# A Billion-scale Approximation Algorithm for Maximizing Benefit in Viral Marketing

Hung T. Nguyen, My T. Thai, *Member, IEEE,* and Thang N. Dinh†, *Member, IEEE*

**Abstract**—Online social networks have been one of the most effective platforms for marketing and advertising. Through the "world-of-mouth" exchanges, so-called viral marketing, the influence and product adoption can spread from few key influencers to billions of users in the network. To identify those key influencers, a great amount of work has been devoted for the Influence Maximization (IM) problem that seeks a set of $k$ seed users that maximize the expected influence. Unfortunately, IM encloses two impractical assumptions: 1) any seed user can be acquired with the same cost, 2) all users are equally interested in the advertisement. In this paper, we propose a new problem, called *Cost-aware Targeted Viral Marketing* (CTVM), to find the most cost-effective seed users who can influence the most relevant users to the advertisement. Since CTVM is NP-hard, we design an efficient $(1-1/\sqrt{e}-\epsilon)$-approximation algorithm, named BCT, to solve the problem in billion-scale networks. Comparing with IM algorithms, we show that BCT is both theoretically and experimentally faster than the state-of-the-arts while providing better solution quality. Moreover, we prove that under the Linear Threshold model, BCT is the first *sub-linear time* algorithm for CTVM (and IM) in dense networks. We carry a comprehensive set of experiments on various real-networks with sizes up to several billion edges in diverse disciplines to show the absolute superiority of BCT on both CTVM and IM domains. Experiments on Twitter dataset, containing 1.46 billions of social relations and 106 millions tweets, show that BCT can identify key influencers in trending topics in only few minutes.

**Index Terms**—Viral Marketing, Influence Maximization, Sampling Alg.

## 1 Introduction

With billions of active users, Online social networks (OSNs) such as Facebook, Twitter and LinkedIn have become critical platforms for marketing and advertising. Through the "word-of-mouth" exchanges, information, innovation, and brand-awareness can disseminate widely over the network. Many notable examples includes the ALS Ice Bucket Challenge, resulting in more than 2.4 million uploaded videos on Facebook and $98.2m donation to the ALS Association in 2014; the customer initiative #PlayItForward of ToyRUs on Twitter that draws more than $35.5m; and the unrest in many Arab countries in 2012. Despite the huge economic and political impact, viral marketing in billion-scale OSNs is still a challenging problem due to the huge numbers of users and social interactions.

†*Corresponding author email* tndinh@vcu.edu
*H. T. Nguyen and T. N. Dinh are currently with the Department of Computer Science, Virgina Commonwealth University, Richmond, VA, 20284 USA. Email: {hungnt,tndinh}@vcu.edu.*
*M. T. Thai is with the Department of Computer Science and Information Science and Engineering, University of Florida, Gainesville, FL, 32611 USA. Email: mythai@cise.ufl.edu.*

A central problem in viral marketing is the *Influence Maximization* (IM) problem that seeks a seed set of $k$ *influential* individuals in a social network that can (directly and indirectly) influence the maximum number of people. Kempe et al. [1] was the first to formulate IM as a combinatorial optimization problem on the two pioneering diffusion models, namely, *Independent Cascade* (IC) and *Linear Threshold* (LT). Since IM is NP-hard, they 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]–[8].

Unfortunately, the formulation of viral marketing as the IM problem encloses two impractical assumptions: 1) any seed user can be acquired with the same cost and 2) the same benefit obtained when influencing one user. The first assumption implies that incentivizing high-profile individuals costs the same as incentivizing common users. This often leads to impractical solutions with unaffordable seed nodes, e.g., the solutions in Twitter often include celebreties like Katy Perry or President Obama. The second assumption can mislead the company to influence "wrong audience" who are neither interested nor potentially profitable. In practice, companies often target not all users but specific sets of potential customers, decided by the factors like age and gender. Moreover, the targeted users can bring different amount of benefit to the company. Thus, simply counting the number of influenced users, as in the case of IM, does not measure the true impact of the campaign and lead to the choosing of wrong seed set. A few recent works attempt to address the above two issues *separately*. In [9] the authors study the *Budgeted Influence Maximization* (BIM) that considers an arbitrary cost for selecting a node and propose an $(1-1/\sqrt{e}-\epsilon)$ approximation algorithm for the problem. However, their algorithm is not scalable enough for billion-scale networks. Recently, there is a serial works in [10], [11] investigating the *Targeted Viral Marketing* (TVM) problem, in which they attempt to influence a subset of users in the network. Unfortunately, all of these methods rely on heuristics strategy and provide no performance guarantees.

In this paper, we introduce the *Cost-aware Targeted Viral Marketing* (CTVM) problem which takes into account both arbitrary cost for selecting a node and arbitrary benefit for influencing a node. Given a social network abstracted by a graph $G = (V, E)$, each node $u$ represents a user with a *cost* $c(u)$ to select into the seed set and a *benefit* $b(u)$ obtained when $u$ is influenced. Given a budget $B$, the

goal is to find a seed set $S$ with total cost at most $B$ that maximizes the expected total benefit over the influenced nodes. CTVM is more relevant in practice as it generalizes other viral marketing problems including TVM, BIM and the fundamental IM. However, the problem is much more challenging with heterogeneous costs and benefits. As we show in Section 3, extending the state-of-the-art method for IM in [8] may increase the running time by a factor $|V|$, making the method unbearable for large networks.

We introduce BCT, an efficient approximation algorithm for CTVM for billion-scale networks. Given arbitrarily small $\epsilon > 0$, our algorithm guarantees a $(1 - 1/\sqrt{e} - \epsilon)$-approximate solution in general case and a $(1 - 1/e - \epsilon)$-approximate solution when nodes have uniform costs. BCT also dramatically outperforms the existing state-of-the-art methods for IM, e.i., IMM, TIM/TIM+, when nodes have uniform costs and benefits. In particular, BCT only takes several minutes to process a network with 41.7 million nodes and 1.5 billion edges.

Our contributions are summarized as follows:

- We propose the *Cost-aware Targeted Viral Marketing* (CTVM) problem that consider *heterogeneous costs and benefits* for nodes in the network. Our problem generalizes other viral marketing problems including TVM, BIM, and the fundamental IM problems.
- We propose BCT, an efficient algorithm that returns $(1-1/\sqrt{e}-\epsilon)$-approximate solutions for CTVM with a high probability. Moreover, the algorithm meets the theoretical thresholds in [14] on the sufficient number of samples. Also, BCT is a sub-linear time algorithm for CTVM (and IM) in dense graphs, under the LT model.
- We provide extensive experiments on various real networks. The experiments suggest that BCT, considering both cost and benefit, provides significantly higher quality solutions than existing methods, while running several times faster than the state-of-the art ones. Further, we also demonstrate the ability of BCT to identify key influencers in trending topics in a Twitter dataset of 1.5 billion social relations and 106 million tweets within few minutes. The experiments also indicate that BCT is robust against noise and various schemes to assign node costs.

**Related works.** Kempe et al. [1] is the first to formulate IM as an optimization problem. They show the problem to be NP-complete and devise an $(1 - 1/e - \epsilon)$ approximation algorithm. Also, IM cannot be approximated within a factor $(1 - \frac{1}{e} + \epsilon)$ [17] under a typical complexity assumption. 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 speed up.

Several heuristics are developed to derive 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 speed up 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 works on learning the parameters of influence propagation models [20], [21].

Recently, Borgs et al. [13] 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. Tang et al. [8] reduces the running time to $O((k + l)(m + n)\log n/\epsilon^2)$ and present the first algorithm that is scalable for billion-size networks. The key result is that only $\theta = (8 + 2\epsilon)n\frac{\ln 2/\delta + \ln\binom{n}{k}}{\epsilon^2 OPT_k}$ samples (RR sets) are needed to guarantee $(1 - 1/e - \epsilon)$ approximate solution.

In the Tang et al. [8], a remaining challenge is to estimate the unknown $OPT_k$, the maximum influence spread. The two heuristics TIM/TIM+ provided in [8] may incur many times more samples, thus, are not efficient enough for large networks. Compared with TIM/TIM+ on IM problem, our conference paper in [22], based on a different approach of stopping condition, both improves the theoretical threshold of the sufficient number of samples to guarantee $(1 - 1/e - \epsilon)$-solution quality by a factor of more than 6 and guarantees that our actual number of generated samples concentrated around a small constant times the threshold while TIM/TIM+ fail to provide any guarantee. Independently with our work, IMM, proposed in [14], follows the framework of TIM/TIM+ but improve further the threshold by a factor of up to 5. IMM also provides a guarantee on achieving some constant times the threshold, however, the guarantee is quite loose. That is their expected number of samples is at least 3 times larger than the theoretical threshold. In this paper, we adopt a stopping condition approach and prove that our method theoretically and practically surpasses all other methods.

In another direction, Nguyen and Zheng [15] investigate the BIM problem in which each node can have an arbitrary selecting cost. They proposed a $(1 - 1/\sqrt{e} - \epsilon)$ approximation algorithm (called BIM) based on a greedy algorithm for Budgeted Max-Coverage in [23] and two other heuristics. However, the greedy algorithm relies on massive simulations and thus, severely suffers from scalability while the two heuristics have no approximation guarantee. In contrast, we aim towards efficient approximation algorithm with $1-1/e-\epsilon$ guarantee running on billion-scale networks. Instead of massive simulation, we employ the advanced technique of reverse influence sampling combined with the optimal stopping condition in Monte Carlo estimation [24].

A line of works in [10], [11], [25] consider Topic-aware Influence Maximization problem in which edges are associated with a topic-dependent user-to-user social influence strengths. The problem also asks for a set of $k$ users that maximize user adoptions. However, all of the proposed methods do not possess any theoretical guarantee on the solution quality. For a comprehensive overview, we provide the summary of the related algorithms in Table 1.

