 1] with mean $\mu_Z$ and variance $\sigma_Z^2$. A Monte Carlo estimator of $\mu_Z$,*

$$\hat{\mu}_Z = \frac{1}{T}\sum_{i=1}^{T} Z_i \tag{2}$$

*is said to be an $(\epsilon, \delta)$-approximation of $\mu_Z$ if*

$$\Pr[(1-\epsilon)\mu_Z \leq \hat{\mu}_Z \leq (1+\epsilon)\mu_Z] \geq 1-\delta \tag{3}$$

Having defined the $(\epsilon, \delta)$-approximation, the first inherent question is how (when) we can achieve that approximation. We investigate two criteria in Lems. 2 and 3.

Define $\Upsilon(\epsilon, \delta/2) = 2c\ln(2/\delta)/\epsilon^2$ with $c = 2(e-2)$ and $\mathrm{Cov}(Z) = \sum_{i=1}^{T} Z_i$. The following lemma is called *Zero-One Estimator Theorem* [27] stating the first criterion.

LEMMA 2 ([27]). *Let $\theta^*$ be the optimal number samples that guarantee an $(\epsilon, \delta)$-approximation of $\mu_Z$, if $T = \frac{\Upsilon(\epsilon,\delta/2)}{\mu_Z}$, then $\hat{\mu}_Z = \frac{\mathrm{Cov}(Z)}{T}$ is an $(\epsilon, \delta)$-approximate of $\mu_Z$ and*

$$T \leq \mathsf{c}_1 \cdot \theta^* \tag{4}$$

*where $\mathsf{c}_1$ is a constant.*

Based also on [27], we have the second criterion to guarantee $(\epsilon, \delta)$-approximation as stated below.

LEMMA 3 ([27]). *Let $\theta^*, \mu_Z$ be defined as in Lem. 2 and $T$ be the number of samples at which $\mathrm{Cov}(Z) \geq 1 + (1+\epsilon)\Upsilon(\epsilon, \delta/2)$, then $\hat{\mu}_Z$ is an $(\epsilon, \delta)$-approximation of $\mu_Z$ and $T \leq (1+\epsilon)\mathsf{c}_1 \cdot \theta^*$ where $\mathsf{c}_1$ is the constant in Lem. 2.*

Note that if only one side of the event in Eq. 3 is required, then $\Upsilon(\epsilon, \delta/2)$ becomes

$$\Upsilon(\epsilon, \delta) = 2c\ln\frac{1}{\delta}\frac{1}{\epsilon^2} \tag{5}$$

## 3.2 RIS Framework and Thresholds

Based on Lem. 1, the IM problem can be solved by the following two step algorithm.

- Generate a collection of RR sets, $\mathcal{R}$, from $G$.
- Use the greedy algorithm for the Max-coverage problem [28] to find a seed set $\hat{S}_k$ that covers the maximum number of RR sets and return $\hat{S}_k$ as the solution.

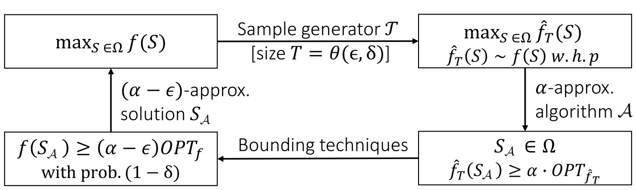



Figure 2: Overview of algorithms based on optimization over samples ($\epsilon$ is the error from approximating $f(S)$ by $\hat{f}_T(S)$, $OPT_f$ and $OPT_{\hat{f}_T}$ are optimal solutions of $f(.)$ and $\hat{f}_T(.)$).

This two-step algorithm is actually an instance of a general class of methods illustrated in Fig. 2. The original problem is an maximization problem of $f(.)$ over $\Omega_S$ which is usually very hard to solve/approximate directly. Instead, we find an estimate $\hat{f}_T(.)$ of $f(.)$. The estimation function $\hat{f}_T(.)$ is constructed by generating $T = \theta(\epsilon, \delta)$ samples where $\theta(\epsilon, \delta)$ is an **explicit** threshold. The threshold $\theta(\epsilon, \delta)$ decides the estimation quality of $\hat{f}_T(.)$ compared to $f(.)$ and usually is the most critical point in the methods. After having the function $\hat{f}_T(.)$, an $\alpha-$approximation algorithm $\mathcal{A}$ which is easier and efficient to find the final solution $S_\mathcal{A}$ of $\hat{f}_T(.)$ as well as $f(.)$. The function $f(.)$ that characterizes our maximization objective covers a wide range of important problems, e.g., targeted viral marketing [29], densest subgraph [30], which have very high complexity to approximate and hence, need to rely on sampling algorithms. Thus, our proposed *stop-and-stare* methods can be modified to address many other optimization problems under this category. We illustrate this ability by extending SSA and D-SSA to solve the targeted viral marketing in our experiments section.

Similar to determining $\theta$, the core question in applying the above algorithm for influence maximization problem is that: *How many RR sets are sufficient to provide a good approximate solution?* In this case, the function $f(S)$ is the influence function $\mathbb{I}(S)$ and the samples are generated by RIS sampling. [8, 16] propose two such theoretical thresholds and two probing techniques to realistically estimate those thresholds. However, their thresholds are not known to be any kind of minimum and the probing method is *ad hoc* in [8] or far from the proposed threshold in [16]. Thus, they cannot provide any guarantee on the optimality of the number of samples generated.

Since the probing for an explicit threshold is seen to admit certain limitations, we propose a new approach. Instead of explicitly expressing a theoretical threshold and trying to probe for it, we characterize the conditions that all the RIS-based algorithms need to attain and state the sufficient number of samples to satisfy those conditions. Thus, we can define the minimum samples according to the necessary conditions and further, propose SSA and D-SSA to achieve the minimum. We will first define our RIS framework as enforcing the necessary conditions to guarantee the best known solution quality and then propose two minimum thresholds based on the precision parameters in our framework.

Suppose that there is an optimal seed set, $S_k^*$, which has the maximum influence in the network. If there are multiple optimal sets with influence, $OPT_k$, we choose the first one alphabetically to be $S_k^*$. Given $0 \leq \epsilon, \delta \leq 1$, our unified RIS framework enforces two conditions:

$$\Pr[\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_a)\mathbb{I}(\hat{S}_k)] \geq 1-\delta_a \tag{6}$$

and

$$\Pr[\hat{\mathbb{I}}(S_k^*) \geq (1-\epsilon_b)OPT_k] \geq 1-\delta_b \tag{7}$$

where $\delta_a + \delta_b \leq \delta$ and $\epsilon_a + (1-1/e)\epsilon_b \leq \epsilon$.

Based on the above conditions, we define the RIS threshold as the following.

DEFINITION 4 (RIS THRESHOLD). *Given a graph $G$, $0 \leq \epsilon_a, \epsilon_b, \delta_a, \delta_b \leq 1$, $N(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ is called an RIS Threshold in $G$ w.r.t $\epsilon_a, \epsilon_b, \delta_a, \delta_b$ if any number $|\mathcal{R}| \geq N(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ of random RR sets generated from $G$ is sufficient to guarantee both Eqs. 6 and 7.*

With the two above conditions, we now prove that any $N \geq N(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets are sufficient to guarantee $(1-1/e-\epsilon)$-approximation ratio.

THEOREM 1. *Given a graph $G$, $0 \leq \epsilon_a, \epsilon_b, \delta_a, \delta_b \leq 1$, let $\epsilon \geq \epsilon_a + (1-1/e)\epsilon_b$ and $\delta \geq \delta_a + \delta_b$, if the number of RR sets $|\mathcal{R}| \geq N(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$, then the two-step algorithm in our RIS framework returns $\hat{S}_k$ satisfying*

$$\Pr[\mathbb{I}(\hat{S}_k) \geq (1 - 1/e - \epsilon) OPT_k] \geq 1 - \delta \quad (8)$$

*which means $\hat{S}_k$ is an $(1 - 1/e - \epsilon)$-approximate solution.*

For brevity, the proof is presented in the appendix.

**Existing RIS thresholds.** For any $\epsilon, \delta \in (0, 1)$, Tang et al. established in [8] an RIS threshold,

$$N(\frac{\epsilon}{2}, \frac{\epsilon}{2}, \frac{\delta}{2\binom{n}{k}}, \frac{\delta}{2\binom{n}{k}}) = (8 + 2\epsilon) n \frac{\ln 2/\delta + \ln \binom{n}{k}}{\epsilon^2 OPT_k}$$

In a later study [16], they reduced this number to another RIS threshold,

$$N(\epsilon_1, \epsilon - \epsilon_1, \frac{\delta}{2\binom{n}{k}}, \frac{\delta}{2\binom{n}{k}}) = 2n \frac{((1 - 1/e)\alpha + \beta)^2}{\epsilon^2 OPT_k},$$

where $\alpha = (\frac{\ln 2}{\delta})^{\frac{1}{2}}$, $\beta = (1 - 1/e)^{\frac{1}{2}} (\frac{\ln 2}{\delta} + \ln \binom{n}{k})^{\frac{1}{2}}$ and $\epsilon_1 = \frac{\epsilon \cdot \alpha}{(1 - 1/e)\alpha + \beta}$.

Unfortunately, computing $OPT_k$ is intractable, thus, the proposed algorithms have to generate $\theta \frac{OPT_k}{KPT^+}$ RR sets, where $KPT^+$ is the expected influence of a node set obtained by sampling k nodes with replacement from $G$ and the ratio $\frac{OPT_k}{KPT^+} \geq 1$ is not upper-bounded. That is they may generate many times more RR sets than needed.

## 3.3 Two Types of Minimum Thresholds

Based on the definition of RIS threshold, we now define two strong theoretical limits, i.e. type-1 minimum and type-2 minimum thresholds. In Section 5, we will prove that our first proposed algorithm, SSA, achieves, within a constant factor, a type-1 minimum threshold and later, in Section 6, our dynamic algorithm, D-SSA, is shown to obtain, within a constant factor, the strongest type-2 minimum threshold.

Definition 5 (type-1 minimum threshold). *Given $0 \leq \epsilon, \delta \leq 1$ and $0 \leq \epsilon_a, \epsilon_b, \delta_a, \delta_b \leq 1$ satisfying $\delta_a + \delta_b \leq \delta$ and $\epsilon_a + (1 - 1/e)\epsilon_b \leq \epsilon$, $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ is called a type-1 minimum threshold w.r.t $\epsilon_a, \epsilon_b, \delta_a, \delta_b$ if $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ is the smallest number of RR sets that satisfies both Eq. 6 and Eq. 7.*

If $N(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ is an RIS threshold, then any $N$ such that $N \geq N(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ is also an RIS threshold. We choose the smallest number over all the RIS thresholds to be type-1 minimum as defined in Def. 5. All the previous methods [17, 8, 16] try to approximate $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ for some setting of $\epsilon_a, \epsilon_b, \delta_a, \delta_b$, however, they fail to provide any guarantee on how close their numbers are to that threshold. In contrast, we show that SSA achieves, within a constant factor, an type-1 minimum threshold in Sec. 5. Next, we give the definition of the strongest type-2 minimum threshold which is achieved by D-SSA as shown in Sec. 6.

Definition 6 (type-2 minimum threshold). *Given $0 \leq \epsilon, \delta \leq 1$, $N^2_{min}(\epsilon, \delta)$ is called the type-2 minimum threshold if*

$$N^2_{min}(\epsilon, \delta) = \min_{\epsilon_a, \epsilon_b, \delta_a, \delta_b} N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b) \quad (9)$$

*where $\epsilon_a + (1 - 1/e)\epsilon_b = \epsilon$ and $\delta_a + \delta_b = \delta$.*

From Def. 6, it follows that type-2 minimum is the strongest possible threshold that we can achieve in the RIS-framework.

