Cost-Aware Targeted Viral Marketing with Time Constraints in Social Networks

Ke Xu and Kai Han(✉)

School of Computer Science and Technology, Suzhou Institute for Advanced Study,
University of Science and Technology of China, Hefei, People's Republic of China
ustcxk@mail.ustc.edu.cn, hankai@ustc.edu.cn

Abstract. Online social networks have been one of the most effective platforms for marketing which is called *viral marketing*. The main challenge of viral marketing is to seek a set of k users that can maximize the expected influence, which is known as Influence Maximization (IM) problem. In this paper, we incorporate *heterogeneous costs and benefits* of users and *time constraints*, including *time delay* and *time deadline* of influence diffusion, in IM problem and propose *Cost-aware Targeted Viral Marketing with Time constraints* (CTVMT) problem to find the most cost-effective seed users who can influence the most relevant users within a time deadline. We study the problem under IC-M and LT-M diffusion model which extends IC and LT model with time constraints. Since CTVMT is NP-hard under two models, we design a **BCT-M** algorithm using two new benefit sampling algorithms designed for IC-M and LT-M respectively to get a solution with an approximation ratio. To the best of our knowledge, this is the first algorithm that can provide approximation guarantee for our problem. Our empirical study over several real-world networks demonstrates the performances of our proposed solutions.

Keywords: Social network · Influence maximization · Time constraints

1 Introduction

Recently, online social networks rapidly increasing to involve billions of active users, which makes it play an important role in daily life. For instance, online social networks such as Facebook, Twitter, have become critical platforms for marketing and advertising. Information and invention can propagate wildly over the network with the help of word-of-mouth effect and we call this marketing *Viral Marketing*. There is an extensively studied problem named Influence Maximization (IM) in viral marketing. It aims to find a seed set of k influential users in a social network so that the number of people influenced by seed set, named *influence spread*, can be maximum.

In their seminal paper, Kempe *et al.* [1] formulated IM as a combinatorial optimization problem and proposed two classical diffusion models, namely, *Independent Cascade* (IC) and *Linear Threshold* (LT) model. However, Chen *et al.*

H. Gao et al. (Eds.): CollaborateCom 2018, LNICST 268, pp. 75–91, 2019.
https://doi.org/10.1007/978-3-030-12981-1_5

[2] considered that IC and LT do not fully incorporate important *temporal* factors that have been well observed in the dynamic of influence diffusion. First, the propagation of influence from one person to another may incur a certain amount of *time delay* and second, the spread of influence may be *time-critical* in practice, i.e., beyond a certain *time deadline*, the spread of influence is meaningless. We conclude the two temporal factors as *time constraints*. They proposed *Independent Cascade with Meeting events* (IC-M) and *Linear Threshold with Meeting events* (LT-M) model to capture the delay of propagation. They studied the IM problem with time constraints under IC-M and LT-M models and proposed some heuristic algorithms.

Unfortunately, except for the time constraints, IM problem still ignores the different cost when select a user into seed set and the different benefit from an influenced user. Nguyen *et al.* [3] extended the IM problem to *Cost-aware Targeted Viral Marketing* (CTVM) problem with the addition of arbitrary costs and benefits above but did not consider the time constraints.

In this paper, we propose *Cost-aware Targeted Viral Marketing with Time constraints* (CTVMT) problem connecting CTVM problem and time constraints together. In our problem settings, every user has his own cost when he is selected into seed set and also has his own benefit when he gets influenced. We aim to find a set of influential users within a predefined budget in a social network such that they can influence the largest number of targeted users when reaching the time deadline. Formally, given a social network G, a budget B and a time deadline T, let each node v_i in G refer to a user and e_{ij} which denotes the edge from v_i to v_j refer to the relationship between users. $c(v_i)$ denotes the cost of v_i and $b(v_i)$ denotes the benefit of v_i. $m(e_{ij}) \in (0, 1]$ denotes the *meeting probability* that v_i can meet v_j at any time round t. The CTVMT problem is to identify a seed set $S = \{v_1, v_2, \ldots, v_j\}$ in G, such that (1) the total cost of seed set is within the budget B, i.e., $c(S) = \sum_{v_i \in S} c(v_i) \leq B$, and (2) the users in seed set incur influence spread in G and maximize total benefits when reaching the time deadline T.