There have also been a number of interesting studies on area related to our problem. [26]–[28] focus on designing

TABLE 1: Main results of related methods ($k$ is the number of seed nodes, $n, m$ are the numbers of edges and edges, Approx. indicates whether the algorithm provides guarantee i.e., $1 - 1/e - \epsilon$ optimal with $(1 - \delta)$-probability where $\delta = 1/n^l$).

| Method | IM | BIM | TIM | CTVM | Approx. | Model | Time Complexity |
|---|---|---|---|---|---|---|---|
| Naive Greedy [1] | ✓ | | | | ✓ | LT+IC | $O(kmnR)$, $R$ is the #Monte Carlo simulations (typically 10000) |
| CELF [12] | ✓ | | | | ✓ | LT+IC | $O(kmnR)$, empirically faster than Naive Greedy |
| CELF++ [5] | ✓ | | | | ✓ | LT+IC | $O(kmnR)$, optimized CELF |
| Simpath [4] | ✓ | | | | | LT | $O(kmnR)$, empirically faster than Naive Greedy |
| LDAG [3] | ✓ | | | | | MIA | $O(n^2 + kn^2 \log(n))$ (see [3] for details) |
| Borgs's method [13] | ✓ | | | | ✓ | LT+IC | $O(kl^2(m+n)\log^2(n)/\epsilon^3)$ |
| TIM+ [8], IMM [14] | ✓ | | | | ✓ | LT+IC | $O((k+l)(n+m)\log(n)/\epsilon^2)$ |
| BIM [15] | | ✓ | | | ✓ | LT+IC | $O(n^2(\log(n)+d) + kn(1+d))$ ($d$ is max in-degree) |
| KB-TIM [16] | | | ✓ | | ✓ | IC | – |
| BCT (this paper) | ✓ | ✓ | ✓ | ✓ | ✓ | LT+IC | $\begin{cases} O((k+l)n\log(n)/\epsilon^2 & \text{for the LT model} \\ O((k+l)(n+m)\log(n)/\epsilon^2) & \text{for the IC model} \end{cases}$ |

data mining or machine learning algorithms to extract influence cascade model parameters from real datasets. [29] study the problem of adaptive seeding which, given a budget, asks for policy of selecting nodes in two stages to maximize the total influence: a portion of the given budget is spent on the first stage to reveal more available nodes for seeding and the rest is used for selecting nodes later.

**Organization.** The rest of the paper is organized as follows. In Section 2, we present network model, propagation models, and the problem definitions. Section 3 presents our BCT algorithm for CTVM. We analyze BCT approximation factor and time complexity in Section 4. Experimental results are shown in Section 5. We conclude in Section 6.

## 2 Models and Problem Definitions

In this section, we formally define the CTVM problem and present an overview of the Reverse Influence Sampling approaches in Borgs et al. [13] and Tang et al. [8], [14]. For readability, we focus on the *Linear Threshold* (LT) propagation model [1] and summarize our similar results for the *Independent Cascade* (IC) model in Subsection 4.5.

### 2.1 Model and Problem Definition

Let $G = (V, E, c, b, w)$ be a social network with a node set $V$ and a directed edge set $E$, with $|V| = n$ and $|E| = m$. Each node $u \in V$ has a selecting cost $c(u) \geq 0$ and a benefit $b(u)$ if $u$ is influenced. Each directed edge $(u, v) \in E$ is associated with an influence weight $w(u, v) \in [0, 1]$ such that $\sum_{u \in V} w(u, v) \leq 1$.

Our model assumes that all the parameters, $c(u), b(u)$ $\forall u \in V$ and $w(u, v)$ $\forall (u, v) \in E$ are given. In fact, these can be estimated depending on the specific context when applying our method. The cost of node $u$, $c(u)$, manifests how hard (how much effort) it is to initially influence the respective person, e.g., convince him to adopt the product. Thus, $c(u)$ is usually regarded proportionally to some centrality measures, e.g., the degree centrality [15].

Similarly, the node benefit $b(u)$ refers to the gain of influencing node $u$ and hence is context-dependent, e.g., in targeted viral marketing, $b(u)$ is assigned 1 if $u$ is in our targeted group and 0 outside [10], [11] or learned from the interest level on the relevant topic, e.g., number of tweets/retweets with specific keywords on Twitter network. Additionally, $w(u, v)$ indicates the probability of $u$ influencing $v$ which is widely evaluated as the interaction frequency from $u$ to $v$ [1], [8] or learned from action logs [28].

Given a graph $G$ and a subset $S \subset V$, referred to as the *seed set*, in the LT model the influence cascades in $G$ as follows. First, every node $v \in V$ independently selects a *threshold* $\lambda_v$ uniformly at random in $[0, 1]$. Next the influence propagation happens in round $t = 1, 2, 3, \ldots$

- At round 1, we *activate* nodes in the seed set $S$ and set all other nodes *inactive*. The cost of activating the seed set $S$ is given $c(S) = \sum_{u \in S} c(u)$.
- At round $t > 1$, an inactive node $v$ is activated if the weighted number of its activated neighbors reaches its threshold, i.e., $\sum_{\text{active neighbor } u} w(u, v) \geq \lambda_v$.
- Once a node becomes activated, it remains activated in all subsequent rounds. The influence propagation stops when no more nodes can be activated.

Denote by $\mathbb{I}(S)$ the expected number of activated nodes given the seed set $S$, when the expectation is taken among all $\lambda_v$ values from their uniform distributions. We call $\mathbb{I}(S)$ the *influence spread* of $S$ in $G$ under the LT model.

The LT is shown in [1] to be equivalent to the reachability in a random graph $g$, called *live-edge graph* or *sample graph*, defined as follows: Given a graph $G = (V, E, w)$, for every $v \in V$, select at most one of its incoming edges at random, such that the edge $(u, v)$ is selected with probability $w(u, v)$, and no edge is selected with probability $1 - \sum_u w(u, v)$. The selected edges are called *live* and all other edges are called *blocked*. By claim 2.6 in [1], the influence spread of a seed set $S$ equals the expected number of nodes reachable from $S$ over all possible sample graphs, i.e.,

$$\mathbb{I}(S) = \sum_{g \sqsubseteq G} \Pr[g] |R(g, S)|, \tag{1}$$

where $\sqsubseteq$ denotes that the sample graph $g$ is generated from $G$ with a probability denoted by $\Pr[g]$, and $R(g, S)$ denotes the set of nodes reachable from $S$ in $g$.

Similarly, the *benefit* of a seed set $S$ is defined as the expected total benefit over all influenced nodes, i.e.,

$$\mathbb{B}(S) = \sum_{g \sqsubseteq G} \Pr[g] \sum_{u \in R(g,S)} b(u). \tag{2}$$

We are now ready to define our problem as follows.

**Definition 1** (Cost-aware Targeted Viral Marketing -CTVM). *Given a graph $G = (V, E, c, b, w)$ and a budget $B > 0$, find a seed set $S \subset V$ with total cost $c(S) \leq B$ to maximize $\mathbb{B}(S)$.*

CTVM generalizes the viral marketing problems:

- Influence Maximization (IM): IM is a special case of CTVM with $c(u) = 1$ and $b(u) = 1$ $\forall u \in V$.
- Budgeted Influence Maximization (BIM) [15]: find a seed set with total cost at most $B$, that maximizes $\mathbb{I}(S)$. That is $b(u) = 1$ $\forall u \in V$.

- Targeted Viral Marketing (TVM): find a set of $k$ node to maximize the number of influenced nodes in a targeted set $T$. This is $c(u) = 1\ \forall u \in V$ and benefits $c(v) = 1$ if $v \in T$, and $c(w) = 0$ otherwise.

Since IM is a special case of CTVM, CTVM inherits the IM's complexity and hardness of approximation. Thus CTVM is an NP-hard problem and cannot be approximated within a factor $1 - 1/e + \epsilon$ for any $\epsilon > 0$, unless $P = NP$.

In Table 2, we summarize the frequently used notations.

TABLE 2: Table of Notations

| Notation | Description |
|---|---|
| $n, m$ | #nodes, #links in $G$, respectively |
| $\mathbb{I}(S), \mathbb{I}(S,u)$ | Influence Spread of seed set $S \subseteq V$ and influence of $S$ on a node $v$. For $v \in V$, $\mathbb{I}(v) = \mathbb{I}(\{v\})$ |
| $\Gamma$ | Sum of all node benefits, $\sum_{v \in V} b(v)$ |
| $\mathbb{B}(S)$ | Benefit of seed set $S \subseteq V$ |
| $\hat{\mathbb{B}}(S)$ | $\hat{\mathbb{B}}(S) = \frac{deg_{\mathcal{H}}(S)}{m_{\mathcal{H}}}\Gamma$ - an estimator of $\mathbb{B}(S)$ |
| $\mathsf{OPT}_k$ | The maximum $\mathbb{B}(S)$ for any size-$k$ seed set $S$ |
| $S_k^*$ | An optimal size-$k$ seed node, $\mathbb{B}(S_k^*) = OPT_k$ |
| $m_{\mathcal{H}}$ | #hyperedges in hypergraph $\mathcal{H}$ |
| $deg_{\mathcal{H}}(S)$, $S \subseteq V$ | #hyperedges incident at some node in $S$. Also, $deg_{\mathcal{H}}(v)$ for $v \in V$ |
| $\alpha$ | $\alpha = \sqrt{\ln(1/\delta) + \ln 2}$ |
| $\beta$ | $\beta = \sqrt{(1-1/e) \cdot (\ln \binom{n}{k} + \ln(1/\delta) + \ln 2)}$ |
| $\epsilon_2$ | $\epsilon_2 = \frac{\epsilon\beta}{((1-1/e)\alpha+\beta)}$ |
| $\Lambda_L$ | $\Lambda_L = (1+\epsilon_2)\frac{(2+2\epsilon_2/3)\Gamma(\ln(6/\delta)+\ln\binom{n}{k})}{\epsilon_2^2}$ |

### 2.2 Summary of the RIS Approach

The major bottle-neck in previous methods for IM [1], [2], [4], [15] is the inefficiency in estimating the influence spread. To address this, Borgs et al. [13] introduced a novel approach for IM, called Reverse Influence Sampling (RIS), which is the foundation for TIM/TIM+ algorithms, the state-of-the-art methods for IM [8].

Given a graph $G = (V, E, c, b, w)$, RIS captures the influence landscape of $G$ through generating a hypergraph $\mathcal{H} = (V, \{\mathcal{E}_1, \mathcal{E}_2, \ldots\})$. Each hyperedge $\mathcal{E}_j \in \mathcal{H}$ is a subset of nodes in $V$ and constructed as follows.

**Definition 2** (Random Hyperedge)**.** *Given $G = (V, E, w)$, a random hyperedge $\mathcal{E}_j$ is generated from $G$ by 1) selecting a random node $v \in V$ 2) generating a sample graph $g \sqsubseteq G$ and 3) returning $\mathcal{E}_j$ as the set of nodes that can reach $v$ in $g$.*

Node $v$ in the above definition is called the *source* of $\mathcal{E}_j$ and denoted by $\mathsf{src}(\mathcal{E}_j)$. Observe that $\mathcal{E}_j$ contains the nodes that can influence its source $v$. If we generate multiple random hyperedges, influential nodes will likely appear more often in the hyperedges. Thus a seed set $S$ that *covers* most of the hyperedges will likely maximize the influence spread $\mathbb{I}(S)$. Here a a seed set $S$ covers a hyperedge $\mathcal{E}_j$, if $S \cap \mathcal{E}_j \neq \emptyset$. This is captured in the following lemma in [13].