## 3.4 Achieving the Minimum Thresholds

In the following sections, we propose two approximation algorithms, namely, *Stop-and-Stare* (SSA) and the dynamic version D-SSA, which respectively achieve the two theoretical minimum thresholds as well as the best known worst-case approximation ratio. In more details, both SSA and D-SSA employ the stop-and-stare strategy that doubles the number of RIS samples and checks the quality of current solution by an independent influence estimation step. This strategy guarantees that we do not oversample, i.e., doubling the necessary number in the worst case. On a specific setting of the tuple $\{\epsilon_a, \epsilon_b, \delta_a, \delta_b\}$, SSA guarantees a type-1 minimum threshold corresponding to that configuration.

By specifically setting the parameters, SSA algorithm is simpler than the dynamic D-SSA. D-SSA achieves the type-2 minimum threshold through dynamically finding the best set of parameter values at each exponential checking points. Thus, $\{\epsilon_a, \epsilon_b, \delta_a, \delta_b\}$ are not specified in advance but automatically detected by D-SSA. Further, D-SSA can reuse the RR sets generated in the independent influence estimation without changing the independence property of RR sets in subsequent iterations.

# 4. STOP-AND-STARE ALGORITHM (SSA)

In this section, we describe our first Stop-and-Stare Algorithm (SSA) in details. SSA keeps generating RR sets until doubling the current number and *stops* to check the solution obtained from the total RR sets generated. It uses Max-Coverage (Alg. 2) to find the solution, $\hat{S}_k$. Since the influence of $\hat{S}_k$ calculated from those RR sets is biased, SSA independently generates another collection of RR sets in Estimate-Inf procedure (Alg. 3) to obtain an unbiased estimation of $\hat{S}_k$ influence. Then it *stares* at the two influences and if they are close enough (satisfying SSA's stopping conditions), it halts and returns the found solution.

**Algorithm 1** SSA Algorithm

**Input**: Graph $G$, $0 \leq \epsilon, \delta \leq 1$, and a budget $k$.
**Output**: An $(1 - 1/e - \epsilon)$-optimal solution, $\hat{S}_k$ with at least $(1 - \delta)$-probability.

Compute $\epsilon_1, \epsilon_2, \epsilon_3$ according to Eq. 18
$\Lambda_1 \leftarrow (1 + \epsilon_1)(1 + \epsilon_2) 2c \ln \frac{3}{\delta} \frac{1}{\epsilon_3^2}$
$\mathcal{R} \leftarrow$ Generate $\Lambda_1$ random RR sets by RIS
**repeat**
  $<\hat{S}_k, \hat{\mathbb{I}}(\hat{S}_k)> \leftarrow$ Max-Coverage($\mathcal{R}$, $k$, $n$)
  **if** $\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k) \geq \Lambda_1$ **then**
    $\mathbb{I}_c(\hat{S}_k) \leftarrow$ Estimate-Inf($G, \hat{S}_k, \epsilon_2, \frac{\delta}{3}, |\mathcal{R}| \frac{1+\epsilon_2}{1-\epsilon_2} \frac{\epsilon_3^2}{\epsilon_2^2}$)
    **if** $\hat{\mathbb{I}}(\hat{S}_k) \leq (1 + \epsilon_1)\mathbb{I}_c(\hat{S}_k)$ **then**
      **return** $\hat{S}_k$
    **end if**
  **end if**
  $\mathcal{R} \leftarrow$ Generate $|\mathcal{R}|$ random RR sets by RIS
**until** $|\mathcal{R}| \geq (8 + 2\epsilon) n \frac{\ln \frac{2}{\delta} + \ln \binom{n}{k}}{\epsilon^2}$
**return** $\hat{S}_k$

## 4.1 SSA Algorithm

This subsection will detail the main procedure of SSA where Max-Coverage (described in Alg. 2) and Estimate-Inf (described in Alg. 3) are incorporated. The precision parameters $\epsilon_1, \epsilon_2, \epsilon_3, \delta_1, \delta_2$ are specified in Eq. 18 and discussed in

Subsection 5.2.1. An illustration of those precision parameters in SSA (and also later for D-SSA) is provided in Fig. 3. This figure also demonstrate our bounding technique in our framework (Fig. 2): The overall error is accumulated from the sampling errors to estimate of $\mathbb{I}(S_k^*)$ and $\mathbb{I}(\hat{S}_k)$ and the approximation error of Max-Coverage to find the set $\hat{S}_k$.

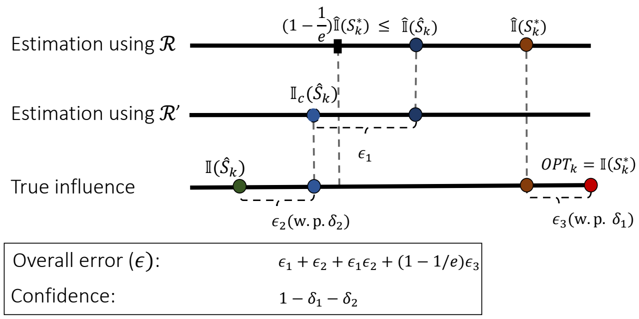



Figure 3: Illustration of precision parameters $\epsilon_1, \epsilon_2, \epsilon_3, \delta_1, \delta_2$ ($\mathbb{I}(.)$ denotes the true influence function, $\hat{\mathbb{I}}(.), \mathbb{I}_c(.)$ are the estimates of $\mathbb{I}(.)$ by the collection $\mathcal{R}$ of RR sets in the main algorithm and by another independent collection $\mathcal{R}'$ of the Estimate-Inf, $S_k^*$ is an optimal solution).

The SSA algorithm is presented in Alg. 1. SSA starts with initializing the values of three variables $\epsilon_1, \epsilon_2$ and $\epsilon_3$ which are critical in deciding the stopping conditions and will be specified in our analysis (Eq. 18) where we prove the type-1 minimum threshold of SSA. Having obtained those values, the algorithm, then, computes $\Lambda_1$ (Line 2) which determines the lower bound for the degree of the selected seed set, $\hat{S}_k$. The central part of our algorithm iterates round by round: at round 0, SSA generates $\Lambda_1$ random RR sets (Line 3) simply because that is the lowest number we can expect to meet the first condition at Line 6; at round $i \geq 1$, it doubles the number of random RR sets by introducing $|\mathcal{R}|$ more sets (Line 12). In each round, a candidate seed set $\hat{S}_k$ is selected by Max-Coverage together with its estimated influence. SSA obtains another estimated influence which is returned by the Estimate-Inf (Line 7) and denoted by $\mathbb{I}_c(\hat{S}_k)$. The two influences are used in the second condition (Line 8). Thus, SSA contains two *stopping conditions*:

(C1) The first condition $\text{Cov}_{\mathcal{R}}(\hat{S}_k) \geq \Lambda_1$ (Line 6) ensures that the coverage of the returned solution is at least $\Lambda_1$ which is important to guarantee the approximability of $\hat{S}_k$ and $S_k^*$ as shown in Lem.5 and 6. This condition remains true after the first time it is established.

(C2) The second condition $\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_1)\mathbb{I}_c(\hat{S}_k)$ (Line 8) compares estimates of $\mathbb{I}(\hat{S}_k)$ obtained by Max-Coverage and by the Estimate-Inf procedure. This comparison is only active after the first stopping condition is met and the number of RR sets is large enough to trigger the Estimate-Inf. If these two estimates are close enough (within a multiplicative factor of $1+\epsilon_1$), we confirm the approximability of $\hat{S}_k$ (Lem. 5).

As we will prove in Sec. 5, the two stopping conditions are sufficient to guarantee the $(1-1/e-\epsilon)$-approximation of $\hat{S}_k$. Note that we use a threshold $(8+2\epsilon)n\frac{\ln\frac{2}{\delta}+\ln\binom{n}{k}}{\epsilon^2}$ which is taken from TIM+ [8] when considering $OPT_k = 1$ to stop SSA in cases of bad events.

**Algorithm 2** Max-Coverage procedure

**Input:** RR sets ($\mathcal{R}$), $k$ and number of nodes ($n$).
**Output:** An $(1-1/e)$-optimal solution, $\hat{S}_k$ and its estimated influence $\mathbb{I}_c(\hat{S}_k)$.

1: $\hat{S}_k = \emptyset$
2: **for** i=1 **to** $k$ **do**
3: $\quad \hat{v} \leftarrow \arg\max_{\{v \in V\}}(\text{Cov}_{\mathcal{R}}(\hat{S}_k \cup \{v\}) - \text{Cov}_{\mathcal{R}}(\hat{S}_k))$
4: $\quad$ Add $\hat{v}$ to $\hat{S}_k$
5: **end for**
6: **return** $<\hat{S}_k, \text{Cov}_{\mathcal{R}}(\hat{S}_k) \cdot n/|\mathcal{R}|>$

## 4.2 Finding Max-Coverage

We describe the Max-Coverage procedure to find a $(1-1/e)$-coverage set. This algorithm plays the role of the $\alpha$-approximation algorithm in Fig. 2 where $\alpha = 1-1/e$. Alg. 2 illustrates the greedy Max-Coverage algorithm to select a size-$k$ seed set. The whole procedure goes through $k$ iterations in which, at each step, a node with maximum relative coverage, with respect to the previously chosen ones, is selected into the seed set $\hat{S}_k$. As a well-known result [31], we have the following lemma.

Lemma 4. *The greedy Max-Coverage returns an $(1-1/e)$-approximate seed set $\hat{S}_k$.*

This algorithm can be implemented in linear time in terms of the total size of all the RR sets as in [17, 8, 16] and, as a result, the complexity is upper-bounded by the generating RR sets. Thus, the complexity of the whole algorithm actually depends only on that for generating RR sets.

## 4.3 Influence Estimation

Estimating influence of a given seed set $S$ is a key component in our method and is used in the places where we have a candidate seed set and need to check whether the estimate of that set in the main algorithm is good enough. Our goal is to obtain a good approximation (within a certain multiplicative-error) of the given seed set influence.

**Algorithm 3** Estimate-Inf procedure

**Input**: Graph $G$, a seed set $S$, $0 \leq \epsilon_2, \delta_2 \leq 1$ and maximum number of samples, $T_{max}$.
**Output**: $\mathbb{I}_c(S)$ such that $\mathbb{I}_c(S) \leq (1+\epsilon_2)\mathbb{I}(S)$ with at least $(1-\delta_2)$-probability or exceeding $T_{max}$.

1: $\Lambda_2 = 1 + 2c(1+\epsilon_2)\ln\frac{1}{\delta_2}\frac{1}{\epsilon_2^2}$
2: $\text{Cov} = 0$
3: **for** $T$ **from** 1 **to** $T_{max}$
4: $\quad$ Generate $R_j \leftarrow$RIS$(G)$
5: $\quad \text{Cov} = \text{Cov} + \min\{|R_j \cap S|, 1\}$
6: $\quad$ **if** $\text{Cov} \geq \Lambda_2$ **then**
7: $\quad\quad$ **return** $n \times \text{Cov}/T$ {$n$: number of nodes}
8: $\quad$ **end if**
9: **end for**
10: **return** -1 {Exceeding $T_{max}$ RR sets}

The estimating procedure is called Estimate-Inf and detailed in Alg. 3. Given a node set $S$ and two parameters, $\epsilon_2, \delta_2$, Estimate-Inf returns an estimate $\mathbb{I}_c(S)$ of $\mathbb{I}(S)$ such that $\mathbb{I}_c(S) \leq (1+\epsilon_2)\mathbb{I}(S)$ with at least $(1-\epsilon_2)$-probability. The most crucial point in the procedure is determining $\Lambda_2$ (Line 1) and the related condition (Line 6) which compares the coverage of $S$ with $\Lambda_2$. In our analysis, we show that the condition is sufficient to guarantee $\mathbb{I}_c(S) \leq (1+\epsilon_2)\mathbb{I}(S)$ with at least $(1-\epsilon_2)$-probability. The main step in this procedure is the **for** loop in which random RR sets are drawn one at a time until satisfying the condition. The loop iterates through the maximum of $T_{max}$ iterations where, as described in Alg. 1, $T_{max}$ is a constant multiply of the total number of RR sets generated in the main algorithm at that point. Choosing $T_{max}$ is a crucial point of this algorithm since we run this at every iteration in SSA. At the beginning when the solution $\hat{S}$ is not

We define a random variable $X = \min\{|R_j \cap S|, 1\}$, where $R_j$ is a random RR set and $\mu_X = \mathbb{I}(S)/n$ (Lem. 1). From Lem. 3 of the $(\epsilon, \delta)$-approximation criteria, we obtain a direct corollary as stated below,

COROLLARY 1. *The Estimate-Inf procedure of SSA returns an estimate, $\mathbb{I}_c(S)$, of $\mathbb{I}(S)$ such that*

$$\Pr[\mathbb{I}_c(S) \leq (1+\epsilon_2)\mathbb{I}(S)] \geq 1-\delta_2 = 1-\delta/3 \qquad (10)$$

# 5. SSA GUARANTEE AND PERFORMANCE ANALYSIS

In this section, we will prove that SSA returns a $(1-1/e-\epsilon)$-approximate solution with at least $(1-\delta)$-probability in Subsec. 5.1. Subsequently, SSA is shown to require no more than a constant factor of a type-1 minimum threshold of RR sets with the same probability in Subsec. 5.2.