CTVMT is more relevant in practice since it considers more realistic factors including time and value. We show that CTVMT problem is NP-hard and propose an algorithm named **BCT-M** to address it. Our algorithm uses the framework of **BCT** algorithm [3] which is an efficient approximation algorithm for CTVM problem. With the help of *Benefit Sampling* (BS) strategies designed elaborately for CTVMT under IC-M and LT-M model respectively, we prove that our algorithm take over guaranteeing a $(1 - 1/\sqrt{e} - \epsilon)$-approximate solution for arbitrary costs and a $(1 - 1/e - \epsilon)$-approximate solution for uniform costs. In summary, the contributions of this paper are as follows.

- We propose the *Cost-aware Targeted Viral Marketing with Time constraints* (CTVMT) problem that consider heterogeneous costs and benefits of users and time constraints, including time delay and time deadline, of influence diffusion. Our problem generalizes other viral marketing problems such as IM, CTVM and time constrained IM.

- We propose a **BCT-M** algorithm using benefit sampling strategies elaborately designed for CTVMT under IC-M and LT-M model. Our algorithm is efficient and has the approximation ratio which is $(1-1/\sqrt{e}-\epsilon)$ for arbitrary costs case and $(1-1/e-\epsilon)$ for uniform case.
- We perform extensive experiments on various real-world datasets. The performance of our algorithm demonstrate its efficiency and effectiveness in finding higher quality seed set satisfying our constraints.

The remainder of this paper is organized as follows. The related work is reviewed in the next section. Section 3 introduces preliminary knowledges and presents the definition of CTVMT problem. In Sect. 4, we present the **BCT-M** algorithm, and in Sect. 5 we analyze the approximation ratio of our algorithm. Section 6 presents the experimental study. Finally, the last section concludes this paper. The key notations used in this paper are given in Table 1.

2 Related Work

Influence Maximization and CTVM. Kempe *et al.* [1] is the first to formulate Influence Maximization (IM) as a discrete optimization problem. They create two classical diffusion models, namely, Independent Cascade (IC) model and Linear Threshold (LT) model. They also prove that IM problem is NP-hard under these two models. However, because of the monotonicity and submodularity of $\sigma(S)$, they propose a greedy algorithm to approximately solve it and prove its approximation ratio is $(1-1/e-\epsilon)$.

The major bottle-neck in IM is calculating the influence spread of any given set and it has been proved to be #P-hard [4,5]. A number of approaches have been proposed to estimate the influence spread [6–8]. Kempe *et al.* [1] use Monte Carlo (MC) simulation method which is computationally expensive so that it is not efficient and scalable. Leskovec *et al.* [9] propose a mechanism named CELF to accelerate MC method with reducing the number of times required to calculate influence spread. Chen *et al.* [10] propose two fast heuristics algorithms, namely DegreeDiscount and PMIA, to select users at each step of the greedy algorithm.

Recently, Borgs *et al.* [11] make a theoretical breakthrough and present an 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 a sequential work, Tang *et al.* [12] reduce the running time and show that their algorithm is also very efficient in billion-scale networks. Nguyen *et al.* [3] extend the IM problem with the addition of assumptions that each user has his own cost and benefit. Hence, they propose Cost-aware Targeted Viral Marketing (CTVM) problem and design a BCT algorithm with the help of Reverse Influence Sampling (RIS) framework. The BCT algorithm is scalable and efficient. Specifically, it has an approximation ratio of $(1-1/\sqrt{e}-\epsilon)$. However, these work above do not consider the temporal factors in reality.

Influence Maximization with Temporal Factors. Chen *et al.* [2] propose the time-critical IM problem with two new diffusion models named IC-M and LT-M model. They incorporate *time delay* denoted by *meeting events* and *time deadline* of propagation into IM problem, and prove some properties of these two models. They propose two heuristic algorithms, namely MIA-M and MIA-C, for IC-M model and LDAG-M algorithm for LT-M model. Liu *et al.* [13] independently propose time-constrained IM problem and LAIC model to simulate the influence propagation process with latency information. They propose an algorithm based on influence spreading path. However, these two work do not consider the heterogeneous costs and benefits of users and their heuristic algorithms can not provide any approximation guarantee.