We denote by $m_{\mathcal{H}}$ the number of hyperedges in $\mathcal{H}$.

**Lemma 1.** *[13] Given $G = (V, E, w)$ and a random hyperedge $\mathcal{E}_j$ generated from $G$. For each seed set $S \subset V$,*

$$\mathbb{I}(S) = n \Pr[S \text{ covers } \mathcal{E}_j]. \tag{3}$$

**RIS framework.** Based on the above lemma, the IM problem can be solved using the following framework.

- Generate multiple random hyperedges from $G$
- Use the greedy algorithm for the Max-coverage problem [23] to find a seed set $S$ that covers the maximum number of hyperedges and return $S$ as the solution.

**Thresholds for Sufficient Number of Samples.** The core issue in applying the above framework is that: *How many hyperedges are sufficient to provide a good approximation solution?* For any $\epsilon, \delta \in (0, 1)$, Tang et. al. established in [8] a theoretical threshold

$$\theta = (8 + 2\epsilon)n\frac{\ln 2/\delta + \ln \binom{n}{k}}{\epsilon^2 \mathsf{OPT}_k}, \tag{4}$$

and proved that when the number of hyperedges in $\mathcal{H}$ reaches $\theta$, the above framework returns an $(1 - 1/e - \epsilon)$-approximate solution with probability $1 - \delta$. Here $\mathsf{OPT}_k$ denotes the maximum influence spread $\mathbb{I}(S)$.

Unfortunately, computing $\mathsf{OPT}_k$ is intractable, thus, TIM/TIM+ in [8] have to approximate $\mathsf{OPT}_k$ by a heuristic $KPT^+$ and thus, generate $\theta\frac{\mathsf{OPT}_k}{KPT^+}$ hyperedges, where the ratio $\frac{\mathsf{OPT}_k}{KPT^+} \geq 1$ is not upper-bounded. That is TIM/TIM+ may generate many times more hyperedges than needed. In contrast, our BCT algorithm in Section 3 guarantees that the number of hyperedges is at most a constant time of the theoretical threshold (with high probability). Thus, its running time is smaller and more predictable.

The same group of authors further reduce the threshold $\theta$ (Theorem 1 in [14]) to,

$$\theta = \frac{2n \cdot ((1-1/e) \cdot \alpha + \beta)^2}{\mathsf{OPT}_k \cdot \epsilon^2}, \tag{5}$$

where

$$\alpha = \sqrt{\ln(1/\delta) + \ln 2}, \text{ and} \tag{6}$$

$$\beta = \sqrt{(1-1/e) \cdot (\ln \binom{n}{k} + \ln(1/\delta) + \ln 2)}. \tag{7}$$

Define

$$\epsilon_2 = \frac{\epsilon\beta}{(1-1/e)\alpha + \beta}, \tag{8}$$

then, the threshold $\theta$ can be rewritten as follows,

$$\theta = \frac{2n\beta^2}{\mathsf{OPT}_k\epsilon_2^2} = \frac{(2-2/e)n(\ln \binom{n}{k} + \ln(1/\delta) + \ln 2)}{\mathsf{OPT}_k\epsilon_2^2} \tag{9}$$

which is shown in [14] to be 5 times smaller than that of Eq. 4. IMM also improves the estimation of $KPT^+$ to be bounded by some constant times $OPT_k$ with high probability. However, the bound is loose and the estimation process is complicated. On the other hand, the proposed BCT algorithm in this paper adopts the better threshold in [14] with our approach in [22] which: 1) avoids a possibly complicated and expensive estimation phase, 2) achieves a better bound on the actual number of samples and 3) solves the more general CTVM problem (covers IM problem).

**Remark.** The most intuitive way to extend the RIS framework to cope with benefit of the nodes is to modify the RIS framework to find a seed set $S$ that covers the maximum *weighted* number of hyperedges, where the weight of a hyperedge $\mathcal{E}_j$ is the benefit of the source $\mathsf{src}(\mathcal{E}_j)$. However following the same analysis in Tang et al. [8], [14], we need

$$\theta_B = \theta\mathbf{b_{max}}, \tag{10}$$

where $b_{max} = \max\{b(u)|u \in V\}$.

Unfortunately, $\theta_B$ can be as large as $n$ times $\theta$ in the worst-case. To see this, we can (wlog) normalize the node benefit $b(u)$ so that $\sum_{u \in V} b(u) = n$. Then note that $b_{max}$ could be as large as $\sum_{u \in V} b(u) = n$.

## 3 BCT Approximation Algorithm

In this section, we present BCT - a scalable approximation algorithm for CTVM. BCT combines two novel techniques: BSA (Alg. 1), a sampling strategy to estimate the benefit and a powerful stopping condition to smartly detect when the sufficient number of hyperedges is reached.

**Algorithm 1** BSA - Benefit Sampling Alg. for LT model

**Input**: Weighted graph $\mathcal{G} = (V, E, w)$.
**Output**: A random hyperedge $\mathcal{E}_j \subseteq V$.
1: $\mathcal{E}_j \leftarrow \emptyset$;
2: Pick a node $u$ with probability $\frac{b(u)}{\Gamma}$;
3: **repeat**
4: Add $u$ to $\mathcal{E}_j$;
5: Attempt to select an edge $(v, u)$ using live-edge model;
6: **if** edge $(v, u)$ is selected **then** Set $u \leftarrow v$;
7: **until** $(u \in \mathcal{E}_j)$ OR (no edge is selected);
8: **return** $\mathcal{E}_j$;

### 3.1 Efficient Benefit Sampling Algorithm - BSA

Due to the inefficiency of RIS when applying to CTVM problem, we propose a generalized version of RIS, called Benefit Sampling Algorithm - BSA, for estimating benefit $\mathbb{B}(S)$. The BSA for generating a random hyperedge $\mathcal{E}_j \subseteq V$ under LT model is summarized in Algorithm 1. A similar BSA procedure for IC model can be derived by changing the generating of live-edges in the Lines 5 and 6 of Algorithm 1 to the equivalent live-edge model for IC [1]. The great deal of difference of BSA from RIS is that it *chooses the source node proportional to benefit of each node* as opposed to choosing uniformly at random in RIS. That is the probability of choosing node $u$ is $P(u) = b(u)/\Gamma$ with $\Gamma = \sum_{v \in V} b(v)$. After choosing a starting node $u$, it attempts to select an *in-neighbor* $v$ of $u$ according to the LT model and make $(v, u)$ a *live* edge. Then it "moves" to $v$ and repeat the process. The procedure stops when we encounter a previously visited vertex or no edge is selected. The hyperedge is the set of nodes visited along the process.

Note that the selection of a source node with the probability proportional to the benefit can be done in $O(1)$ after an $O(n)$ preprocessing using the Alias method [30]. Similarly, the selection of the live edge according to the influence weight can also be done in $O(1)$. In contrast, in the IC model [13], it takes a time $\theta(d(v))$ at a node $v$ to generate all live edges pointing to $v$. *This key difference makes the generating hyperedges in the LT model more efficient than that in the IC.*

The key insight into why random hyperedges generated via BSA can capture the benefit landscape is stated in the following lemma.

**Lemma 2.** *Given a fixed set $S \subseteq V$, for a random hyperedge $\mathcal{E}$,*

$$\Pr_{g \sqsubseteq G, u \in V}[\mathcal{E}_j \cap S \neq \emptyset] = \frac{\mathbb{B}(S)}{\Gamma}. \tag{11}$$

The above lemma on computing benefit is similar to Lemma 1 on influence except having the *normalizing constant* $\Gamma$ *in the place of* $n$ in Lemma 1. Thus, the RIS framework can be applied and a similar result to Theorem 1 in [14] on the threshold of hyperedges can be derived as follows.

**Corollary 1.** *Let*

$$\theta_B(\epsilon, \delta) = \frac{(2 - 2/e)\Gamma(\ln \binom{n}{k} + \ln(1/\delta) + \ln 2)}{\mathsf{OPT}_k \epsilon_2^2}. \tag{12}$$

*For any fixed $T \geq \theta_B$, the RIS framework with $T$ random hyperedges, generated by BSA, will return an $(1 - 1/e - \epsilon)$-approximate solution for the CTVM problem.*

### 3.2 Solving Budgeted Max-Coverage Problem

Finding a candidate seed set $\hat{S}_k$ that appears most frequently in the hyperedges is a special version of the Budgeted Max-Coverage problem [31]. Each hyperedge represents an element in the *Budgeted Max-Coverage* problem and each node $v \in V$ is associated with a subset of hyperedges that contains $v$. The cost to select a subset is given by the cost to select the corresponding node into the seed set.

We use the greedy algorithm, denoted by Budgeted-Max-Coverage, in [31] to find a maximum covering set within the budget $B$ is applied. This procedure considers two candidates and chooses the one with higher coverage. The first one is taken from greedy strategy which sequentially selects nodes with highest efficiency, i.e. ratio between marginal coverage gain and its cost of selecting,

$$\forall i = 1..k, v_i = \arg \max_{v \in V \setminus S_{i-1}} \Delta(S_{i-1}, v), S_i = S_{i-1} \cup \{v_i\}$$

where $\Delta(S_{i-1}, v) = \frac{\mathrm{Cov}(S_{i-1} \cup \{v\}) - \mathrm{Cov}(S_{i-1})}{c(v)}$ and $\mathrm{Cov}(S_{i-1})$ is the number of hyperedges incident to at least a node in $S_{i-1}$. The second solution is just a node with highest coverage within the budget. [31] proved that this procedure returns a $(1 - 1/\sqrt{e})$-approximate cover if the nodes' cost are non-uniform, or, $(1 - 1/e)$-approximate cover, otherwise.

Note that we can improve the approximation ratio to $(1 - 1/e)$ for the case of non-uniform costs, however, the time complexity ($\Omega(n^4)$) becomes impractical.