## 5.1 Approximation Guarantee

In this subsection, we will prove that SSA achieves the approximation factor of $(1-1/e-\epsilon)$ with at least $(1-\delta)$-probability. The proof essentially contains two core components which are two conditions in our RIS framework (Subsection 3.2): 1) prove that at termination SSA achieves a good approximation of the selected solution, $\hat{S}_k$, (Lem. 5) and 2) the hidden optimal solution, $S_k^*$, is also well-estimated (Lem. 6). Thus, combining these with Theo. 1 (Subsection 3.2) gives us the approximation factor $(1-1/e-\epsilon)$ stated in Theo. 2.

The first component states the quality of the estimated influence of the returned solution, $\hat{\mathbb{I}}(\hat{S}_k)$, that SSA has at termination is shown in Lem. 5.

LEMMA 5. *SSA returns a seed set, $\hat{S}_k$, with*

$$\Pr[\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_{12})\mathbb{I}(\hat{S}_k)] \geq 1-\delta/3 \qquad (11)$$

*where $\epsilon_{12} = \epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2$.*

The proof is presented in the appendix. Based on Lem. 5, we prove the second component which also contains the influence estimation of the optimal solution, $S_k^*$.

LEMMA 6. *SSA terminates with*

$$\Pr[\hat{\mathbb{I}}(S_k^*) \leq (1-\epsilon_3)OPT_k \text{ or } \hat{\mathbb{I}}(\hat{S}_k) \geq (1+\epsilon_{12})\mathbb{I}(\hat{S}_k)] \leq 2\delta/3$$

*Thus,*

$$\Pr[\hat{\mathbb{I}}(S_k^*) \geq (1-\epsilon_3)OPT_k] \geq 1-2\delta/3 \qquad (12)$$

Lem. 5 and 6 are sufficient to prove the approximation quality of SSA as stated by the following theorem.

THEOREM 2. *Given $0 \leq \epsilon, \delta \leq 1$ and $\epsilon_1, \epsilon_2, \epsilon_3$ satisfying $\epsilon_1+\epsilon_2+\epsilon_1\epsilon_2+(1-1/e)\epsilon_3 \leq \epsilon$, SSA returns a seed set, $\hat{S}_k$, such that*

$$\Pr[\mathbb{I}(\hat{S}_k) \geq (1-1/e-\epsilon)OPT_k] \geq 1-\delta \qquad (13)$$

PROOF. To prove the theorem, we will show that $|\mathcal{R}|$ is an RIS threshold and then apply Theo. 1 to obtain the $(1-1/e-\epsilon)$-approximation property of SSA. Actually, we will later prove in Sec. 5 that $|\mathcal{R}|$ is not just RIS threshold but, to within a constant factor, a type-1 minimum threshold.

The first condition to become an RIS threshold is taken from Lem. 5,

$$\Pr[\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_1)(1+\epsilon_2)\mathbb{I}(\hat{S}_k)] \geq 1-\delta/3 \qquad (14)$$

The second condition is obtained from Lem. 6,

$$\Pr[\hat{\mathbb{I}}(S_k^*) \geq (1-\epsilon_3)OPT_k] \geq 1-2\delta/3 \qquad (15)$$

From Eq. 14 and Eq. 15, we conclude that $|\mathcal{R}|$ is an RIS threshold with $\epsilon_a = (1+\epsilon_1)(1+\epsilon_2)-1$, $\epsilon_b = \epsilon_3$, $\delta_a = \delta/3$ and $\delta_b = 2\delta/3$. Notice that $\epsilon_a + (1-1/e)\epsilon_b = \epsilon$ and $\delta_a + \delta_b = \delta$. By Theo. 1, we have

$$\Pr[\mathbb{I}(\hat{S}_k) \geq (1-1/e-\epsilon)OPT_k] \geq 1-\delta \qquad (16)$$

which completes the proof of Theo. 2. □

## 5.2 Achieving Type-1 Minimum Threshold

We will analyze the number of RR sets generated in both the main and Estimate-Inf procedures and show that SSA requires no more than a constant times a type-1 minimum RR sets. That makes SSA the first method to achieve a type-1 minimum threshold.

### 5.2.1 Parameter Settings

In Theo. 2, we rely on the assumptions that $\epsilon_1, \epsilon_2, \epsilon_3$ are given such that

$$\epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2 + (1-1/e)\epsilon_3 \leq \epsilon. \qquad (17)$$

Determining their values plays an important role in the algorithm. Ideally, we want to generate just enough RR sets in the main algorithm to have a good estimate of $\hat{S}_k$ and then check the influence by Estimate-Inf procedure. That is because we will discard all the RR sets in the Estimate-Inf after running and, thus, if we start checking too early, we will waste a wealth of RR sets. On the other hand, because of the doubling scheme in the main algorithm, if starting checking way latter, we will generate a lot of unnecessary RR sets in the main procedure. In SSA, we determine the values of $\epsilon_1, \epsilon_2, \epsilon_3$ based on experiments and prove in next subsection that SSA generates only a constant times the type-1 threshold w.r.t these settings. In Sect. 6, we propose a dynamic algorithm, D-SSA, to tune and find the best values during the execution and it requires, within a constant factor, the strongest type-2 minimum threshold RR sets.

We carried experiments on various values of $\epsilon_1, \epsilon_2$ and $\epsilon_3$ which satisfy Eq. 17 and found the following values robustly giving low number of RR sets,

$$\begin{aligned} \epsilon_1 &= \epsilon/6 \\ \epsilon_2 &= \epsilon/2 \\ \epsilon_3 &= \epsilon/4(1-1/e) \end{aligned} \qquad (18)$$

### 5.2.2 Number of RR sets in the main algorithm

Here, we will analyze the number of RR sets generated in the main algorithm and show that this number is at most constant times $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ where $\epsilon_a = \epsilon_{12} = \epsilon_1+\epsilon_2+\epsilon_1\epsilon_2$, $\epsilon_b = \epsilon_3$ and $\delta_a = \delta/3, \delta_b = 2\delta/3$. From Subsec. 5.1,

$$\Pr[\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_a)\mathbb{I}(\hat{S}_k)] \geq 1-\delta_a \qquad (19)$$

and

$$\Pr[\hat{\mathbb{I}}(S_k^*) \geq (1-\epsilon_b)OPT_k] \geq 1-\delta_b \quad (20)$$

are obtained by enforcing two stopping conditions:

$$\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k) \geq \Lambda_1 \quad \text{(C1)}$$

and

$$\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_1)\mathbb{I}_c(\hat{S}_k) \quad \text{(C2)}$$

Now, to determine the number of RR sets generated in SSA, we will start from $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets where Eq. 19 and Eq. 20 are satisfied and determine how many more RR sets needed to meet (C1) and (C2). More specifically, we prove that in the cases that the inequalities in Eq. 19 and Eq. 20 hold with $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets, SSA needs at most $\frac{\mathsf{c}_1}{1-1/e-\epsilon}N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets, where $\mathsf{c}_1$ is a constant defined in Lem. 2. Thus, SSA will need no more than $\frac{\mathsf{c}_1}{1-1/e-\epsilon}N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets with the same probability that two inequalities in Eq. 19 and Eq. 20 hold which is at least $1-\delta$. The satisfaction of (C1) is stated as follows.

Lemma 7. *Suppose that with $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets, we have*

$$\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_a)\mathbb{I}(\hat{S}_k) \quad (21)$$

*and*

$$\hat{\mathbb{I}}(S_k^*) \geq (1-\epsilon_b)OPT_k \quad (22)$$

*Then SSA needs at most $\frac{\mathsf{c}_1}{1-1/e-\epsilon}N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets to satisfy condition (C1).*

The proof is presented in the appendix. The following lemma states that condition (C2) is also satisfied with the same number of RR sets.

Lemma 8. *Given all the assumptions as in Lem. 7, SSA needs at most $\frac{\mathsf{c}_1}{1-1/e-\epsilon}N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets to satisfy condition (C2) with at least $(1-\delta/3)$-probability.*

From Lem. 7 and Lem. 8 and the fact that SSA stops when two stopping conditions, i.e., (C1) and (C2), are satisfied, we have the following lemma.

Lemma 9. *The main algorithm of SSA generates, within a constant factor, $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets with at least $(1-\delta)$-probability.*

Proof. From Lems. 7 and 8, we obtain that the number of RR sets $|\mathcal{R}| \geq \frac{\mathsf{c}_1}{1-1/e-\epsilon}N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ is the necessary condition to satisfy both (C1) and (C2) with the probability accumulated from Eq. 19, Eq. 20 and Lem. 8 that is $1-\delta_b-\delta/3 = 1-\delta$ (follows from Lem. 6 that the inequalities in Eq. 19 and 20 are both satisfied with probability $1-\delta_b$). Therefore, due to the doubling scheme, SSA will generate at most $\frac{2\mathsf{c}_1}{1-1/e-\epsilon}N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ which remains to be a constant times $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$. □

### 5.2.3 Number of RR sets in Estimate-Inf procedure

As presented in Alg. 3, the number of RR sets generated in Estimate-Inf procedure is always smaller than $\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2}$ times the number of RR sets currently in the main algorithm. Therefore, the total number of RR sets generated during the running of SSA is smaller than $\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2}$ times the sum of RR sets present at each iteration in the main algorithm. In turn, due to the doubling behavior, the sum of RR sets is smaller than twice that number at the last iteration. Thus, based on Lem. 9, we have the following lemma.

Lemma 10. *Estimate-Inf procedure of SSA generates, to within a constant factor, $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets with at least $(1-\delta)$-probability.*

The constant in the lemma is $2\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2}$ times that in Lem. 9 that makes it

$$2\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2}\frac{2\mathsf{c}_1}{1-1/e-\epsilon}N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b) \quad (23)$$

Combining Lems. 9 and 10, we conclude the overall number of RR sets by the following theorem.

Theorem 3. *SSA generates, to within a constant factor, $N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ RR sets with at least $(1-\delta)$-probability.*

The total number in this theorem is the sum of those in Lems. 9 and 10,

$$(1+2\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2})\frac{2\mathsf{c}_1}{1-1/e-\epsilon}N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b) \quad (24)$$

# 6. DYNAMIC ALGORITHM

In this section, we propose the dynamic algorithm, named D-SSA, that automatically selects $\epsilon_1, \epsilon_2, \epsilon_3, \delta_1, \delta_2$ during its execution. While maintaining the $(1-1/e-\epsilon)$-approximate solution as in SSA, D-SSA requires, to within a constant factor, the type-2 minimum threshold. This is the strongest result over all IM methods following the RIS framework.

## 6.1 D-SSA Algorithm

In Section 4, SSA can be seen as generating two independent collections of RR sets: one is in the main algorithm to find the maximum seed set and the another is for estimating the influence of the seed set found in the main procedure. Recall from Section 5 that we want to start checking the influence of $\hat{S}_k$ at the moment of having generated just enough RR sets so that the RR sets for checking are not wasted. However, detecting that moment is challenging since it depends not only on the networks but also the particular execution of generating RR sets.