Table 1. Table of notations

Notation	Definition
n, n'	The number of users and edges in a network G
e_{ij}	The edge in a network from node v_i to v_j
$b(v_i)$	The benefit of v_i when it is influenced
$c(v_i)$	The cost of v_i when it is selected into seed set
$m(e_{ij})$	The meeting probability of edge e_{ij}
$w(e_{ij})$	The propagation probability of edge e_{ij}
B	The budget of seed set
T	The time deadline of influence propagation
Ω	The sum of all user benefits, $\Omega = \sum_{v \in V} b(v)$
$\sigma_T(S)$	The expected number of influenced users by seed set S
$\mathbb{B}_T(S)$	The feedback benefits of seed set S
$deg_H(S)$	The number of hyperedges incident at some user in S
c	$c = 2(e-2) \approx \sqrt{2}$
k_{max}	$k_{max} = \max\{k : \exists S \subset V, \lvert S \rvert = k, c(S) \leq B\}$
Υ_L^u	$\Upsilon_L^u = 8c\left(1 - \frac{1}{2e}\right)^2 \left[\ln \frac{1}{\delta} + \ln \binom{n}{k} + \frac{2}{n}\right] \frac{1}{\epsilon^2}$
Υ_L^c	$\Upsilon_L^c = 8c\left(1 - \frac{1}{2e}\right)^2 \left[\ln \frac{1}{\delta} + k_{max} \ln n + \frac{2}{n}\right] \frac{1}{\epsilon^2}$
Λ_L	$\Lambda_L = \left(1 + \frac{e\epsilon}{2e-1}\right) \Upsilon_L$

3 Preliminaries and Problem Definition

In this section, we will introduce *Independent Cascade with Meeting events* (IC-M) and *Linear Threshold with Meeting events* (LT-M) model [2] briefly as a start. These two models have some useful properties to help us addressing the CTVMT problem. Then we present the definition of the CTVMT problem.

3.1 IC-M and LT-M Diffusion Models

Given a social network denoted by a directed graph $G = (V, E, b, c, m, w)$ in which V is a node set to denote social network users and E is a directed edge set to denote the relationship between users. $|V| = n$, $|E| = n'$ and each node $v_i \in V$ has a cost $c(v_i) > 0$ if v_i is selected into seed set and a benefit $b(v_i) \geq 0$ if v_i is influenced. Each directed edge that from v_i to v_j is denoted as $e_{ij} \in E$, and $m(e_{ij}) \in (0, 1]$ denotes the *meeting probability* that v_i can meets v_j at any time t. Furthermore, Each directed edge $e_{ij} \in E$ is associated with an influence weight $w(e_{ij}) \in (0, 1]$ which denotes the propagation probability from v_i to v_j, and specifically, keep that $\forall v_j \in V, \sum_{v_i \in V} w(e_{ij}) \leq 1$ for the LT-M model.

Given a network G, a *seed set* $S \subseteq V$ and a *time deadline* T, the influence propagation process in G under the IC-M and LT-M model are following respectively.

Propagation Process Under IC-M Model. The influence propagation under IC-M model happens in round $t = 0, 1, 2, 3, \ldots, T$.

- (*Begining*) At round 0, only the nodes in the seed set S are *activated*. All other nodes stay *inactive*. The cost of activating the seed set is $c(S) = \sum_{v_i \in S} c(v_i)$ and that's all we need to pay.
- (*Propagation*) At round $t \geq 1$, an active node v_i can meet its neighbor v_j with the *meeting probability* $m(e_{ij})$. Only at the first meeting events happened to its inactive neighbor v_j, v_i has the only chance to activate v_j with *propagation probability* $w(e_{ij})$.
- (*Stop Condition*) Once a node becomes activated, it remains activated in all subsequent rounds. The influence propagation process stops when *time deadline* T is reached or no more nodes can be activated.

Propagation Process Under LT-M Model. Firstly, every node $v_i \in V$ choose a *threshold* θ_{v_i} uniformly at random in $[0, 1]$ independently. Next the influence propagation happens in round $t = 0, 1, 2, 3, \ldots, T$. LT-M model has the same *Begining* and *Stop Condition* as IC-M model, hence we only introduce the *Propagation* of LT-M to save space.

- (*Propagation*) At round $t \geq 1$, an active node can meet its neighbor v_j with the *meeting probability* $m(e_{ij})$, once the meeting events happened to its inactive neighbor v_j, it's an *effective* active neighbor for v_j. An inactive node v_j can be activated if and only if the weighted number of its *effective* active neighbors reaches its threshold, i.e., $\sum_{\textit{effective active neighbor } v_i} w(e_{ij}) \geq \theta_{v_j}$.

The *feedback benefits* of S in G under the IC-M and LT-M model, denoted by $\mathbb{B}_T(S)$, is defined as the expected benefits sum of nodes activated finally including seed set S.