**Algorithm 2** BCT Algorithm

**Input**: Graph $G = (V, E, b, c, w)$, budget $B > 0$, and two precision parameters $\epsilon, \delta \in (0, 1)$.
**Output**: $\hat{S}_k$ - An $(1 - 1/e - \epsilon)$-approximate seed set.
1: $\Lambda_L = (1 + \epsilon_2)\frac{2 + 2\epsilon_2/3}{2 - 2/e}\theta_B(\epsilon, \delta/3)\mathsf{OPT}_k$
(Or $\Lambda_L = \Lambda_L^{cost}$ in Eq. 21 for non-uniform costs;)
2: $N_t = \Lambda_L$; $\mathcal{H} \leftarrow (V, \mathcal{E} = \emptyset)$; $t \leftarrow 0$;
3: **repeat**
4: **for** $j = 1$ **to** $N_t - |\mathcal{E}|$ **do**
5: Generate $\mathcal{E}_j \leftarrow$BSA($\mathcal{G}$); Add $\mathcal{E}_j$ to $\mathcal{E}$;
6: **end for**
7: $t \leftarrow t + 1$; $N_t = 2N_{t-1}$;
8: $\hat{S}_k =$ Budgeted-Max-Coverage($\mathcal{H}$, $B$);
9: **until** $deg_{\mathcal{H}}(\hat{S}_k) \geq \Lambda_L$;
10: **return** $\hat{S}_k$;

### 3.3 BCT - The Main Algorithm

BCT algorithm for the CTVM problem is presented in Algorithm 2. The algorithm uses BSA (Algorithm 1) to generate hyperedges and Budgeted-Max-Coverage [22] to find a candidate seed set $\hat{S}_k$ following the RIS framework.

BCT keeps generating hyperedges by BSA sampling (Algorithm 1) until the degree of the seed set selected by Budgeted-Max-Coverage exceeds a threshold $\Lambda_L$ (the stopping condition). Specifically, at iteration $1 \leq t \leq O(\log n)$, it consider the hypergraph $\mathcal{H}$ that consists of the first $2^{t-1}\Lambda_L$ hyperedges. That is the number of samples (aka hyperedges) are double after each iteration. In each iteration, Budgeted-Max-Coverage algorithm is called to select a seed set $\hat{S}_k$ within the budget $B$ and stops the algorithm if the degree of $\hat{S}_k$ exceeds $\Lambda_L$, $deg_{\mathcal{H}}(\hat{S}_k) \geq \Lambda_L$. Otherwise, it advances to the next iteration.

## 4 Approximation and Complexity Analysis

We prove that BCT will stop within $O(\theta_B)$ samples (aka hyperedges) and return an $(1-1/e-\epsilon)$-approximate solution.

Note that BCT can be used with any threshold for the sufficient number of samples (not only the one in [14]). That is if a better threshold $\theta' < \theta$ exists, we can use $\theta'$ in BCT to guarantee BCT will stop within $O(\theta')$ samples whp.

### 4.1 Approximation Guarantee for uniform cost CTVM

Assume the case of uniform node cost. The proof consists of two steps: 1) the "stopping time" (aka the number of hyperedges) $m_{\mathcal{H}}$ concentrates on an interval $[T^*, \mathsf{c}T^*]$ for some fixed $\mathsf{c} > 4$ (Lemma 3 and 5); and 2) for that interval the candidate seed set $\hat{S}_k$ is a $(1-1/e-\epsilon)$-approximate solution whp (Lemma 4).

Given a seed set $S \subset V$, denote by $\hat{\mathbb{B}}_T(S)$ and $deg_T(S)$ the estimate of $\mathbb{B}(S)$ and the degree of $S$ of the hypergraph with the first $T$ random hyperedges, respectively.

**Lemma 3.** *Let $T^* = \frac{2+2\epsilon_2/3}{2-2/e}\theta_B(\epsilon, 2\delta_2) = \frac{\Lambda_L \Gamma}{(1+\epsilon_2)\mathsf{OPT}_k}$ hyperedges, where $\epsilon_2$ is defined in Eq. 8 and $\delta_2 = \delta/6$. We have,*

$$\Pr[m_{\mathcal{H}} \leq T^*] \leq \delta_2. \tag{13}$$

Let $t_0 = \left\lceil \log_2 \frac{T^*}{\Lambda_L} \right\rceil + 1$ be the smallest iteration such that $2^{t_0-1}\Lambda_L \geq T^*$. The above lemma is equivalent to,

$$\Pr[t < t_0] \leq \delta_2. \tag{14}$$

For iterations $t \geq t_0$, we now show that the candidate solution $\hat{S}_k$ will be an $(1-\frac{1}{e}-\epsilon)$-approximate solution whp.

**Lemma 4.** *For any iteration $t \geq t_0$, the candidate solution $\hat{S}_k$ satisfies that*

$$\Pr[\mathbb{B}(\hat{S}_k) \leq (1-1/e-\epsilon)\mathsf{OPT}_k] \leq (2\delta_2)^{2^{t-t_0}}. \tag{15}$$

*Proof.* This is a direct consequence of Corollary 1. We can verify that the number of samples in iteration $t$ is

$$|\mathcal{E}| = 2^{t-1}\Lambda_L \geq 2^{t-t_0}\theta_B(\epsilon, 2\delta_2) \geq \theta_B(\epsilon, (2\delta_2)^{2^{t-t_0}}). \tag{16}$$

This yields the proof. □

The upper-bound on the number of hyperedges generated by BCT is stated in the following lemma.

**Lemma 5.** *For $\epsilon \in (0, 1-1/e)$ and $\mathsf{c} = 4\left\lceil \frac{1+\epsilon_2}{1-1/e-\frac{\epsilon+\epsilon_2}{2}} \right\rceil$,*

$$\Pr[m_{\mathcal{H}} \geq \mathsf{c}T^*] \leq \delta_2. \tag{17}$$

Finally, we prove the overall approximation guarantee of BCT in the following Theorem 1.

**Theorem 1.** *Given $0 < \epsilon < 1-1/e, 0 < \delta < 1$,*
$\Pr[m_{\mathcal{H}} = O(\theta_B(\epsilon,\delta))$ *and* $\mathbb{B}(\hat{S}_k) \geq (1-\frac{1}{e}-\epsilon)\mathsf{OPT}_k] \geq 1-\delta$.

*Proof.* Assume that none of the following "bad" events in Lemmas 3, 5 and 4 happens.

(b1) $\Pr[m_{\mathcal{H}} \leq T^*] \leq \delta_2$
(b2) $\Pr[m_{\mathcal{H}} \geq \mathsf{c}T^*] \leq \delta_2$
(b3) $\forall t \geq t_0, \Pr[\mathbb{B}(\hat{S}_k) \leq (1-1/e-\epsilon)\mathsf{OPT}_k] \leq (2\delta_2)^{2^{t-t_0}}$

That is the following inequalities

(i1) $m_{\mathcal{H}} \geq T^*$,
(i2) $m_{\mathcal{H}} \leq \mathsf{c}T^*$, and
(i3) $\forall t \geq t_0, \mathbb{B}(\hat{S}_k) \geq (1-1/e-\epsilon)\mathsf{OPT}_k$

hold together with probability at least

$$1 - [\delta_2 + \delta_2 + (2\delta_2 + (2\delta_2)^2 + (2\delta_2)^4 + \dots)]$$
$$\geq 1 - (2\delta_2 + \frac{2\delta_2}{1-2\delta_2}) \geq 1 - \delta$$

The last one is due to $\delta_2 = \delta/6 \leq 1/6$.

From the above inequalities, we will have $T^* \leq m_{\mathcal{H}} \leq cT^*$. And the algorithm will stop in one of (at most) $\log_2 \mathsf{c}+1$ iterations, starting from $t_0$. Further, no matterwhat the iteration that the algorithm will stop at, the candidate seed set $\hat{S}_k$ satisfies $\mathbb{B}(\hat{S}_k) \geq (1-1/e-\epsilon)\mathsf{OPT}_k$. Since $T^* = O(\theta_B(\epsilon,\delta))$,
$\Pr[m_{\mathcal{H}} = O(\theta_B(\epsilon,\delta))$and $\mathbb{B}(\hat{S}_k) \geq (1-\frac{1}{e}-\epsilon)\mathsf{OPT}_k] \geq 1-\delta$.
That completes the proofs. □

### 4.2 Time Complexity

The overall time complexity of BCT comprises of two components: 1) for generating hyperedges and 2) for running Greedy algorithm for Max-Coverage. The result is stated in the following theorem and the proof is presented in our conference paper [22].

**Theorem 2.** *BCT has an expected running time for uniform cost CTVM problem under LT model of $O(\frac{\log(\binom{n}{k}/\delta)}{\epsilon_2^2}n)$.*

**Remark**. From Theorem 2, under the LT model, the time complexity does not depend on the number of edges in the original graph, hence, uniform-cost BCT has a sub-linear time complexity in dense graphs.

### 4.3 Sample Complexity and Comparison to IMM

Since the number of samples (hyperedges) decides the complexity of BCT, IMM [14] and any algorithm using sampling techniques, we compare number of hyperedges generated by BCT with the current state-of-the-art IMM. We can prove a tighter version of Lemma 5 as stated in the lemma below.

**Lemma 6.** *Let $\delta_2 \in (0,1), 0 < \epsilon < (1-1/e)$, BCT returns $\hat{S}_k$,*

$$deg_{\mathcal{H}}(\hat{S}_k) \leq 2\frac{(1+\epsilon_2)\cdot(2+2\epsilon_2/3)\cdot\log(6\binom{n}{k}/\delta_2)}{\epsilon_2^2}, \tag{18}$$

*due to doubling hyperedges every round and,*

$$\Pr[m_{\mathcal{H}} \geq \frac{2}{(1-\epsilon_2)(1-1/e-\epsilon)}T^*] \leq \delta_2 = \delta/6. \tag{19}$$

In comparison with IMM [14], BCT theoretically generates at least $3/2$ times fewer samples than IMM. IMM approaches the problem by trying to achieve an estimate $KPT^+$ of $OPT_k$ such that $KPT^+ \leq OPT_k$ and then deriving the sufficient number of samples by replacing $OPT_k$ in $T^*$ of Lemma 3 by $KPT^+$ and the constant $(2+2\epsilon_2/3)$ by $(2-2/e)$ to get $T_2^*$. Thus, Lemma 9 in [14] states the number of samples generated by IMM, $|\mathcal{R}|$, as follows,

$$\Pr\left[|\mathcal{R}| \leq 3\frac{(1+\epsilon')^2}{1-1/e}\max\{T_2^*, T_2'\}\right] \geq 1-\delta, \tag{20}$$

where $\epsilon' = \sqrt{2}\epsilon$ and

$$T_2' = \frac{(2+\frac{2}{3}\epsilon')(\log\binom{n}{k} + \log(1/\delta) + \log\log_2(n))\cdot n}{\epsilon'^2 OPT_k}.$$

Comparing Eqs. 20 and 19, we see that $\max\{T_2^*, T_2'\} \geq T^*$ and $3\frac{1+\sqrt{2}\epsilon}{1-1/e} > 2\frac{1}{(1-\epsilon_2)(1-1/e-\epsilon)}$ (assume that $\epsilon$ is small). Thus, the number of samples generated by BCT is always less than that of IMM and the ratio between the two is approximately $3/2$ (when $\epsilon \leq \sqrt{2}-1 > 0.4$ which is usually the case). In fact, our experiments show that BCT is up to 10x faster than IMM proving the practical efficiency.