The dynamic algorithm D-SSA, described in Alg. 4, addresses thoroughly these issue by dynamically computing the values of $\epsilon_1, \epsilon_2, \epsilon_3, \delta_1, \delta_2$ along its execution and stops whenever the success probability meets the requirement (Line 12). D-SSA also reuses the checking RR sets for finding seed set without affecting the independence of RR sets in subsequent iterations (Line 16). More specifically, D-SSA uses the newly generated RR sets to estimates the influence of the seed set found in the previous iteration and obtain the current value of $\epsilon_1$ (Line 7). From the value of $\epsilon_1$, $\epsilon_2$ and $\epsilon_3$ can be computed accordingly (Line 8). The formula for computing $\epsilon_2$ and $\epsilon_3$ are based on the condition that

$$\epsilon_1+\epsilon_2+\epsilon_1\epsilon_2+\epsilon_3(1-1/e) \leq \epsilon \quad (25)$$

and considering $\epsilon_2+\epsilon_1\epsilon_2$ and $\epsilon_3(1-1/e)$ having similar roles. After that it relies on the two stopping conditions mentioned in SSA to calculate $\delta_1$ and $\delta_2$. D-SSA stops when sum of $\delta_1$ and $\delta_2$ is less than or equal to $\theta$ which signifies that the success probability meets the requirement.

## 6.2 D-SSA Analysis

We will sequentially show that D-SSA achieves $(1-1/e-\epsilon)$-approximation factor in Subsec. 6.2.1 and requires only, to within a constant factor, the strongest type-2 minimum threshold of the RR sets in Subsec. 6.2.2.

**Algorithm 4** D-SSA Algorithm

**Input**: Graph $G$, $0 \le \epsilon, \delta \le 1$, and $k$.
**Output**: An $(1-1/e-\epsilon)$-optimal solution, $\hat{S}_k$.
1: $\Lambda \leftarrow 2c(1+\epsilon)^2 \ln\frac{2}{\delta}\frac{1}{\epsilon^2}$
2: $\mathcal{R} \leftarrow$ Generate $\Lambda$ random RR sets by RIS
3: $<\hat{S}_k, \hat{\mathbb{I}}(\hat{S}_k)> \leftarrow$ Max-Coverage$(\mathcal{R}, k)$
4: **repeat**
5: $\quad\mathcal{R}' \leftarrow$ Generate $|\mathcal{R}|$ random RR sets by RIS
6: $\quad\mathbb{I}_c(\hat{S}_k) \leftarrow \mathrm{Cov}_{\mathcal{R}'}(\hat{S}_k) \cdot n/|\mathcal{R}'|$
7: $\quad\epsilon_1 \leftarrow \hat{\mathbb{I}}(\hat{S}_k)/\mathbb{I}_c(\hat{S}_k) - 1$
8: $\quad$**if** $(\epsilon_1 \le \epsilon)$ **then**
9: $\quad\quad\epsilon_2 \leftarrow \frac{\epsilon-\epsilon_1}{2(1+\epsilon_1)}$, $\epsilon_3 \leftarrow \frac{\epsilon-\epsilon_1}{2(1-1/e)}$
10: $\quad\quad\delta_1 \leftarrow e^{-\frac{\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k)\cdot\epsilon_3^2}{2c(1+\epsilon_1)(1+\epsilon_2)}}$
11: $\quad\quad\delta_2 \leftarrow e^{-\frac{(\mathrm{Cov}_{\mathcal{R}'}-1)(\hat{S}_k)\cdot\epsilon_2^2}{2c(1+\epsilon_2)}}$
12: $\quad\quad$**if** $\delta_1 + \delta_2 \le \delta$ **then**
13: $\quad\quad\quad$**return** $\hat{S}_k$
14: $\quad\quad$**end if**
15: $\quad$**end if**
16: $\quad\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{R}'$
17: $\quad<\hat{S}_k, \hat{\mathbb{I}}(\hat{S}_k)> \leftarrow$ Max-Coverage$(\mathcal{R}, k)$
18: **until** $|\mathcal{R}| \ge (8+2\epsilon)n \cdot \frac{\ln\frac{2}{\delta}+\ln\binom{n}{k}}{\epsilon^2}$
19: **return** $\hat{S}_k$

### 6.2.1 Approximation Guarantee

We show that D-SSA preserve the $(1-1/e-\epsilon)$-approximation factor of SSA by the following theorem.

THEOREM 4. *Given a graph $G$, $0 \le \epsilon \le 1-2/e$ and $0 \le \delta \le 1$ as the inputs, D-SSA returns a $(1-1/e-\epsilon)$-approximate solution.*

PROOF. we prove that the two stopping conditions in SSA are still hold in D-SSA and thus D-SSA has the same approximation factor as SSA does.

Directly from the Alg. 4, when D-SSA terminates, the following conditions are satisfied,

$$\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k) = 2c(1+\epsilon_1)(1+\epsilon_2)\ln\frac{1}{\delta_1}\frac{1}{\epsilon_3^2} \tag{26}$$

and

$$\mathrm{Cov}_{\mathcal{R}'}(\hat{S}_k) = 1 + 2c(1+\epsilon_2)\ln\frac{1}{\delta_2}\frac{1}{\epsilon_2^2} \tag{27}$$

with $\epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2 + (1-1/e)\epsilon_3 = \epsilon$ and $\delta_1 + \delta_2 = \delta$.

Eq. 26 is the first stopping conditions in SSA. Eq. 27 is the checking condition in the Estimate-Inf procedure in SSA and, together with the setting of $\epsilon_1$, we obtain the second stopping condition in SSA.

$$\hat{\mathbb{I}}(\hat{S}_k) \ge (1+\epsilon_1)(1+\epsilon_2)\mathbb{I}(\hat{S}_k) \tag{28}$$

Thus, the $(1-1/e-\epsilon)$-approximation factor is followed from SSA which completes the proof. □

### 6.2.2 Achieving the Type-2 Minimum Threshold

Here, we will prove a much stronger result than that of SSA that D-SSA requires only, to within a constant factor, the type-2 minimum of RR sets $N^2_{min}(\epsilon, \delta)$. Since $\epsilon^*_a \le \epsilon$ and $\delta^*_b \le \delta$, let denote the constant

$$M = \frac{\ln\frac{2}{\delta^*_b}}{\ln\frac{1}{\delta^*_b}}\frac{4\cdot(1+\epsilon^*_a)^2}{(\frac{1}{2}-\frac{1}{e}-\frac{\epsilon}{2})^3} \le \frac{\ln\frac{2}{\delta}}{\ln\frac{1}{\delta}}\frac{4\cdot(1+\epsilon)^2}{(\frac{1}{2}-\frac{1}{e}-\frac{\epsilon}{2})^3} \tag{29}$$

Alternatively speaking, for a given graph $G$, we will show that D-SSA terminates when the number of RR sets in $\mathcal{R}$ is larger than or equal to $M\mathsf{c}_1 N^2_{min}(\epsilon, \delta)$, where $\mathsf{c}_1$ is a constant defined in Lem. 2, with at least $(1-\delta)$-probability. Due to the doubling scheme, D-SSA will generate no more than twice that number. More specifically, we will prove that the stopping condition (Line 11) is satisfied. Recall $N^2_{min}(\epsilon, \delta)$ RR sets implies

$$\Pr[\hat{\mathbb{I}}(\hat{S}_k) \le (1+\epsilon^*_a)\mathbb{I}(\hat{S}_k)] \ge 1-\delta^*_a \tag{30}$$

and

$$\Pr[\hat{\mathbb{I}}(S^*_k) \ge (1-\epsilon^*_b)OPT_k] \ge 1-\delta^*_b \tag{31}$$

where

$$N^1_{min}(\epsilon^*_a, \epsilon^*_b, \delta^*_a, \delta^*_b) = \min_{\epsilon_a, \epsilon_b, \delta_a, \delta_b} N^1_{min}(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$$

Apply Theo. 1, we have

$$\Pr[\mathbb{I}(\hat{S}_k) \ge (1-1/e-\epsilon)OPT_k] \ge 1-\delta \tag{32}$$

First, we show that the value of $\epsilon_1$ is upper bounded by the following lemma.

LEMMA 11. *Given $0 \le \epsilon \le 1-2/e$ and $0 \le \delta \le 1$, if $|\mathcal{R}| \ge M\mathsf{c}_1 N^2_{min}(\epsilon, \delta)$, then,*

$$\Pr[\epsilon_1 \le \frac{\epsilon^*_a + \epsilon^*_b/2}{1-\epsilon^*_b/2}] \ge 1-\delta \tag{33}$$

*where $\mathsf{c}_1$ is a constant defined in Lem. 2.*

The proof is presented in the appendix. Based on the result of Lem. 11, we next prove that D-SSA will terminate when $|\mathcal{R}| \ge M\mathsf{c}_1 N^2_{min}(\epsilon, \delta)$ with at least $(1-\delta)$-probability.

LEMMA 12. *If $|\mathcal{R}| \ge M\mathsf{c}_1 N^2_{min}(\epsilon, \delta)$ and $\epsilon_1 \le \frac{\epsilon^*_a+\epsilon^*_b/2}{1-\epsilon^*_b/2}$, then,*

$$\delta_1 + \delta_2 \le \delta \tag{34}$$

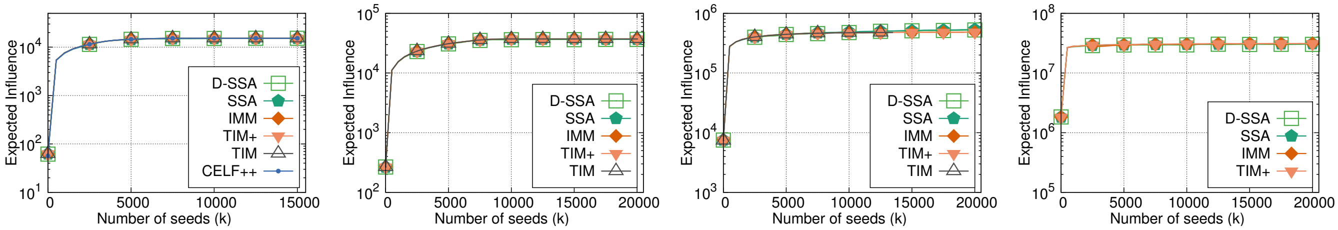
Therefore, D-SSA needs at most a constant times the type-2 minimum with the probability of $\Pr[\epsilon_1 \le \frac{\epsilon^*_a+\epsilon^*_b/2}{1-\epsilon^*_b/2}]$ which is at least $1-\delta$. Since we double the RR sets every round, the actual number of RR sets is at most twice the necessary one. Thus, we conclude that D-SSA generates at most a constant times the type-2 minimum with probability of at least $1-\delta$.

THEOREM 5. *Given a graph $G$, $0 \le \epsilon \le 1-2/e$ and $0 \le \delta \le 1$, let $N^2_{min}(\epsilon, \delta)$ be the type-2 minimum threshold of RR sets. D-SSA generates no more than, to within a constant factor, $N^2_{min}(\epsilon, \delta)$ RR sets with the probability of at least $(1-\delta)$.*

The Theo. 5 follows directly from Lems. 11 and 12.