Properties of IC-M and LT-M Models. IC-M and LT-M models are shown in [2] to be equivalent to the reachability in a random graph X. We call it *sample graph with meeting events* hereafter. We will introduce the definition of this random graph of two models respectively.

For IC-M model, its *sample graph with meeting events* is defined as follows. Given a graph $G = (V, E, m, w)$, for every $e_{ij} \in E$, we flip a coin once with bias $w(e_{ij})$, and we declare the edge *live* with probability $w(e_{ij})$ or *blocked* with probability $1 - w(e_{ij})$. Then, for each meeting event (a directed e_{ij} edge and a time step $t \in [1, T]$), we flip a coin with bias $m(e_{ij})$ to determine if v_i will meet v_j at t. All coin-flips are independent. Therefore, a certain set of outcomes of all coin flips corresponds to a *sample graph with meeting events*, denoted by $X = X_M \cdot X_E$, which is a deterministic graph (with all blocked edges removed) obtained by conditioning on that particular set of outcomes, where X_M is a set of outcomes of all *meeting events* and X_E is a set of outcomes of all *live-or-blocked* identities for all edges. Since the coin-flips for meeting events and those for live-edge selections are orthogonal, and all flips are independent, any X_E on top of a X_M leads to a sample graph with meeting events X.

As for LT-M model, for each meeting event, we also flip a coin with bias $m(e_{ij})$ to determine if v_i will meet v_j at t. But for every node $v_j \in V$, we select at most one of its incoming edges at a random, such that the edge is selected with probability $w(e_{ij})$, and no edge is selected with probability $1 - \sum_{v_i} w(e_{ij})$. The selected edge is called *live* and all other edges are called *blocked.* And for the same reason, there is a *sample graph with meeting events* $X = X_M \cdot X_E$ with the same definitions of X, X_E and X_M.

Let $\mathcal{E}_I$ denote the event that I is the true realization of the corresponding random process. By Theorems 2 and 4 in [2], the influence spread of a seed set S equals the expected number of nodes reachable within deadline T from S over all sample graph with meeting events of IC-M and LT-M respectively, i.e.,

$$\sigma_T(S) = \sum_{X \sqsubseteq G} \Pr[\mathcal{E}_X] \cdot |\sigma_T^X(S)| \tag{1}$$

Where $\sqsubseteq$ denotes that the sample graph with meeting events X is generated from G with probability denoted by $\Pr[\mathcal{E}_X] = \Pr[\mathcal{E}_{X_E}] \cdot \Pr[\mathcal{E}_{X_M}]$. And $\sigma_T^X(S)$ denotes the set of nodes reachable from S in X within a deadline T.

Similarly, the *feedback benefits* of a seed set S equals the expected benefit sum of nodes reachable within deadline T from S over all sample graph with meeting events of IC-M and LT-M respectively, i.e.,

$$\mathbb{B}_T(S) = \sum_{X \sqsubseteq G} \Pr[\mathcal{E}_X] \sum_{v \in \sigma_T^X(S)} b(v) \tag{2}$$

And we can get the following theorem easily with some modifications of Theorems 2 and 4 in [2]. We omit the proof of it due to space constraint.

Theorem 1. *The feedback benefits function $\mathbb{B}_T(S)$ is monotone and submodular for an arbitrary instance of the IC-M and LT-M model, given a deadline constraint $T \geq 1$.*

3.2 Problem Definition

We propose the definition of *Cost-aware Targeted Viral Marketing with Time constraints* (CTVMT) problem in this subsection. Informally, CTVMT aims to find a set of user within a budget B in a social network such that its feedback benefits is maximum when time deadline T is reached.

Definition 1 (CTVMT). *Given a budget B and a time deadline T, the Cost-aware Targeted Viral Marketing with Time constraints (CTVMT) problem aims to select a seed set $S \subseteq V$ in a social network $G = (V, E, b, c, m, w)$:*

$$S = \arg \max_{S \subseteq V, c(S) \leq B} \mathbb{B}_T(S) \tag{3}$$

where $\mathbb{B}_T(S)$ is the feedback benefits of S in G, i.e., the benefits sum of users activated finally including seed set S.

Unfortunately, the CTVMT problem is NP-hard.

Theorem 2. *The CTVMT problem is NP-hard.*

Proof. The Cost-aware Targeted Viral Marketing (CTVM) problem is NP-hard [3]. It can be regarded as a special case of the CTVMT problem where the *meeting probability* is 1, i.e., $\forall e_{ij} \in E,\ m(e_{ij}) = 1$, and the time deadline is ∞. Hence the CTVMT problem is NP-hard.