### 4.4 Approximation Algorithm for Arbitrary Cost CTVM

We analyze the CTVM algorithm under the heterogeneous selecting costs. First observe that in this case, the candidate seed sets may have different sizes since the total cost of each set must be less that the given budget $B$. However, we can obtain an upper-bound $k_{max} = \max\{k : \exists S \subset$

$V, |S| = k, c(S) \leq B\}$ by iteratively selecting the smallest cost nodes until reaching the budget $B$. We then guarantee that all subsets of size up to $k_{max}$ are well approximated. The number of such seed sets is subsequently bounded above by $\sum_{k \leq k_{max}} \binom{n}{k} \leq n^{k_{max}}$. Thus, the computation of $\alpha$ and $\beta$ at the step of calculating $\epsilon_1$ are updated to $\epsilon_1'$,

$$\epsilon_2' = \frac{\epsilon\sqrt{(1-1/e)k_{max}\log(n \cdot 2/\delta)}}{(1-1/e)\sqrt{\log(2/\delta)} + \sqrt{(1-1/e)k_{max}\log(n \cdot 2/\delta)}}$$

Thus, $\Lambda_L$ is also updated to $\Lambda_L^2$ as follows,

$$\Lambda_L^{cost} = \frac{(1+\epsilon_2') \cdot (2 + 2\epsilon_2'/3) \cdot \log(6\binom{n}{k}/\delta)}{\epsilon_2'^2}. \quad (21)$$

In addition, the Weighted-Max-Coverage algorithm used in CTVM only guarantees $(1-1/\sqrt{e})$ approximate solutions, as shown in [23]. Putting these modifications together, we have the following Theorem 3. The proofs are similar to that of Theorem 1 and 2 and is omitted for clarity.

**Theorem 3.** *Given a budget $B$, $0 \leq \epsilon \leq 1$ and $0 \leq \delta \leq 1$, BCT for arbitrary cost CTVM problem returns a solution $\hat{S}$ that,*

$$\Pr[\mathbb{B}(\hat{S}) \geq (1 - 1/\sqrt{e} - \epsilon)OPT] \geq 1 - \delta, \quad (22)$$

*and runs in time $O(\frac{\log(\binom{n}{k}/\delta)}{\epsilon_2'^2} n)$.*

### 4.5 Extension to IC model

When applying BCT for IC model, the only change is in the BSA procedure to generate hyperedges following the IC model, as originally presented in [13]. Thus, our results for LT model translate directly over for IC model. Specifically, the following theorem states the solution guarantee and time complexity of BCT to the uniform cost version.

**Theorem 4.** *Given a budget $B$, $0 \leq \epsilon \leq 1$ and $0 \leq \delta \leq 1$, BCT for uniform cost CTVM problem returns $\hat{S}$ where*

$$\Pr[\mathbb{B}(\hat{S}) \geq (1 - 1/e - \epsilon)OPT] \geq 1 - \delta, \quad (23)$$

*and runs in time $O(\frac{\log(\binom{n}{k}/\delta)}{\epsilon_2^2}(m+n))$.*

Similar to Theorem 5, we obtain the performance guarantee for the arbitrary cost version under IC model in the following theorem.

**Theorem 5.** *Given a budget $B$, $0 \leq \epsilon \leq 1$ and $0 \leq \delta \leq 1$, for arbitrary cost CTVM problem, BCT returns a solution $\hat{S}$,*

$$\Pr[\mathbb{B}(\hat{S}) \geq (1 - 1/\sqrt{e} - \epsilon)OPT] \geq 1 - \delta, \quad (24)$$

*and runs in time $O(\frac{\log(\binom{n}{k}/\delta)}{\epsilon_2'^2}(m+n))$.*

## 5 Experiments

In this section, we experimentally evaluate and compare the performance of BCT to other influence maximization methods on three aspects: *the solution quality, the scalability, and the applicability* of BCT on various network datasets including our case study on a billion-scale dataset with both links and content.

### 5.1 Experimental Settings

All the experiments are carried on a Linux machine with a 2.2Ghz Xeon 8 core processor and 64GB of RAM.

#### *Algorithms compared*

We choose three groups of methods to test on:

(1) Designed for IM task, including the top four state-of-the-art algorithms, i.e., IMM [14], TIM/TIM+ [8], CELF++ [5] and SIMPATH [4].
(2) Designed for BIM task, namely, BIM algorithm [15].
(3) Our method BCT for the general CTVM problem.

In the first experiment, we will compare between these groups of methods on CTVM problem and the second experiment reports results on IM task. Our last set of experiments are on Twitter - a billion-scale network where we first test the scalability of BCT against IMM and TIM+ (the current most scalable methods for solving IM problem) on IM task. Next, we acquire a Twitter's tweet dataset and extract two groups of users who tweet/retweet the same topic and run our BCT algorithm to find the users who attract the most interested people in the same topics.

**TABLE 3:** Datasets' Statistical Summary

| Dataset | #Nodes | #Edges | Type | Avg. degree |
|---|---|---|---|---|
| NetHEPT [3] | 15K | 59K | undirected | 4.1 |
| NetPHY [3] | 37K | 181K | undirected | 13.4 |
| Enron [32] | 37K | 184K | undirected | 5.0 |
| Epinions [3] | 132K | 841K | directed | 13.4 |
| DBLP [3] | 655K | 2M | undirected | 6.1 |
| Twitter [33] | 41.7M | 1.5G | directed | 70.5 |

#### *Datasets*

For a comprehensive experimental purpose, we select a diverse set of 6 datasets with sizes from thousands to millions in various disciplines: NetHEPT, NetPHY, DBLP are citation networks, Email-Enron is communication network, Twitter and Epinions are online social networks. The description summary of those datasets is provided in Table 3.

#### *Parameter Settings*

*Computing the edge weights*. Following the conventional computation as in [4], [8], [15], [34], the weight of the edge $(u, v)$ is calculated as follows,

$$w(u, v) = 1/d_{in}(v) \quad (25)$$

where $d_{in}(v)$ denotes the in-degree of node $v$.

*Computing the node costs*. Intuitively, the more famous one is, the more difficult it is to convince that person. Hence, we assign the cost of a node proportional to the out-degree:

$$c(u) = nd^{out}(u)/\sum_{v \in V} d^{out}(v) \quad (26)$$

where $d^{out}(v)$ is the out-degree of node $v$.

*Computing the node benefits*. In the first experiment, we choose a random $p = 20\%$ of all the nodes to be the target set and assign benefit 1 to all of them while in case studies, the benefit is learned from a separate dataset.

In all the experiments, we keep $\epsilon = 0.1$ and $\delta = 1/n$ as a general setting or directly mentioned otherwise. For the other parameters, we take the recommended values in the corresponding papers if available.

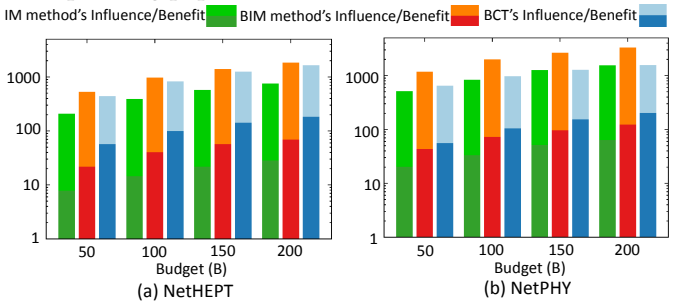
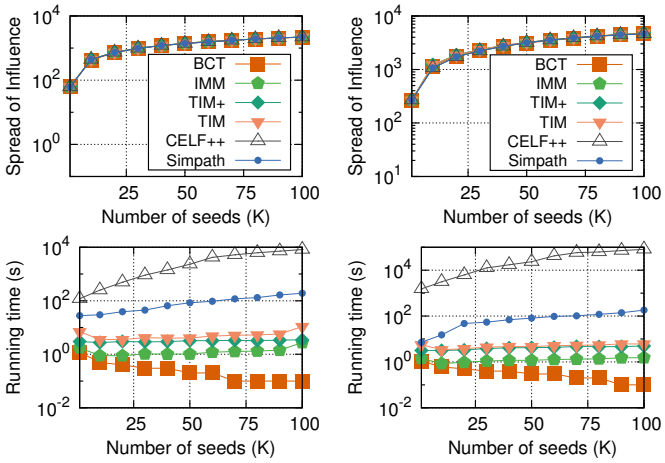
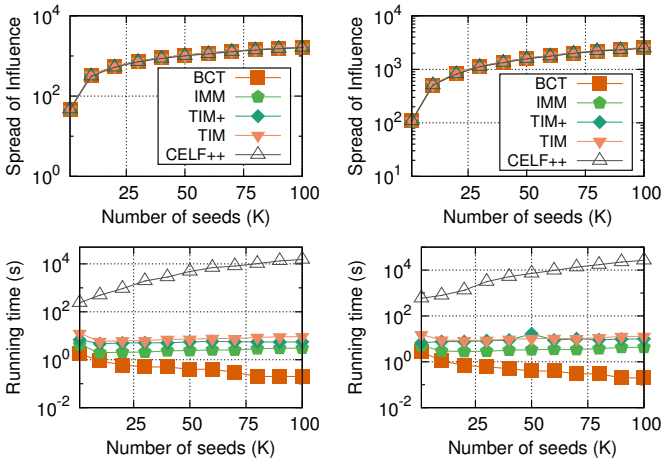



**Fig. 1:** Comparisons on CTVM problem. The whole column indicates influence of the selected seeds while the darker colored portion reflects the benefit gained from that set.