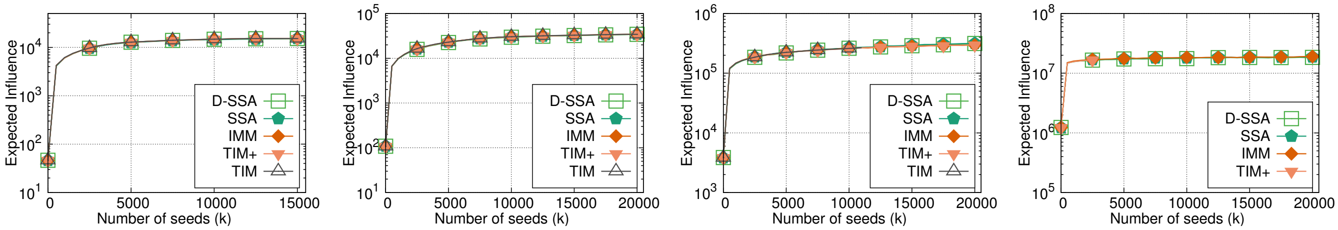
## 7. EXPERIMENTS

Backing by the strong theoretical results, we will experimentally show that SSA and D-SSA outperform the existing state-of-the-art IM methods by a large margin. Specifically, SSA and D-SSA are several orders of magnitudes faster than IMM and TIM+, the best existing IM methods with approximation guarantee, while having the same level of solution quality. SSA and D-SSA also require several times less memory than the other algorithms. To demonstrate the applicability of the proposed algorithms, we apply our methods on a critical application of IM, i.e., Targeted Viral Marketing (TVM) introduced in [29] and show the significant improvements in terms of performance over the existing methods.




(a) NetHEPT (b) NetPHY (c) DBLP (d) Twitter

Figure 4: Expected Influence under LT model.




(a) NetHEPT (b) NetPHY (c) DBLP (d) Twitter

Figure 5: Expected Influence under IC model.

Table 2: Datasets' Statistics

| Dataset | #Nodes | #Edges | Avg. degree |
|---|---|---|---|
| NetHELP[3] | 15K | 59K | 4.1 |
| NetPHY[3] | 37K | 181K | 13.4 |
| Enron[3] | 37K | 184K | 5.0 |
| Epinions[3] | 132K | 841K | 13.4 |
| DBLP[3] | 655K | 2M | 6.1 |
| Orkut[3] | 3M | 117M | 78 |
| Twitter [32] | 41.7M | 1.5G | 70.5 |
| Friendster[3] | 65.6M | 1.8G | 54.8 |

## 7.1 Experimental Settings

All the experiments are run on a Linux machine with 2.2Ghz Xeon 8 core processor and 100GB of RAM. We carry experiments under both LT and IC models on the following algorithms and datasets.

**Algorithms compared.** On IM experiments, we compare SSA and D-SSA with the group of top algorithms that provide the same $(1 - 1/e - \epsilon)$-approximation guarantee. More specifically, CELF++ [33], one of the fastest greedy algorithms, and IMM [16], TIM/TIM+ [8], the best current RIS-based algorithms, are selected. For experimenting with TVM problem, we apply our Stop-and-Stare algorithms on this context and compare with the most efficient method for the problem, termed KB-TIM, in [29].

**Datasets.** For experimental purposes, we choose a set of 8 datasets from various disciplines: NetHEPT, NetPHY, DBLP are citation networks, Email-Enron is communication network, Epinions, Orkut, Twitter and Friendster are online social networks. The description summary of those datasets is in Table 2. On Twitter network, we also have the actual tweet/retweet dataset and we use these data to extract the target users whose tweets/retweets are relevant to a certain set of keywords. The experiments on TVM are run on the Twitter network with the extracted targeted groups of users.

**Parameter Settings.** For computing the edge weights, we follow the conventional computation as in [8, 13, 4, 26], the weight of the edge $(u, v)$ is calculated as $w(u, v) = \frac{1}{d_{in}(v)}$ where $d_{in}(v)$ denotes the in-degree of node $v$.

In all the experiments, we keep $\epsilon = 0.1$ and $\delta = 1/n$ as a general setting or explicitly stated otherwise. For the other parameters defined for particular algorithms, we take the recommended values in the corresponding papers if available. We also limit the running time of each algorithm in a run to be within 24 hours.

## 7.2 Experiments with IM problem

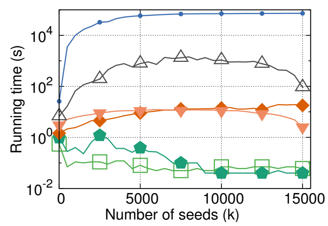
To show the superior performance of the proposed algorithms on IM task, we ran the first set of experiments on four real-world networks, i.e., NetHEPT, NetPHY, DBLP, Twitter. We also test on a wide spectrum of the value of $k$, typically, from 1 to 20000, except on NetHEPT network since it has only 15233 nodes. The solution quality, running time, memory usage are reported sequentially in the following. We also present the actual number of RR sets generated by SSA, D-SSA and IMM when testing on four other datasets, i.e., Enron, Epinions, Orkut and Friendster.

### *7.2.1 Solution Quality*

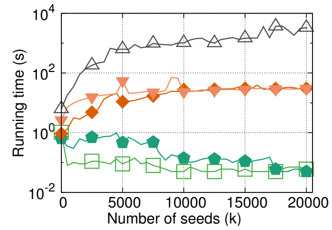
We first compare the quality of the solution returned by all the algorithms on LT and IC models. The results are presented in Fig. 4 and Fig. 5, respectively. The CELF++ algorithm is only able to run on NetHEPT due to time limit. From those figures, all the methods return comparable seed set quality with no significant difference. The results directly give us a better viewpoint on the basic network property that a small fraction of nodes can influence a very large portion of the networks. Most of the previous researches only find up to 50 seed nodes and provide a limited view of this phenomenon. Here, we see that after around 2000 nodes have been selected, the influence gains of selecting more seeds become very slim.

### *7.2.2 Running time*

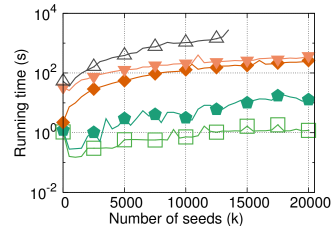
We next examine the performance in terms of running time of the tested algorithms. The results are shown in Fig. 6 and Fig. 7. Both SSA and D-SSA significantly outper-



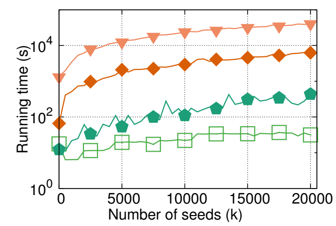

(a) NetHEPT




(b) NetPHY

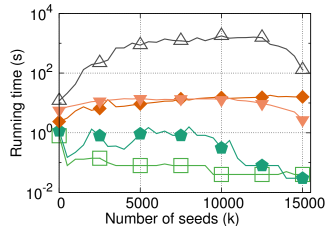



(c) DBLP

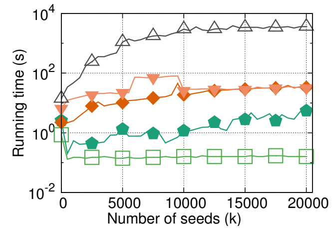



(d) Twitter

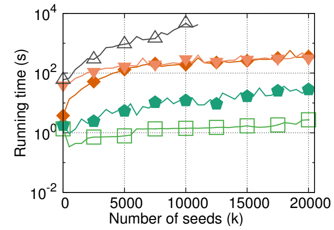
Figure 6: Running time under LT model



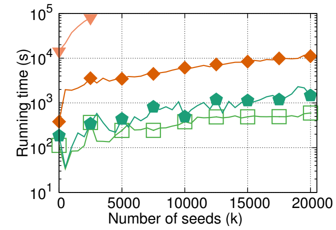

(a) NetHEPT




(b) NetPHY

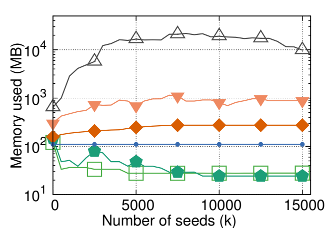



(c) DBLP

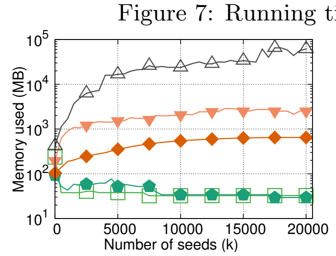



(d) Twitter

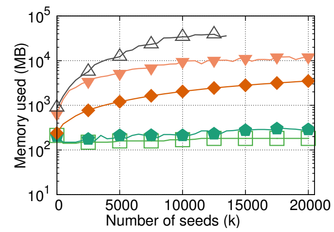
Figure 7: Running time under IC model



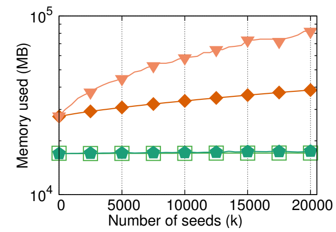

(a) NetHEPT




(b) NetPHY

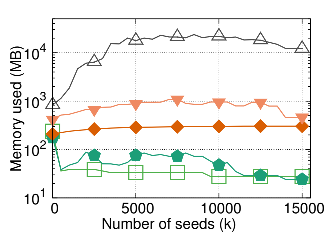



(c) DBLP

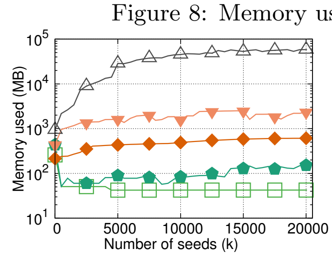



(d) Twitter

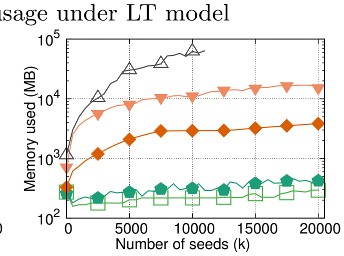
Figure 8: Memory usage under LT model



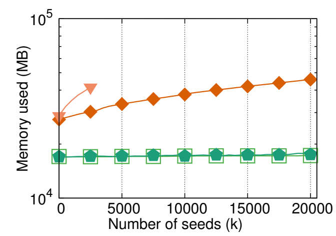

(a) NetHEPT




(b) NetPHY




(c) DBLP




(d) Twitter

Figure 9: Memory usage under IC model

| Data | **running time**(s) | | | | | | | | | **number of RR sets**(thousands) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $k=1$ | | | $k=500$ | | | $k=1000$ | | | $k=1$ | | | $k=500$ | | | $k=1000$ | | |
| | D-SSA | SSA | IMM | D-SSA | SSA | IMM | D-SSA | SSA | IMM | D-SSA | SSA | IMM | D-SSA | SSA | IMM | D-SSA | SSA | IMM |
| Enron | **0.2** | 0.6 | 0.7 | **0.1** | 0.2 | 3.1 | **0.2** | 0.6 | 6.9 | **120** | 180 | 2.8 | **30** | 60 | 580 | **120** | 180 | 910 |
| Epin. | **0.4** | 0.8 | 0.8 | **0.1** | **0.1** | 4.4 | **0.3** | 0.7 | 12.1 | **160** | 300 | 400 | **40** | 60 | 1214 | **160** | 240 | 1852 |
| Orkut | **0.5** | 1.1 | 2.1 | **0.6** | 3.4 | 17.4 | **0.6** | 4.3 | 45.5 | **30** | 60 | 124 | **60** | 240 | 760 | **60** | 240 | 1392 |
| Frien. | **6.7** | 10.3 | 270 | **3.5** | 5.9 | 4321.8 | **12.4** | 27.7 | 8028 | **60** | 120 | 262 | **30** | 60 | 1322 | **120** | 240 | 2250 |

Table 3: Across dataset view of performance of SSA, D-SSA and IMM on various datasets under LT model.

form the other competitors by a huge margin. Comparing to IMM, the best known algorithm, SSA and D-SSA run up to several orders of magnitudes faster. TIM+ and IMM show similar running time since they operate on the same philosophy of estimating optimal influence first and then calculating the necessary samples to guarantee the approximation for all possible seed sets. However, each of the two steps displays its own weakness. In contrast, SSA and D-SSA follows the Stop-and-Stare mechanism to thoroughly address those weaknesses and thus exhibit remarkable improvements.

Comparing between SSA and D-SSA, since D-SSApossesses the type-2 minimum threshold compared to the weaker type-1 threshold of SSA with the same precision settings $\epsilon, \delta$, D-SSA achieves considerably better performance and commits up to an order of magnitudes speedup.