4 BCT-M Algorithm

In this section, we present **BCT-M** algorithm to solve the CTVMT problem. We will firstly introduce the RIS approach briefly, which is the foundation of our algorithm. Then we present our **BCT-M** framework [3] and two new sampling algorithms for IC-M and LT-M model respectively, namely **BS-IC-M** and **BS-LT-M** algorithm.

4.1 Summary of the RIS Approach

Reverse Influence Sampling (RIS) is a novel approach for IM to estimate the influence spread of any given nodes set proposed by Borgs *et al.* [11]. We extend the RIS framework under IC-M and LT-M models.

Given $G = (V, E, m, w)$, RIS captures the influence landscape of G through generating a hypergraph $H = (V, \{\varepsilon_1, \varepsilon_2, \dots\})$. Each hyperedge $\varepsilon_j \in H$ is a set of nodes in V and constructed as follows.

Definition 2. *Given $G = (V, E, m, w)$ and a time deadline T, a **random hyperedge** ε_j is generated from G by **(1)** selecting a random node $v \in V$, **(2)** generating a sample graph with meeting events $X \sqsubseteq G$ and **(3)** returning ε_j as the set of nodes that can reach v in X within the time deadline T.*

Node v in the above definition is called the *source* of ε_j and denoted by $\mathbf{src}(\varepsilon_j)$. Observe that ε_j contains the nodes that can influence its source v within time deadline T. 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 $\sigma_T(S)$. Here a seed set S covers a hyperedge ε_j, if $S \cap \varepsilon_j \neq \emptyset$. This observation is captured in the following lemma in [11].

Lemma 1 **([11]).** *Given $G = (V, E, m, w)$ and a random hyperedge ε_j generated from G. For each seed set S,*

$$\sigma_T(S) = n \cdot \Pr[S \textit{ covers } \varepsilon_j] \tag{4}$$

Time Constrained RIS Framework. Based on the above lemma, the time constrained IM problem can be solved using the following time constrained RIS framework.

- Generate multiple random hyperedges from G using sample graph with meeting events model.
- Use the greedy algorithm for the Max-Coverage problem to find a seed set S that covers the maximum number of hyperedges and return S as the solution.

Nguyen *et al.* [3] extended RIS to estimate feedback benefits $\mathbb{B}(S)$. They modified the RIS framework to find a seed set S that covers the maximum *weighted* number of hyperedges, where the weight of a hyperedge ε_j is the benefit of the source src(ε_j). Given a seed set $S \subset V$, define a random variable $x_j = b(src(\varepsilon_j)) \times 1_{(S \textit{ covers } \varepsilon_j)}$, i.e., $x_j = b(src(\varepsilon_j))$ if $S \cap \varepsilon_j \neq \emptyset$ and $x_j = 0$, otherwise. They showed similar to Lemma 1, that

$$\mathbb{B}(S) = n \cdot \mathbb{E}[x_j] \tag{5}$$

We can also use the same method to extend time constrained RIS framework to estimate feedback benefits $\mathbb{B}_T(S)$, i.e.,

$$\mathbb{B}_T(S) = n \cdot \mathbb{E}[x_j] \tag{6}$$

4.2 BCT-M Framework and Two New Sampling Algorithms

We present **BCT-M** framework which is proposed in [3] and two new sampling algorithms designed for IC-M and LT-M model respectively in this subsection. We note that our algorithm with new sampling algorithms takes over the approximation ratio of **BCT** algorithm. To the best of our knowledge, our algorithm is the first one addressing the CTVMT problem with an approximation ratio.

BCT-M algorithm [3] for the CTVMT problem is presented in Algorithm 1. The algorithm uses **BS** algorithm, either **BS-IC-M** (Algorithm 3) under IC-M model or **BS-LT-M** (Algorithm 4) under LT-M model, to generate hyperedges and **Weighted-Max-Coverage** (Algorithm 2) [3] to find a candidate seed set S following the time constrained RIS framework. The algorithm runs in rounds and after each round, **Weighted-Max-Coverage** algorithm is called to select a seed set $\hat{S}$ within the budget B and stop the algorithm if the degree of $\hat{S}$ exceeds Λ_L. Otherwise, it continues to generate more hyperedges.