**(a)** NetHEPT **(b)** NetPHY

**Fig. 2:** Comparison on IM problem under the LT model.




**(a)** NetHEPT **(b)** NetPHY

**Fig. 3:** Comparison on IM problem under the IC model.

### 5.2 Experimental results

We carry three experiments on both CTVM and IM tasks to compare the performance of BCT with other state-of-the-art methods. In the first experiments, we compare three groups of algorithms, namely, IM based methods, BIM and BCT on CTVM problem. We choose four algorithms in the category of IM methods: CELF++, SIMPATH, TIM/TIM+ and IMM, which are well known algorithms for IM. The results are presented in Fig. 1. We conduct the second and third experiments on the classical IM task with different datasets and various $k$ values. The results are shown in Table 4 and Fig. 2 for LT model and Fig. 3 for IC model.

**TABLE 4:** Comparison between different methods on IM problem and various datasets (with $\epsilon = 0.1, k = 50, \delta = \frac{1}{n}$).

| Method | Spread of Influence | | | Running Time (s) | | |
|---|---|---|---|---|---|---|
| | *Epin.* | *Enron* | *DBLP* | *Epin.* | *Enron* | *DBLP* |
| BCT | 16280 | 16726 | 108400 | **0.19** | **0.14** | **0.58** |
| IMM | 16290 | 16716 | 108430 | 2 | 1.5 | 3.5 |
| TIM+ | 16293 | 16732 | 108343 | 6 | 3 | 12 |
| TIM | 16306 | 16749 | 107807 | 8 | 4 | 17 |
| Simpath | 16291 | 16729 | 103331 | 23 | 18 | 136 |

#### 5.2.1 *Comparison of solution quality on CTVM*

Fig. 1 shows the results of the three groups of methods (CELF++ with 10000 sampling times represents the first group) for solving CTVM problem on NetHEPT and NetPHY networks. We see that BCT outperforms the other methods by a large margin on CTVM problem. With the same amount of budget, CTVM returns a solution which is up to order of magnitudes better than that of BIM and IM based methods in terms of benefit. Because IM algorithms only desire to maximize the influence and thus usually aim at the most influential nodes, unfortunately, those nodes are very expensive or have high cost. As a consequence, when nodes have heterogeneous cost, IM methods suffer severely in terms of both influence and benefit. On the other hand, BIM optimizes cost and influence while ignoring benefit of influencing nodes that causes BIM to select cheaper nodes with high influence. Hence, the seed sets returned by BIM have the highest influence among all but relatively low benefit. Even though BCT returns seed set with lower value of influence than BIM, the majority of the influenced nodes are our target and thus achieves the most benefit.

#### 5.2.2 *Comparison of solution quality on IM*

In the previous experiment, one can argue that CTVM performs better because it focuses on optimizing the benefit and the others do not. This experiment compares BCT to the other algorithms with IM problem where the node costs are all 1 and so as the node benefits on various datasets. Fig. 2 and Fig. 3 display the spread of influence and running time on NetHEPT and NetPHY under the LT and IC models respectively. Table 4 shows the cross-dataset view of the results when we fix a setting and run on multiple data.

Fig. 2, Fig. 3 and Table 4 reveal that all the tested algorithms including BCT and the top methods on IM problem achieve the same level of performance in terms of spread of influence. Specifically, they all expose the phenomenon that the first few seed nodes ($\leq 25$) can influence a large portion of the networks and after that point, the gains of adding more seeds are marginal. The phenomenon is explained by the submodularity property of the influence function.

#### 5.2.3 *Comparison of running time*

The experimental results in Fig. 2, Fig. 3 and Table 4 also confirm our theoretical establishment in Section 4 that BCT for uniform cost CTVM requires much less number of hyperedges needed by IMM, TIM and TIM+. As such, the running time of BCT in all the experiments are significantly lower than the other methods. In average, BCT is up to 10 and 50 times faster IMM and TIM/TIM+, respectively. Since both Simpath and CELF++ require intensive graph simulation, these methods have very poor performance compared to BCT, TIM/TIM+ and IMM which apply advanced techniques to approximate the influence landscape. That is illustrated by the distinct separation of two groups.

#### 5.2.4 *Robustness Testing*

In this experiment, we test the robustness of our algorithm against noise possibly incurred in computing the edge weights in the diffusion model. We take NetHEPT with the previously calculated edge weights as our 'ground-truth' network and then add various noises, e.g., in different levels and noise models to it. In particular, we consider Gaussian and Uniform noise models where the added noise follows either a Gaussian or Uniform distribution respectively. To account for noise levels, we select 4 different values $0.2, 0.4, 0.6, 0.8$, which correspond to 20% to 80% noise since the edge weight is between 0 and 1, and assign them to be the variance of the distribution (larger variance

signifies noisier data) while the mean values are set at 0. Thus, each pair of noise level and model, we have a specific distribution of noise and use that for generating noise. For each such pair, we generate 30 noisy networks and run BCT to find 50 seed nodes and then take the average over 30 runs. We then use the original HetHEPT without noise to recompute the influence and quantify the effect of noise.

**TABLE 5:** Robustness results (% to true value).

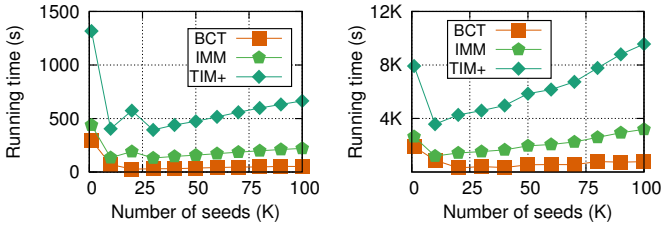
| | | Uniform Noise | | | | Gaussian Noise | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\epsilon$ | True | 0.2 | 0.4 | 0.6 | 0.8 | 0.2 | 0.4 | 0.6 | 0.8 |
| 0.1 | 1588 | 99.7 | 99.2 | 98.6 | 97.9 | 98.9 | 98.0 | 96.7 | 95.7 |
| 0.2 | 1504 | 99.9 | 99.7 | 99.2 | 98.8 | 100.1 | 99.7 | 98.3 | 95.8 |
| 0.4 | 1312 | 99.9 | 99.5 | 98.4 | 98.8 | 100.9 | 99.4 | 98.7 | 98.2 |

The experimental results under the IC model are depicted in Table 5 where the 'true value' refers to the results run on the original network. Interestingly, BCT performs very well under the noises introduced to the network. For example, with 80% noise, the quality of BCT only degrades by less than 2% with Uniform noise and 5% with Gaussian case in average. In some rare cases on network with 20% Gaussian noise, we see the qualities of over 100% compared to true values. This happens when $\epsilon$ is large implying the provided solution guarantee $1 - 1/e - \epsilon$ is small. Thus, the seed set found on small noisy network may get better than that on the original. Moreover, different from the Uniform case, Gaussian noise is highly concentrated at 0.

### 5.3 Twitter: A billion-scale social network

In this subsection, we design two case studies on Twitter network: one is to compare the scalability of BCT with IMM and TIM+ - the fastest existing methods and the another is using BCT to find a set of users who have highest benefit with respect to a particular topic in Twitter.

#### 5.3.1 *Compare BCT against IMM and TIM+*




**(a)** LT model. **(b)** IC model.

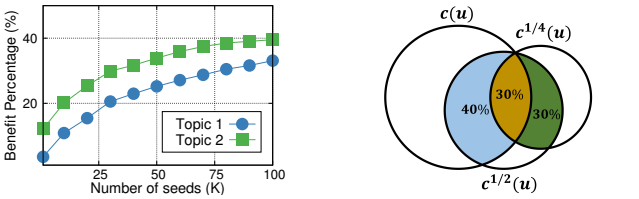
**Fig. 4:** BCT, IMM and TIM+ on Twitter

Figure 4 shows the results of running BCT and TIM+ on Twitter network dataset using both LT and IC models with $k$ ranging from 1 to 100. Twitter has 1.5 billion edges and all the other methods, except BCT, IMM and TIM+, fail to process it within a day in our experiments. The results, here, are consistent with the other results in the previous experiments. Regardless of the values of $k$, in LT model, BCT is always several times faster than IMM or TIM+ and in IC model, this ratio is in several orders of magnitude since influence in IC model is much larger and, thus, harder for IMM or TIM+ to have a close estimate of the optimality which decides the complexity of these algorithms.

We also measure the memory consumed by these two algorithms and observe that, in average, BCT requires around 20GB but IMM and TIM+ always need more than 30GB. This is a reasonable since in addition the memory for the original graph, BCT needs much less number of hyperedges than that generated by IMM or TIM+.

**TABLE 6:** Topics, related keywords

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




**(a)** Benefit on Twitter. **(b)** Different Cost Schemes.

**Fig. 5:** Case Study on real-world Twitter

#### 5.3.2 *A Case Study on Twitter network.*

We study the twitter network using BCT by extracting some trending topics from the retrieved tweet dataset and find who are most influential in those topics based on Twitter network. First we choose two most popular topics with related keywords (Table 6) as reported in [33]. Based on the list of keywords, we use a Twitter's tweet dataset to extract a list of users who mentioned the keywords in their posts and the number of those tweets/retweets. The number of tweets/retweets reveals the interest of the user on the topic, thus, we consider this as the benefit of that node. Lastly, we run BCT on Twitter with the extracted parameters.

Fig. 5a shows the benefit percentage, which is computed as the percentage of benefit gained by the selected seed set over the total benefit. We see that apparently the very first chosen nodes have high benefit and it continues increasing later but with much lower rate. Looking into the first 5 Twitters chosen by the algorithm, they are users with only few thousands of followers (unlike Katy Perry or President Obama who got more than 50 millions followers) but highly active posters in the corresponding topic. For example, on the first political topic, the first selected users is a Canadian poster, who is originally from Iran and has about 4000 followers and but generates more than 210K posts on the movements of governments in the US and Iran.

On Twitter we test different schemes of assigning node costs, i.e., proportional to a concave function. We employ square and fourth roots, denoted by $c^{1/2}(u)$ and $c^{1/4}(u)$ respectively, w.r.t. out-degree and run BCT on each case. The results are presented in Figure 5b. We see that BCT is relatively robust with different concave cost functions, e.g., 70% of nodes returned in case of $c^{1/2}(u)$ overlaps with that of $c(u)$-linear cost, and 60% overlap for the pair $c^{1/4}(u)$ to $c^{1/2}(u)$. Another interesting result is that the number of selected seeds gets smaller when the cost function gets farther from linear, i.e., $c(u) \to c^{1/2}(u)$ and $c^{1/2}(u) \to c^{1/4}(u)$.