#### 7.2.3 *Memory Usage and Number of RR sets*

This experiment is divided into two parts: 1) we report the memory usage in the previous experiments and 2) since the gain in influence peaks at the selection of 1 to 1000 nodes, we carry new experiments on four other datasets, i.e., Enron, Epinion, Orkut and Friendster, with $k \in \{1, 500, 1000\}$ to show the view across datasets of SSA, D-SSA and IMM.

*Memory Usage.* The results on memory usage of all the algorithms are shown in Fig. 8 and Fig. 9. We can see that there is a strong correlation between running time and memory usage. It is not a surprise that SSA and D-SSA require much less memory, up to orders of magnitude, than the other methods since the complexity is represented by the number of RR sets and these methods achieve type-1 and type-2 minimum thresholds of RR sets.

*Across datasets view.* We ran SSA, D-SSA and IMM on four other datasets, i.e., Enron, Epinions, Orkut and Friendster, with $k \in \{1, 500, 1000\}$ under LT model. The results are presented in Table 3. In terms of running time, the table reflects our previous results that SSA and D-SSA largely outperform IMM, up to several orders of magnitudes. In particular, for Friendster - the largest network in our testbed with more than 65 million nodes, D-SSA is 1200-fold faster than IMM in case of $k = 500$. The same pattern happens in terms of the number of RR sets generated. As shown, even in the most extreme cases of selecting a single node, SSA and D-SSA require several times fewer RR sets than IMM.

### 7.3 Experiments with TVM problem

In this experiments, we will modify our Stop-and-Stare algorithms to work on Targeted Viral Marketing (TVM) problem and compare with the best existing method, i.e., KB-TIM in [29] to show the drastic improvements when applying our methods. In short, we will describe how we select the targeted groups from actual tweet/retweet datasets of Twitter and how to modify D-SSA and SSA for TVM problem. Then, we will report the experimental results.

#### 7.3.1 *TVM problem and methods*

Targeted Viral Marketing (TVM) is a central problem in economics in which, instead of maximizing the influence over all the nodes in a network as in IM, it targets a specific group whose users are relevant to a certain topic and aims at optimizing the influence to that group only. Each node in the targeted group is associated with a weight which indicates the relevance of that user to the topic. The best current method for solving TVM is proposed in [29] in which the authors introduce weighted RIS sampling (called WRIS) and integrate it into TIM+ method [8] to derive an approximation algorithm, termed KB-TIM. WRIS only differs from the original RIS at the point of selecting the sampling root. More specifically, WRIS selects the root node proportional to the node weights instead of uniform selection as in RIS.

In the same way, we incorporate WRIS into D-SSA and SSA for solving TVM problem. By combining the analysis of WRIS in [29] and our previous proofs, it follows that the modified D-SSA and SSA preserve the $(1 - 1/e - \epsilon)$-approximation property as in IM problem.

#### 7.3.2 *Extracting the targeted groups*

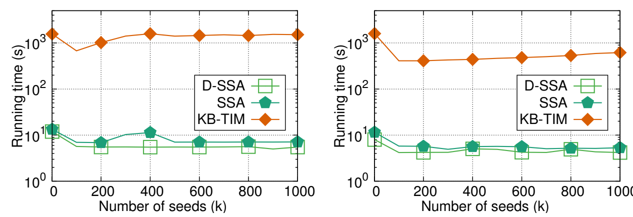
We use tweet/retweet dataset to extract the users' interests on two political topics as described in [32]. We choose two groups of most popular keywords as listed in Table 4,

Table 4: Topics, related keywords

| *Topic* | *Keywords* | #Users |
|---|---|---|
| 1 | **bill clinton, iran, north korea, president obama, obama** | 997,034 |
| 2 | **senator ted kenedy, oprah, kayne west, marvel, jackass** | 507,465 |

and mine from the tweet data who posted tweets/reweets containing at least one of those keywords in each group and how many times. We consider those users to be the targeted groups in TVM experiments with the relevance/interest of each user on the topic proportional to the frequency of having those keywords in their tweets.

#### 7.3.3 *Experimental results*




(a) Topic 1 (b) Topic 2

Figure 10: Running time on Twitter network

We run SSA, D-SSA and KB-TIM on Twitter network under LT model with the targeted groups extracted from tweet dataset as described previously. Since all the algorithms have the same guarantee on the returned solution, we only measure the performance of these methods in terms of running time and the results are depicted in Fig. 10. In both cases, D-SSA and SSA consistently witness at least two order of magnitude improvements (up to 500 times) in running time compared to KB-TIM. D-SSA is also consistently faster than SSA due to the more optimal type-2 threshold.

## 8. CONCLUSION

In this paper, we make several significant contributions in solving the fundamental influence maximization (IM) problem. We provide the unified RIS framework which generalizes the best existing technique of using RIS sampling to find an $(1 - 1/e - \epsilon)$-approximate solution in billion-scale networks. We introduce the RIS threshold that all the algorithms following the framework need to satisfy and two minimum thresholds, i.e., type-1 and type-2. Interestingly, we are able to develop two novel algorithms, SSA and D-SSA, which are the first methods meeting the two minimum thresholds. Since IM plays a central roles in a wide range of practical applications, e.g., viral marketing, controlling diseases, virus/worms, detecting contamination and so on, the developments of SSA and D-SSA will immediately result in a burst in performance and allow their applications to work in billion-scale domains.

## 9. ACKNOWLEDGMENTS

The work of My T. Thai is partially supported by NSF CCF 1422116.

## 10. REFERENCES


[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *KDD'03*, pp. 137–146, ACM New York, NY, USA, 2003.

[2] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *ACM KDD '07*, (New York, NY, USA), pp. 420–429, ACM, 2007.

[3] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *ACM KDD '10*, (New York, NY, USA), pp. 1029–1038, ACM, 2010.

[4] A. Goyal, W. Lu, and L. Lakshmanan, "Simpath: An efficient algorithm for influence maximization under the linear threshold model," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pp. 211–220, IEEE, 2011.

[5] A. Goyal, W. Lu, and L. Lakshmanan, "Celf++: optimizing the greedy algorithm for influence maximization in social networks," in *Proceedings of the 20th international conference companion on World wide web*, pp. 47–48, ACM, 2011.

[6] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Sketch-based influence maximization and computation: Scaling up with guarantees," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 629–638, ACM, 2014.

[7] N. Ohsaka, T. Akiba, Y. Yoshida, and K.-i. Kawarabayashi, "Fast and accurate influence maximization on large networks with pruned monte-carlo simulations," in *Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[8] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *Proceedings of SIGMOD international conference on Management of data*, pp. 75–86, ACM, 2014.

[9] D. T. Nguyen, H. Zhang, S. Das, M. T. Thai, and T. N. Dinh, "Least cost influence in multiplex social networks: Model representation and analysis," in *Data Mining (ICDM), 2013 IEEE 13th International Conference on*, pp. 567–576, IEEE, 2013.

[10] T. N. Dinh, Y. Shen, D. T. Nguyen, and M. T. Thai, "On the approximability of positive influence dominating set in social networks," *Journal of Combinatorial Optimization*, vol. 27, no. 3, pp. 487–503, 2014.

[11] Y. Shen, T. N. Dinh, H. Zhang, and M. T. Thai, "Interest-matching information propagation in multiple online social networks," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, (New York, NY, USA), pp. 1824–1828, ACM, 2012.

[12] H. T. Nguyen, M. T. Thai, and T. N. Dinh, "Cost-aware targeted viral marketing in billion-scale networks," in *INFOCOM, 2016 Proceedings IEEE*, pp. 55–59, IEEE, 2013.

[13] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *KDD '09*, (New York, NY, USA), pp. 199–208, ACM, 2009.

[14] K. Jung, W. Heo, and W. Chen, "Irie: Scalable and robust influence maximization in social networks," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*, pp. 918–923, IEEE, 2012.

[15] Y. Wang, G. Cong, G. Song, and K. Xie, "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*, pp. 1039–1048, ACM, 2010.

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

[17] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14, pp. 946–957, SIAM, 2014.

[18] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques* (J. Stoer, ed.), vol. 7 of *Lecture Notes in Control and Information Sciences*, pp. 234–243, Springer, 1978.

[19] N. Chen, "On the approximability of influence in social networks," *SIAM Journal of Discrete Mathematics*, vol. 23, no. 3, pp. 1400–1415, 2009.

[20] A. Goyal, F. Bonchi, and L. Lakshmanan, "Learning influence probabilities in social networks," in *Proceedings of the third ACM international conference on Web search and data mining*, pp. 241–250, ACM, 2010.

[21] K. Kutzkov, A. Bifet, F. Bonchi, and A. Gionis, "Strip: stream learning of influence probabilities," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 275–283, ACM, 2013.

[22] T. Dinh, H. Zhang, D. Nguyen, and M. Thai, "Cost-effective viral marketing for time-critical campaigns in large-scale social networks," *Networking, IEEE/ACM Transactions on*, 2014.

[23] H. Zhang, T. Dinh, and M. Thai, "Maximizing the spread of positive influence in online social networks," in *Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on*, pp. 317–326, July 2013.

[24] W. Chen, W. Lu, and N. Zhang, "Time-critical influence maximization in social networks with time-delayed diffusion process," in *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.

[25] N. Du, L. Song, M. Gomez-Rodriguez, and H. Zha, "Scalable influence estimation in continuous-time diffusion networks," in *Advances in neural information processing systems*, pp. 3147–3155, 2013.

[26] H. Nguyen and R. Zheng, "On budgeted influence maximization in social networks," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 6, pp. 1084–1094, 2013.

[27] P. Dagum, R. Karp, M. Luby, and S. Ross, "An optimal algorithm for monte carlo estimation," *SIAM J. Comput.*, vol. 29, pp. 1484–1496, Mar. 2000.

[28] S. Khuller, A. Moss, and J. S. Naor, "The budgeted

maximum coverage problem," *Information Processing Letters*, vol. 70, no. 1, pp. 39–45, 1999.

[29] Y. Li, D. Zhang, and K.-L. Tan, "Real-time targeted influence maximization for online advertisements," *Proceedings of the VLDB Endowment*, vol. 8, no. 10, 2015.

[30] S. Bhattacharya, M. Henzinger, D. Nanongkai, and C. Tsourakakis, "Space-and time-efficient algorithm for maintaining dense subgraphs on one-pass dynamic streams," in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pp. 173–182, ACM, 2015.

[31] G. L. Nemhauser and L. A. Wolsey, "Maximizing submodular set functions: formulations and analysis of algorithms," *North-Holland Mathematics Studies*, vol. 59, pp. 279–301, 1981.

[32] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?," in *Proceedings of the 19th international conference on World wide web*, pp. 591–600, ACM, 2010.

[33] A. Goyal, W. Lu, and L. V. Lakshmanan, "Celf++: optimizing the greedy algorithm for influence maximization in social networks," in *Proceedings of the 20th international conference companion on World wide web*, pp. 47–48, ACM, 2011.