## 6 Conclusion

In this paper, we propose the CTVM problem that generalizes several viral marketing problems including the classical IM. We propose BCT an efficient approximation algorithm to solve CTVM in billion-scale networks and show that it is both theoretically sound and practical for large networks. The algorithm can be employed to discover more practical solutions for viral marketing problems, as illustrated through the discovering of influential users w.r.t. trending topics in Twitter media site.

## ACKNOWLEDGMENT

We sincerely thank the reviewers for the insightful comments. The work of Dr. My T. Thai was supported in part by the NSF grants CNS-1443905 and CCF-1422116.

## REFERENCES

[1] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *KDD*. ACM New York, NY, USA, 2003, pp. 137–146.
[2] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *KDD*. New York, NY, USA: ACM, 2007, pp. 420–429.
[3] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *KDD*. New York, NY, USA: ACM, 2010, pp. 1029–1038.
[4] A. Goyal, W. Lu, and L. V. Lakshmanan, "Simpath: An efficient algorithm for influence maximization under the linear threshold model," in *ICDM*. IEEE, 2011, pp. 211–220.
[5] ——, "Celf++: optimizing the greedy algorithm for influence maximization in social networks," in *WWW*. ACM, 2011, pp. 47–48.
[6] E. Cohen, D. Delling, T. Pajor, and R. F. Werneck, "Sketch-based influence maximization and computation: Scaling up with guarantees," in *CIKM*. ACM, 2014, pp. 629–638.
[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 *AAAI*, 2014.
[8] Y. Tang, X. Xiao, and Y. Shi, "Influence maximization: Near-optimal time complexity meets practical efficiency," in *SIGMOD*. ACM, 2014, pp. 75–86.
[9] N. P. Nguyen, G. Yan, and M. T. Thai, "Analysis of misinformation containment in online social networks," *Comput. Netw.*, vol. 57, no. 10, pp. 2133–2146, Jul. 2013.
[10] N. Barbieri, F. Bonchi, and G. Manco, "Topic-aware social influence propagation models," *KAIS*, vol. 37, no. 3, pp. 555–584, 2013.
[11] S. Chen, J. Fan, G. Li, J. Feng, K.-l. Tan, and J. Tang, "Online topic-aware influence maximization," *VLDB*, pp. 666–677, 2015.
[12] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *KDD*. ACM, 2005, p. 187.
[13] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier, "Maximizing social influence in nearly optimal time," in *SODA*. SIAM, 2014, pp. 946–957.
[14] Y. Tang, Y. Shi, and X. Xiao, "Influence maximization in near-linear time: A martingale approach," in *SIGMOD*. ACM, 2015, pp. 1539–1554.
[15] H. Nguyen and R. Zheng, "On budgeted influence maximization in social networks," *JSAC*, vol. 31, no. 6, pp. 1084–1094, 2013.
[16] Y. Li, D. Zhang, and K.-L. Tan, "Real-time targeted influence maximization for online advertisements," *VLDB*, pp. 1070–1081, 2015.
[17] U. Feige, "A threshold of ln n for approximating set cover," *Journal of ACM*, vol. 45, no. 4, pp. 634–652, 1998.
[18] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*, J. Stoer, Ed. Springer, 1978, pp. 234–243.
[19] N. Chen, "On the approximability of influence in social networks," *SIDMA*, pp. 1400–1415, 2009.
[20] A. Goyal, F. Bonchi, and L. Lakshmanan, "Learning influence probabilities in social networks," in *WSDM*. ACM, 2010, pp. 241–250.
[21] K. Kutzkov, A. Bifet, F. Bonchi, and A. Gionis, "Strip: stream learning of influence probabilities," in *SIGKDD*. ACM, 2013, pp. 275–283.
[22] H. T. Nguyen, T. N. Dinh, and M. T. Thai, "Cost-aware targeted viral marketing in billion-scale networks," in *INFOCOM*. IEEE, 2016, pp. 1–9.
[23] S. Khuller, A. Moss, and J. S. Naor, "The budgeted maximum coverage problem," *Inform. Process. Lett.*, pp. 39–45, 1999.
[24] P. Dagum, R. Karp, M. Luby, and S. Ross, "An optimal algorithm for monte carlo estimation," *SICOMP*, pp. 1484–1496, 2000.
[25] C. Aslay, N. Barbieri, F. Bonchi, and R. A. Baeza-Yates, "Online topic-aware influence maximization queries." in *EDBT*, 2014, pp. 295–306.
[26] M. Cha, A. Mislove, and K. P. Gummadi, "A measurement-driven analysis of information propagation in the flickr social network," in *WWW*. New York, NY, USA: ACM, 2009, pp. 721–730.
[27] J. Tang, J. Sun, C. Wang, and Z. Yang, "Social influence analysis in large-scale networks," in *SIGKDD*. ACM, 2009, pp. 807–816.
[28] A. Goyal, F. Bonchi, and L. V. Lakshmanan, "Learning influence probabilities in social networks," in *WSDM*. ACM, 2010, pp. 241–250.
[29] L. Seeman and Y. Singer, "Adaptive seeding in social networks," in *FOCS*. IEEE, 2013, pp. 459–468.
[30] A. J. Walker, "An efficient method for generating discrete random variables with general distributions," *TOMS*, pp. 253–256, 1977.
[31] S. Khuller, A. Moss, and J. Naor, "The budgeted maximum coverage problem," *Inform. Process. Lett.*, pp. 39–45, 1999.
[32] B. Klimt and Y. Yang, "Introducing the Enron corpus," in *First Conference on Email and Anti-Spam (CEAS)*, 2004.
[33] H. Kwak, C. Lee, H. Park, and S. Moon, "What is twitter, a social network or a news media?" in *WWW*. ACM, 2010, pp. 591–600.
[34] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *KDD*. New York, NY, USA: ACM, 2009, pp. 199–208.

## APPENDIX A
## MARTINGALE VIEW ON BENEFIT ESTIMATION

To recognize the connection between the expected benefit and martingales, we give a general definition as follows,

**Definition 3** (Martingale)**.** *A sequence of random variables $Y_1, Y_2, \ldots$ is a martingale if and only if $\mathbb{E}[Y_i] \leq +\infty$ and $\mathbb{E}[Y_i|Y_1, Y_2, \ldots, Y_{i-1}] = Y_{i-1}$.*

For a random hyperedge $\mathcal{E}_j$, we define random variable,

$$X_j = \begin{cases} 1 & \text{if } S \cap \mathcal{E}_j \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{27}$$

Thus, we have a sequence of random variables $X_1, X_2, \ldots$ corresponding to the series of random hyperedges. Then, we define the second sequence of random variables based on $X_1, X_2, \ldots$ as follows,

$$Z_j = \sum_{i=1}^{j} (X_i - \mathbb{B}(S)/\Gamma) \tag{28}$$

The sequence $Z_1, Z_2, \ldots$ has the following properties [14]:

(1) $\mathbb{E}[Z_j] = 0, \forall j \geq 1$ (since $\mathbb{E}[Z_i] = \mathbb{B}(S)/\Gamma, \forall i \geq 1$)
(2) $\mathbb{E}[Z_j|Z_1, Z_2, \ldots, Z_{j-1}] = Z_{j-1}$

Thus, the two conditions (1) and (2) hold and make the sequence $Z_1, Z_2, \ldots$ a martingale. Hence the following inequalities for martingales follow from [14].

**Lemma 7.** *For any fixed $T > 0$ and $\epsilon > 0$, we have*

$$\Pr[\hat{\mu} \geq (1+\epsilon)\mu] \leq e^{\frac{-T\mu\epsilon^2}{2+\frac{2}{3}\epsilon}},$$

*and*

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

*where $\hat{\mu} = \frac{\sum_{i=1}^{T} X_i}{T}$ is an estimate of $\mu = \mathbb{B}(S)/\Gamma$.*

## APPENDIX B
## PROOFS OF LEMMAS AND THEOREMS

### B.1 Proof of Lemma 2

We start with the definition of $\mathbb{B}(S)$ in Eq. 2 and prove the equivalent formula $\mathbb{B}(S) = \Gamma \Pr_{g \sqsubseteq G, u \in V}[\mathcal{E}_j \cap S \neq \emptyset]$.

$$\begin{aligned}
\mathbb{B}(S) &= \sum_{u \in V} \Pr_{g \sqsubseteq G}[u \in R(g, S)] b(u) \\
&= \sum_{u \in V} \Pr_{g \sqsubseteq G}[\exists v \in S \text{ such that } v \in \mathcal{E}_j(u)] b(u) \\
&= \Gamma \sum_{u \in V} \Pr_{g \sqsubseteq G}[\exists v \in S \text{ such that } v \in \mathcal{E}_j(u)] \frac{b(u)}{\Gamma} \\
&= \Gamma \Pr_{g \sqsubseteq G, u \in V}[\exists v \in S \text{ such that } v \in \mathcal{E}_j] \\
&= \Gamma \Pr_{g \sqsubseteq G, u \in V}[S \cap \mathcal{E}_j \neq \emptyset].
\end{aligned} \tag{29}$$

The transition from the third to forth equality follows from the distribution of choosing node $u$ as a starting node of BSA. Since we select $u$ with probability $P(u) = b(u)/\Gamma$, the forth equality contains the expected probability taken over the benefit distribution. That completes our proof.

### B.2 Proof of Lemma 3

Consider the first $T^* = \frac{(2+2\epsilon_2/3)\cdot\Gamma\cdot\ln\left(\binom{n}{k}/\delta_2\right)}{\mathsf{OPT}_k\cdot\epsilon_2^2}$ hyperedges, for any set $S_k$ of $k$ nodes, by Chernoff's bound (Lemma 7),

$$\begin{aligned}
&\Pr[\hat{\mathbb{B}}_{T^*}(S_k) \geq \mathbb{B}(S_k) + \epsilon_2\mathsf{OPT}_k] \\
&= \Pr[\hat{\mathbb{B}}_{T^*}(S_k) \geq (1+\epsilon_2\frac{\mathsf{OPT}_k}{\mathbb{B}(S_k)})\mathbb{B}(S_k)] \\
&\leq \exp\left(-\frac{T^*\mathbb{B}(S_k)(\epsilon_2\frac{\mathsf{OPT}_k}{\mathbb{B}(S_k)})^2}{(2+\frac{2}{3}\epsilon_2\frac{\mathsf{OPT}_k}{\mathbb{B}(S_k)})\Gamma}\right) \\
&= \exp\left(-\frac{(2+2\epsilon_2/3)\Gamma\ln(\binom{n}{k}/\delta_2)\mathbb{B}(S_k)\epsilon_2^2\mathsf{OPT}_k^2}{\mathsf{OPT}_k^2\epsilon_2^2\mathbb{B}^2(S_k)(2+\frac{2}{3}\frac{\mathsf{OPT}_k}{\mathbb{B}(S_k)})\Gamma}\right) \\
&= \exp\left(-\frac{(2+2\epsilon_2/3)\ln(\binom{n}{k}/\delta_2)}{2\frac{\mathbb{B}(S_k)}{\mathsf{OPT}_k}+2\epsilon_2/3}\right) \leq \delta_2/\tbinom{n}{k}. \qquad (30)
\end{aligned}$$

Moreover,

$$\begin{aligned}
&\Pr[\hat{\mathbb{B}}_{T^*}(S_k) \geq \mathbb{B}(S_k) + \epsilon_2\mathsf{OPT}_k] \\
&\geq \Pr[\hat{\mathbb{B}}_{T^*}(S_k) \geq \mathsf{OPT}_k + \epsilon_2\mathsf{OPT}_k] \\
&= \Pr[deg_{T^*}(S_k)\Gamma/T^* \geq (1+\epsilon_2)\mathsf{OPT}_k] \\
&= \Pr[deg_{T^*}(S_k) \geq (1+\epsilon_2)\mathsf{OPT}_k T^*/\Gamma] \\
&= \Pr[deg_{T^*}(S_k) \geq \Lambda_L] \qquad (31)
\end{aligned}$$

Combine Eqs. 30 and 31, we obtain,

$$\Pr[deg_{T^*}(S_k) \geq \Lambda_L] \leq \delta_2/\tbinom{n}{k} \qquad (32)$$

Apply union bound over all seed sets $S_k$ of size $k$, we have,

$$\Pr[\exists S_k, deg_{T^*}(S_k) \geq \Lambda_L] \leq \delta_2 \qquad (33)$$

which means that with $T^*$ hyperedges, the probability of having a seed set $S_k$ of $k$ nodes with $deg_{T^*}(S_k) \geq \Lambda_L$ is less than $\delta_2$. INote that BCT stops only when the returned seed set $\hat{S}_k$ has $deg_{\mathcal{H}} \geq \Lambda_L$ that implies having a seed set with the coverage at least $\Lambda_L$. Thus, the number of hyperedges generated by BCT is at least $T^*$ with probability of $1-\delta_2$.

### B.3 Proof of Lemma 5

Since $\mathsf{c} > 1$, with $\mathsf{c}T^*$ hyperedges, we have,

$$\begin{aligned}
&\Pr[\hat{\mathbb{B}}_{\mathsf{c}T^*}(S_k) \leq \mathbb{B}(S_k) - \frac{\epsilon_2}{2}\mathsf{OPT}_k] \\
&= \Pr[\hat{\mathbb{B}}_{\mathsf{c}T^*}(S_k) \leq (1-\frac{\epsilon_2}{2}\frac{\mathsf{OPT}_k}{\mathbb{B}(S_k)})\mathbb{B}(S_k)] \\
&\leq \exp\left(-\frac{\mathsf{c}T^*\mathbb{B}(S_k)(\frac{\epsilon_2}{2}\frac{\mathsf{OPT}_k}{\mathbb{B}(S_k)})^2}{2\Gamma}\right) \\
&= \exp\left(-\frac{\mathsf{c}(2+2\epsilon_2/3)\Gamma\cdot\ln\left(\binom{n}{k}/\delta_2\right)\mathbb{B}(S_k)\epsilon_2^2\mathsf{OPT}_k^2}{2\Gamma\mathbb{B}^2(S_k)\mathsf{OPT}_k\epsilon_2^2 4}\right) \\
&= \exp\left(-\frac{\mathsf{c}(2+2\epsilon_2/3)\ln(\binom{n}{k}/\delta_2)\mathsf{OPT}_k}{2\mathbb{B}(S_k)4}\right) \leq \left(\delta_2/\tbinom{n}{k}\right)^{\mathsf{c}/4}
\end{aligned}$$

Thus, since there are at most $\binom{n}{k}$ such sets of size $k$,

$$\Pr[\exists S_k, \hat{\mathbb{B}}_{\mathsf{c}T^*}(S_k) \leq \mathbb{B}(S_k) - \frac{\epsilon_2}{2}\mathsf{OPT}_k] \leq \delta_2^{\mathsf{c}/4} \qquad (34)$$

Moreover, since $\mathsf{c} \geq 8$, thus, $\mathsf{c}T^* > 8T^*$, with $\mathsf{c}T^*$ hyperedges, applying Corollary 1 with parameter settings $\epsilon = \epsilon/2$ and $\delta = (2\delta_2)^{\mathsf{c}/4}$,

$$\Pr[\hat{\mathbb{B}}_{\mathsf{c}T^*}(\hat{S}_k) \leq (1-1/e-\epsilon/2)\mathsf{OPT}_k] \leq (2\delta_2)^{\mathsf{c}/4} \qquad (35)$$

From Eqs. 34 and 35, consider two events:

(e1) $\forall S_k, \hat{\mathbb{B}}_{\mathsf{c}T^*}(S_k) \geq \mathbb{B}(S_k) - \frac{\epsilon_2}{2}\mathsf{OPT}_k$

(e2) $\hat{\mathbb{B}}_{\mathsf{c}T^*}(\hat{S}_k) \geq (1-1/e-\epsilon/2)\mathsf{OPT}_k$

which together happen with probability of at least $1-\delta_2^{\mathsf{c}/4} - (2\delta_2)^{\mathsf{c}/4} \geq 1 - \frac{1}{6}\delta_2 - \frac{4}{6}\delta_2 = 1-\delta_2$ since $\mathsf{c} > 1$ and $\delta_2 = \delta/6 \leq 1/6$. In that case, we further derive,

$$\begin{aligned}
&\hat{\mathbb{B}}_{\mathsf{c}T^*}(\hat{S}_k) \geq \mathbb{B}(\hat{S}_k) - \frac{\epsilon_2}{2}\mathsf{OPT}_k \\
\Leftrightarrow\ &\mathbb{B}_{\mathsf{c}T^*}(\hat{S}_k) \geq (1-1/e-\frac{\epsilon}{2})\mathsf{OPT}_k - \frac{\epsilon_2}{2}\mathsf{OPT}_k \\
\Leftrightarrow\ &deg_{\mathsf{c}T^*}(\hat{S}_k)\frac{\Gamma}{\mathsf{c}T^*} \geq (1-1/e-\frac{\epsilon}{2}-\frac{\epsilon_2}{2})\mathsf{OPT}_k \\
\Leftrightarrow\ &deg_{\mathsf{c}T^*}(\hat{S}_k) \geq \frac{(1-1/e-\frac{\epsilon}{2}-\frac{\epsilon_2}{2})\mathsf{OPT}_k\mathsf{c}T^*}{\Gamma} \\
\Leftrightarrow\ &deg_{\mathsf{c}T^*}(\hat{S}_k) \geq \frac{(1-1/e-\frac{\epsilon}{2}-\frac{\epsilon_2}{2})}{(1+\epsilon_2)}\mathsf{c}\Lambda_L \qquad (36)
\end{aligned}$$

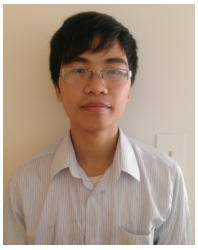
Now, since we set $\mathsf{c} = 4\left\lceil\frac{1+\epsilon_2}{1-1/e-\epsilon-\epsilon_2}\right\rceil > 4$, then,

$$deg_{\mathsf{c}T^*}(\hat{S}_k) > 4\Lambda_L \qquad (37)$$

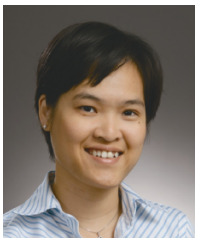
with probability $1-\delta_2$. Note that $\mathsf{c}$ exists due to the condition on $\epsilon$ that $\epsilon < (1-1/e)$ and $\epsilon_2 < \epsilon$.

In other words, with a probability of at least $1-\delta_2$, with $\mathsf{c}T^*$ hyperedges where $\mathsf{c} = 4\left\lceil\frac{1+\epsilon_2}{(1-1/e-\frac{\epsilon}{2}-\frac{\epsilon_2}{2})}\right\rceil$, the stopping condition in BCT will be satisfied. Thus, BCT generates at most $\mathsf{c}T^*$ hyperedges with probability at least $1-\delta_2$, or

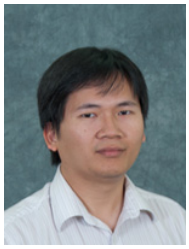
$$\Pr[m_{\mathcal{H}} \leq \mathsf{c}T^*] \geq 1-\delta_2. \qquad (38)$$



**Hung T. Nguyen** received the BS degree in Information Technology from Vietnam National University, Hanoi, Vietnam in 2014. He is currently a PhD candidate at the Department of Computer Science, Virginia Commonwealth University, under the supervision of Dr. Thang N. Dinh. His research focuses on designing efficient approximation algorithms to mine *billion-scale networks*, especially, when the network is evolving and/or uncertain.



**My T. Thai** (M'06) received the Ph.D. degree in Computer Science from the University of Minnesota in 2005. She is Professor at the Computer and Information Science and Engineering Department, University of Florida. Her current research interests include algorithms and optimization on network science and engineering, with a focus on security. She has engaged in many professional activities, such as being the PC chair of EEE IWCMC 12, IEEE ISSPIT 12, and COCOON 2010. She is a founding EiC of Computational Social Networks journal, an Associate Editor of JOCO, IEEE Transactions on Parallel and Distributed Systems, and a series editor of Springer Briefs in Optimization. She has received many research awards including a UF Provosts Excellence Award for Assistant Professors, a DoD YIP, and an NSF CAREER Award.



**Thang N. Dinh** (S'11-M'14) received the Ph.D. degree in computer engineering from the University of Florida in 2013. He is an Assistant Professor at the Department of Computer Science, Virginia Commonwealth University. His research focuses on security and optimization challenges in complex systems, especially social networks, wireless and cyber-physical systems. He serves as PC chair of COCOON'16 and CSoNet'14 and on TPC of several conferences including IEEE INFOCOM, ICC, GLOBECOM, and SOCIALCOM. He is also an associate editor of Computational Social Networks journal and a guest-editor for Journal of Combinatorial Optimization.