# APPENDIX

In our proof, we utilize the following version of Chernoff-Hoeffding's inequality,

LEMMA 13 ([27]). *Let $Z_1, Z_2, \ldots$ be independently and identically distributed samples according to $Z$ in the interval [0, 1] with mean $\mu_Z$ and variance $\sigma_Z^2$. Let $\hat{\mu}_Z = \frac{1}{T}\sum_{i=1}^{T} Z_i$ be an estimate of $\mu_Z$. For any fixed $T > 0, 0 \geq \epsilon \geq 1$,*

$$\Pr[\hat{\mu} \geq (1+\epsilon)\mu] \leq e^{\frac{-T\mu\epsilon^2}{2c}} \tag{35}$$

*and*

$$\Pr[\hat{\mu} \leq (1-\epsilon)\mu] \leq e^{\frac{-T\mu\epsilon^2}{2c}}. \tag{36}$$

## Proof of Theorem 1

According to Def. 4, if the number of RR sets generated $|\mathcal{R}| \geq N(\epsilon_a, \epsilon_b, \delta_a, \delta_b)$ then Eqs. 6 and 7 hold. From Eq. 6, we have

$$\Pr[\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_a)\mathbb{I}(\hat{S}_k)] \geq 1-\delta_a \tag{37}$$

which is equivalent to

$$\Pr[\mathbb{I}(\hat{S}_k) \geq \hat{\mathbb{I}}(\hat{S}_k) - \epsilon_a\mathbb{I}(\hat{S}_k)] \geq 1-\delta_a \tag{38}$$

Let $S_{kmax} \in V$ be the size-$k$ set with maximum coverage in $\mathcal{R}$. The greedy strategy in finding max-coverage returns the set, $\hat{S}_k$, with $\hat{\mathbb{I}}(\hat{S}_k) \geq (1-1/e)\hat{\mathbb{I}}(S_{kmax})$ [31]. Since $\hat{\mathbb{I}}(S_{kmax}) \geq \hat{\mathbb{I}}(S^*_k)$, $\hat{\mathbb{I}}(\hat{S}_k) \geq (1-1/e)\hat{\mathbb{I}}(S^*_k)$. Thus,

$$\Pr[\mathbb{I}(\hat{S}_k) \geq (1-1/e)\hat{\mathbb{I}}(S^*_k) - \epsilon_a\mathbb{I}(\hat{S}_k)] \geq 1-\delta_a$$

Now by substituting $\hat{\mathbb{I}}(S^*_k)$ in this probability with that from Eq. 7 and noting that the right hand side is updated to $1-\delta_a-\delta_b \geq 1-\delta$ to account for the probability in Eq. 7. Hence,

$$\Pr[\mathbb{I}(\hat{S}_k) \geq (1-1/e)(1-\epsilon_b)OPT_k - \epsilon_a\mathbb{I}(\hat{S}_k)] \geq 1-\delta$$

Since $\mathbb{I}(\hat{S}_k) \leq OPT_k$,

$$\Pr[\mathbb{I}(\hat{S}_k) \geq (1-1/e)(1-\epsilon_b)OPT_k - \epsilon_a OPT_k] \geq 1-\delta$$

or,

$$\Pr[\mathbb{I}(\hat{S}_k) \geq (1-1/e-(1-1/e)\epsilon_b - \epsilon_a)OPT_k] \geq 1-\delta$$

and, finally, due to $\epsilon \geq \epsilon_a + (1-1/e)\epsilon_b$,

$$\Pr[\mathbb{I}(\hat{S}_k) \geq (1-1/e-\epsilon)OPT_k] \geq 1-\delta$$

That completes the proof of Theo. 1.

*Remark*: As shown in [8], the number of RR sets represent the complexity of an RIS based method. Let $KPT$ be the expected time to generate an RR set, then $|\mathcal{R}|KPT$ is the complexity of generating all the RR sets. Moreover, the finding Max-coverage step can be done in linear time w.r.t the total size of RR sets which is upper-bounded by the generating RR sets.

## Proof of Lemma 5

Recall that SSA terminates when the selected seed set, $\hat{S}_k$, satisfies two conditions,

$$\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k) \geq \Lambda_1 \tag{39}$$

and

$$\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_1)\mathbb{I}_c(\hat{S}_k) \tag{40}$$

From Corollary 1, we have

$$\Pr[\mathbb{I}_c(S) \leq (1+\epsilon_2)\mathbb{I}(S)] \geq 1-\delta/3 \tag{41}$$

Substitute $\mathbb{I}_c(\hat{S}_k)$ in Eq. 41 with that of Eq. 40,

$$\Pr[\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_1)(1+\epsilon_2)\mathbb{I}(\hat{S}_k)] \geq 1-\delta/3 \tag{42}$$

This completes the proof of Lem. 5.

## Proof of Lemma 6

To prove the lemma, we first find a lower-bound of the number of RR sets, $|\mathcal{R}|$, generated and, thus, show that $|\mathcal{R}| \geq \frac{\Upsilon(\epsilon_3,\delta/3)n}{OPT_k}$ with high probability and, then, apply Chernoff-Hoeffding's inequality. Recall that

$$\hat{\mathbb{I}}(\hat{S}_k) = \frac{\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k)n}{|\mathcal{R}|} \tag{43}$$

Apply the first stopping condition, $\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k) \geq \Lambda_1$,

$$\hat{\mathbb{I}}(\hat{S}_k) \geq \frac{\Lambda_1 n}{|\mathcal{R}|} = \frac{(1+\epsilon_1)(1+\epsilon_2)\Upsilon(\epsilon_3,\delta/3)n}{|\mathcal{R}|}$$

where $\Upsilon(\epsilon_3, \delta/3) = 2c\ln(3/\delta)1/\epsilon_3^2$.
Now, combine the result of Lem. 5 with the above inequality, we have

$$\Pr[\frac{\Upsilon(\epsilon_3,\delta/3)n}{|\mathcal{R}|} \leq \mathbb{I}(\hat{S}_k)] \geq 1-\delta/3$$

Thus,

$$\Pr[|\mathcal{R}| \geq \frac{\Upsilon(\epsilon_3,\delta/3)n}{\mathbb{I}(\hat{S}_k)}] \geq 1-\delta/3 \tag{44}$$

and due to $\mathbb{I}(\hat{S}_k) \leq OPT_k$,

$$\Pr[|\mathcal{R}| \geq \frac{\Upsilon(\epsilon_3,\delta/3)n}{OPT_k}] \geq 1-\delta/3 \tag{45}$$

Eq. 45 says that $|\mathcal{R}| \geq \frac{\Upsilon(\epsilon_3, \delta/3)n}{OPT_k}$ with at least $(1-\delta/3)$-probability. In cases that $|\mathcal{R}| \geq \frac{\Upsilon(\epsilon_3,\delta/3)n}{OPT_k}$ happens, we use Chernoff-Hoeffding's inequality (Lem. 13) on the optimal solution, $S_k^*$, with random variable $Z = \min\{1, |R_j \cap S_k^*|\}$ and $\mu_Z = OPT_k/n$ to obtain

$$\Pr[\hat{\mathbb{I}}(S_k^*) \leq (1-\epsilon_3)OPT_k] \leq e^{-\frac{|\mathcal{R}|OPT_k\epsilon_3^2}{n2c}} \tag{46}$$

Then, due to $|\mathcal{R}| \geq \frac{\Upsilon(\epsilon_3,\delta/3)n}{OPT_k}$,

$$\Pr[\hat{\mathbb{I}}(S_k^*) \leq (1-\epsilon_3)OPT_k] \leq e^{-\frac{\Upsilon(\epsilon_3,\delta/3)\epsilon_3^2}{2c}} = \delta/3 \tag{47}$$

Thus, combining Eq. 45 and Eq. 47 gives

$$\Pr[\hat{\mathbb{I}}(S_k^*) \leq (1-\epsilon_3)OPT_k \text{ or } \hat{\mathbb{I}}(\hat{S}_k) \geq (1+\epsilon_{12})\mathbb{I}(\hat{S}_k)] \leq 2\delta/3$$

which completes the proof of Lem. 6.

## Proof of Lemma 7

From Lem. 2, to achieve the approximation described in Eq. 20, we need at least $\frac{\Upsilon(\epsilon_b,\delta_b)n}{OPT_k}$ and

$$\frac{\Upsilon(\epsilon_b,\delta_b)n}{OPT_k} \leq \mathsf{c}_1 \cdot \theta^* \tag{48}$$

Since $\theta^*$ is the sufficient number of RR sets to achieve Eq. 19 only and $N^1_{min}(\epsilon_a,\epsilon_b,\delta_a,\delta_b)$ is enough to obtain both Eq. 19 and Eq 20, we get $\theta^* \leq N^1_{min}(\epsilon_a,\epsilon_b,\delta_a,\delta_b)$ and, thus,

$$\frac{\Upsilon(\epsilon_b,\delta_b)n}{OPT_k} \leq \mathsf{c}_1 \cdot N^1_{min}(\epsilon_a,\epsilon_b,\delta_a,\delta_b) \tag{49}$$

Multiply both sides by $\frac{1}{1-1/e-\epsilon}$,

$$\frac{\Upsilon(\epsilon_b,\delta_b)n}{(1-1/e-\epsilon)OPT_k} \leq \frac{\mathsf{c}_1 \cdot N^1_{min}(\epsilon_a,\epsilon_b,\delta_a,\delta_b)}{1-1/e-\epsilon} \tag{50}$$

Since $\mathbb{I}(\hat{S}_k) \geq (1-1/e-\epsilon)OPT_k$ as a consequence of Eq, we have

$$\frac{\Upsilon(\epsilon_b,\delta_b)n}{\mathbb{I}(\hat{S}_k)} \leq \frac{\mathsf{c}_1 \cdot N^1_{min}(\epsilon_a,\epsilon_b,\delta_a,\delta_b)}{1-1/e-\epsilon} \tag{51}$$

Recall that $\hat{\mathbb{I}}(\hat{S}_k) = \frac{\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k)n}{|\mathcal{R}|}$ or $|\mathcal{R}| = \frac{\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k)n}{\hat{\mathbb{I}}(\hat{S}_k)}$. Thus, the condition $\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k) \geq \Lambda_1$ is equivalent to

$$|\mathcal{R}| \geq \frac{\Lambda_1 n}{\hat{\mathbb{I}}(\hat{S}_k)} = \frac{(1+\epsilon_b)\Upsilon(\epsilon_b,\delta_b)n}{\hat{\mathbb{I}}(\hat{S}_k)} \tag{52}$$

Since $\hat{\mathbb{I}}(\hat{S}_k) \leq (1+\epsilon_a)\mathbb{I}(\hat{S}_k)$, we also have

$$\frac{\Upsilon(\epsilon_b,\delta_b)n}{\mathbb{I}(\hat{S}_k)} \geq \frac{(1+\epsilon_b)\Upsilon(\epsilon_b,\delta_b)n}{\hat{\mathbb{I}}(\hat{S}_k)} \tag{53}$$

From Eq. 52 and Eq. 53, observe that if we set $|\mathcal{R}| = \frac{\Upsilon(\epsilon_b,\delta_b)n}{\mathbb{I}(\hat{S}_k)}$ then Eq. 52 will hold and, in turn, we achieve $\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k) \geq \Lambda_1$. Based on that and Eq. 51, we conclude that $\frac{\mathsf{c}_1 \cdot N^1_{min}(\epsilon_a,\epsilon_b,\delta_a,\delta_b)}{1-1/e-\epsilon}$ RR sets are sufficient to satisfy the condition (C1).

## Proof of Lemma 8

In this case, the necessary number of RR sets is the same as that to satisfy Eq. 19 where $\epsilon_a = \epsilon_1 + \epsilon_2 + \epsilon_1\epsilon_2$. The only question here is whether the Estimate-Inf succeeds and returns $\mathbb{I}(\hat{S}_k)$ when the main algorithm has $|\mathcal{R}|$ random RR sets. Recall that it fails in the case that the number of RR sets required in Estimate-Inf exceeds $|\mathcal{R}|\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2}$, however, from Eq. 51,

$$|\mathcal{R}| \geq \frac{\Upsilon(\epsilon_b,\delta_b)n}{\mathbb{I}(\hat{S}_k)} = 2c\ln\frac{3}{\delta}\frac{1}{\epsilon_3^2}\frac{n}{\mathbb{I}(\hat{S}_k)} \tag{54}$$

Multiply both side with $\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2}$,

$$|\mathcal{R}|\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2} \geq \frac{1+\epsilon_2}{1-\epsilon_2}2c\ln\frac{3}{\delta}\frac{1}{\epsilon_2^2}\frac{n}{\mathbb{I}(\hat{S}_k)} \tag{55}$$

Based on Lem. 2, $2c\ln\frac{3}{\delta}\frac{1}{\epsilon_2^2}\frac{n}{\mathbb{I}(\hat{S}_k)}$ random RR sets are sufficient to obtain

$$\Pr[\mathbb{I}_c(\hat{S}_k) \geq (1-\epsilon_2)\mathbb{I}(\hat{S}_k)] \leq 1-\delta/3 \tag{56}$$

Thus, combine Eq. 55 and Eq. 56,

$$\Pr[|\mathcal{R}|\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2} \geq (1+\epsilon_2)2c\ln\frac{3}{\delta}\frac{1}{\epsilon_2^2}\frac{n}{\mathbb{I}_c(\hat{S}_k)}] \leq 1-\delta/3$$

or,

$$\Pr[\frac{\mathbb{I}_c(\hat{S}_k)}{n}|\mathcal{R}|\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2} \geq (1+\epsilon_2)2c\ln\frac{3}{\delta}\frac{1}{\epsilon_2^2}] \geq 1-\delta/3$$

The left hand side, $\frac{\mathbb{I}_c(\hat{S}_k)}{n}|\mathcal{R}|\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2}$, is the coverage, Cov, in Estimate-Inf procedure when $|\mathcal{R}|\frac{1+\epsilon_2}{1-\epsilon_2}\frac{\epsilon_3^2}{\epsilon_2^2}$ RR sets are generated and the whole inequality is the condition to succeed. That means with $|\mathcal{R}|$ RR sets in the main algorithm, the Estimate-Inf returns a good estimation of $\mathbb{I}(\hat{S}_k)$ with at least $(1-\delta/3)$-probability.

## Proof of Lemma 11

To prove the lemma, we begin by showing that $\mathbb{I}(S_k^*)$ has an better approximation when $|\mathcal{R}| \geq M\mathsf{c}_1 N^2_{min}(\epsilon,\delta)$. Let $X^* = \min\{1, |R_j \cap S_k^*|\}$ ($R_j$ is a random RR set) be a random variable with $\mu_{X^*} = OPT_k/n$. Apply Lem. 2 on the approximation of $\mu_{X^*}$ to achieve Eq. 31, we have

$$\frac{\Upsilon(\epsilon_b^*,\delta_b^*)}{\mu_{X^*}} \leq \mathsf{c}_1 N^2_{min}(\epsilon,\delta) \tag{57}$$

Multiply both sides by $M$, we have

$$\frac{M\Upsilon(\epsilon_b^*,\delta_b^*)}{\mu_{X^*}} \leq M\mathsf{c}_1 N^2_{min}(\epsilon,\delta) \tag{58}$$

In other words,

$$|\mathcal{R}| \geq \frac{M\Upsilon(\epsilon_b^*,\delta_b^*)}{\mu_{X^*}} \tag{59}$$

Then, since

$$M\Upsilon(\epsilon_b^*,\delta_b^*) = \frac{\ln\frac{2}{\delta_b^*}}{\ln\frac{1}{\delta_b^*}}\frac{4(1+\epsilon_a^*)^2}{(\frac{1}{2}-\frac{1}{e}-\frac{\epsilon}{2})^3}\Upsilon(\epsilon_b^*,\delta_b^*) \geq \Upsilon(\frac{\epsilon_b^*}{2},\frac{\delta_b^*}{2}),$$

thus,

$$|\mathcal{R}| \geq \frac{\Upsilon(\frac{\epsilon_b^*}{2},\frac{\delta_b^*}{2})}{\mu_{X^*}} \tag{60}$$

Apply Lem. 2 with the number of RR sets specified in Eq. 60, we have

$$\Pr[\hat{\mathbb{I}}(S_k^*) \geq (1-\epsilon_b^*/2)OPT_k] \geq 1-\delta_b^*/2 \tag{61}$$

Combine Eq. 30 and Eq. 61 and apply Theo. 1,

$$\Pr[\mathbb{I}(\hat{S}_k) \geq (1 - \frac{1}{e} - \epsilon)OPT_k] \geq 1 - \delta_a^* - \frac{\delta_b^*}{2} \quad (62)$$

In Eq. 59, $|\mathcal{R}|$ depends on $\mu_{X^*} = OPT_k/n$ and in Eq. 62, we obtained the relation between $\mathbb{I}(\hat{S}_k)$ and $OPT_k$ and, thus, we can represent $|\mathcal{R}|$ in terms of $\mathbb{I}(\hat{S}_k)$,

$$\Pr[|\mathcal{R}| \geq \frac{M\Upsilon(\epsilon_b^*, \delta_b^*)(1 - 1/e - \epsilon)}{\mathbb{I}(\hat{S}_k)/n}] \geq 1 - \delta_a^* - \frac{\delta_b^*}{2}$$

and since $M\Upsilon(\epsilon_b^*, \delta_b^*)(1 - 1/e - \epsilon) \geq \Upsilon(\frac{\epsilon_b^*}{2}, \frac{\delta_b^*}{2})$,

$$\Pr[|\mathcal{R}| \geq \frac{\Upsilon(\frac{\epsilon_b^*}{2}, \frac{\delta_b^*}{2})}{\mathbb{I}(\hat{S}_k)/n}] \geq 1 - \delta_a^* - \frac{\delta_b^*}{2} \quad (63)$$

Notice that in each iteration D-SSA generates $|\mathcal{R}|$ new random RR sets and put them in $\mathcal{R}'$ to compute $\mathbb{I}_c(\hat{S}_k)$. Thus,

$$\Pr[|\mathcal{R}'| \geq \frac{\Upsilon(\frac{\epsilon_b^*}{2}, \frac{\delta_b^*}{2})}{\mathbb{I}(\hat{S}_k)/n}] \geq 1 - \delta_a^* - \frac{\delta_b^*}{2} \quad (64)$$

In the cases that $|\mathcal{R}'| \geq \frac{\Upsilon(\frac{\epsilon_b^*}{2}, \frac{\delta_b^*}{2})}{\mathbb{I}(\hat{S}_k)/n}$, apply Lem. 2,

$$\Pr[\mathbb{I}_c(\hat{S}_k) \geq (1 - \epsilon_b^*/2)\mathbb{I}(\hat{S}_k)] \geq 1 - \delta_b^*/2 \quad (65)$$

Combine with the probability in Eq. 64, we have

$$\Pr[\mathbb{I}_c(\hat{S}_k) \geq (1 - \epsilon_b^*/2)\mathbb{I}(\hat{S}_k)] \geq 1 - \delta_a^* - \delta_b^* \quad (66)$$

or, equivalently,

$$\Pr[\mathbb{I}(\hat{S}_k) \leq \frac{1}{1 - \epsilon_b^*/2}\mathbb{I}_c(\hat{S}_k)] \geq 1 - \delta_a^* - \delta_b^* \quad (67)$$

Thus, from Eq. 30 and Eq. 67 with a note that Eq. 30 was used to derive Eq. 67 and, thus, the probability that both happen is actually the probability in Eq. 67, we achieve

$$\Pr[\hat{\mathbb{I}}(\hat{S}_k) \leq \frac{1 + \epsilon_a^*}{1 - \epsilon_b^*/2}\mathbb{I}_c(\hat{S}_k)] \geq 1 - \delta_a^* - \delta_b^* \quad (68)$$

Then, since $\epsilon_1 = \frac{\hat{\mathbb{I}}(\hat{S}_k)}{\mathbb{I}_c(\hat{S}_k)} - 1$ and from Eq. 68,

$$\Pr[\epsilon_1 \leq \frac{1 + \epsilon_a^*}{1 - \epsilon_b^*/2} - 1] \geq 1 - \delta_a^* - \delta_b^* \quad (69)$$

and due to $\delta_a^* + \delta_b^* \leq \delta$,

$$\Pr[\epsilon_1 \leq \frac{\epsilon_a^* + \epsilon_b^*/2}{1 - \epsilon_b^*/2}] \geq 1 - \delta \quad (70)$$

which completes the proof.

## Proof of Lemma 12

Notice that since $\epsilon \leq 1 - 2/e$, we have $\frac{\epsilon_a^* + \epsilon_b^*/2}{1 - \epsilon_b^*/2} \leq \epsilon$ and the condition in Line 9 is satisfied. Given $\epsilon_1 \leq \frac{\epsilon_a^* + \epsilon_b^*/2}{1 - \epsilon_b^*/2}$, we compute $\epsilon_2, \epsilon_3$ as,

$$\epsilon_2 = \frac{\epsilon - \epsilon_1}{2(1 + \epsilon_1)} \geq \frac{1/2 - 1/e - \epsilon/2}{1 + \epsilon_a^*}\frac{\epsilon_b^*}{2} \quad (71)$$

and

$$\epsilon_3 = \frac{\epsilon - \epsilon_1}{2(1 - 1/e)} \geq (1/2 - 1/e - \epsilon/2)\frac{\epsilon_b^*}{2} \quad (72)$$

Since $\epsilon_1 \leq \frac{\epsilon_a^* + \epsilon_b^*/2}{1 - \epsilon_b^*/2}$ also implies that all the inequality used in the proof of Lem. 11 hold, then

$$\hat{\mathbb{I}}(S_k^*) \geq (1 - \epsilon_b^*/2)OPT_k \quad (73)$$

Combine with $\hat{\mathbb{I}}(\hat{S}_k) \geq (1 - 1/e)\hat{\mathbb{I}}(S_k^*)$, we have

$$\hat{\mathbb{I}}(\hat{S}_k) \geq (1 - 1/e)(1 - \epsilon_b^*/2)OPT_k \quad (74)$$

Replace $OPT_k$ in $\mu_{X^*} = OPT_k/n$ by that in the above inequality, we get $\mu_{X^*} \leq \frac{\hat{\mathbb{I}}(\hat{S}_k)}{n(1-1/e)(1-\epsilon_b^*/2)} \leq \frac{\hat{\mathbb{I}}(\hat{S}_k)}{n(\frac{1}{2} - \frac{1}{e} - \frac{\epsilon}{2})}$ and then the number of RR sets in Eq. 59 is rewritten as,

$$|\mathcal{R}| \geq \frac{\ln \frac{2}{\delta_b^*}}{\ln \frac{1}{\delta_b^*}} \frac{4(1 + \epsilon_a^*)^2}{(\frac{1}{2} - \frac{1}{e} - \frac{\epsilon}{2})^2} \frac{\Upsilon(\epsilon_b^*, \delta_b^*)}{\hat{\mathbb{I}}(\hat{S}_k)/n} \quad (75)$$

or, equivalently,

$$|\mathcal{R}|\frac{\hat{\mathbb{I}}(\hat{S}_k)}{n} \geq 2c\ln\frac{2}{\delta_b^*}\frac{4(1 + \epsilon_a^*)^2}{(\frac{1}{2} - \frac{1}{e} - \frac{\epsilon}{2})^2\epsilon_b^{*2}} \geq 2c\ln\frac{2}{\delta_b^*}\frac{1}{\epsilon_b^{*2}}$$

The left hand side is the coverage of $\hat{S}_k$ in $\mathcal{R}$, $\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k)$. Combining that and the calculated $\epsilon_1, \epsilon_2, \epsilon_3$ to find $\delta_1$,

$$\delta_1 = e^{-\frac{\mathrm{Cov}_{\mathcal{R}}(\hat{S}_k)\cdot\epsilon_3^2}{2c(1+\epsilon_1)(1+\epsilon_2)}} \leq e^{-\frac{2c\ln\frac{2}{\delta_b^*}\frac{1}{\epsilon_3^2}\cdot\epsilon_3^2}{2c(1+\epsilon_1)(1+\epsilon_2)}} \leq \delta_b^*/2 \quad (76)$$

To compute the value of $\delta_2$, we base on the fact that $|\mathcal{R}'| = |\mathcal{R}|$ and Eq. 64,

$$\delta_2 = e^{-\frac{\mathrm{Cov}_{\mathcal{R}'}(\hat{S}_k)\cdot\epsilon_2^2}{2c(1+\epsilon_2)}} \leq e^{-\frac{2c\ln\frac{2}{\delta_b^*}\frac{1}{\epsilon_2^2}\cdot\epsilon_2^2}{2c(1+\epsilon_2)}} \leq \delta_b^*/2 \quad (77)$$

From Eq. 76 and Eq. 77, we conclude

$$\delta_1 + \delta_2 \leq \delta_b^*/2 + \delta_b^*/2 = \delta_b^* \leq \delta \quad (78)$$

which proves Lem. 12.