The **Weighted-Max-Coverage** algorithm [3] can find a maximum cover within the budget B. It considers two candidates: one is taken from greedy strategy and the other is just a node having highest coverage within the budget, then it return the one with higher coverage. Khuller *et al.* [14] proved that

Algorithm 1. BCT-M

```
Input: $G = (V, E, b, c, m, w), B, T, \epsilon, \delta \in (0, 1)$.
Output: Seed set $S$.
$\Upsilon_L = \Upsilon_L^u$ for uniform cost and $\Upsilon_L = \Upsilon_L^c$ otherwise ;
$\Lambda_L = (1 + \frac{e\epsilon}{2e-1})\Upsilon_L$ ;
$N_t = \Lambda_L, \hat{S} \leftarrow \emptyset$ ;
$H \leftarrow (V, \varepsilon = \emptyset)$ ;
while $deg_H(\hat{S}) < \Lambda_L$ do
    for $j = 1$ to $N_t - |\varepsilon|$ do
        Generate $\varepsilon_j \leftarrow$ BS$(G, T)$ ;
        Add $\varepsilon_j$ to $\varepsilon$.
    $N_t = 2N_t$ ;
    $\hat{S} =$ Weighted-Max-Coverage$(H, B)$ ;
Return $\hat{S}$.
```

Algorithm 2. Weighted-Max-Coverage

```
Input: Hypergraph $H$ and Budget $B$.
Output: Seed set $S$.
$S \leftarrow \emptyset$ ;
while $P = \{v \in V \setminus S | c(v) \leq B - c(S)\} \neq \emptyset$ do
    $v^* \leftarrow \arg\max_{v \in P} \frac{deg_H(S \cup \{v\}) - deg_H(S)}{c(v)}$
    $S \leftarrow S \cup \{v^*\}$ ;
$u \leftarrow \arg\max_{\{v \in V | c(v) \leq B\}} deg_H(\{v\})$ ;
if $deg_H(S) < deg_H(\{u\})$ then
    $S \leftarrow \{u\}$ ;
Return $S$.
```

this procedure returns a $(1 - 1/\sqrt{e} - \epsilon)$-approximate cover with arbitrary cost. However, if the node cost is uniform, this algorithm only considers the candidate obtained from greedy strategy and has the approximation factor of $(1 - 1/e - \epsilon)$.

We design two new benefit sampling strategies for IC-M and LT-M model respectively to estimate feedback benefit $\mathbb{B}_T(S)$, and we present these as **BS-IC-M** (Algorithm 3) and **BS-LT-M** (Algorithm 4) algorithm.

BS-IC-M algorithm is designed for generating a random hyperedge $\varepsilon_j \subseteq V$ under IC-M model. We use Breadth First Search (BFS) to find all the nodes can reach source node within time constraints, so we structure two queues to help out. We do some initialization and structure two empty queues firstly (line 1–2). Then we choose the source node with the probability of choosing node v_j is $P(v_j) = b(v_j)/\Omega$ where $\Omega = \sum_{v \in V} b(v)$ (line 3). This is the great deal of difference designed for the **heterogeneous benefits** situation, and we will prove that it is the foundation of the accuracy of the estimation of the feedback benefits. Insert source node v_j into candidate queue Q and insert $t_q = 0$ into

Algorithm 3. BS-IC-M

```
   Input: $G = (V, E, m, w), T$.
   Output: A random hyperedge $\varepsilon_j \in V$.
1  $\varepsilon_j \leftarrow \emptyset, t_q \leftarrow 0, \Delta t \leftarrow 0$ ;
2  Two queues $Q \leftarrow \emptyset$ and $Q_t \leftarrow \emptyset$ ;
3  Pick a node $v_j$ with probability $\frac{b(v_j)}{\Omega}$ ;
4  Insert $v_j$ into $Q$ and $t_q = 0$ into $Q_t$ ;
5  while $Q \neq \emptyset$ do
6      $v_q \leftarrow$ extract the first node in $Q$ ;
7      $t_q \leftarrow$ extract the first time stamp in $Q_t$;
8      Add $v_q$ to $\varepsilon_j$ ;
9      Attempt to select all live-edges $e_{iq}$ with probability $w(e_{iq})$ ;
10     foreach edge $e_{iq}$ is selected do
11         if $v_i \notin Q$ then
12             Flip a coin with bias $m(e_{iq})$ until $v_i$ meet $v_q$ and record the number
               of coin-flips with $\Delta t$ ;
13             if $t_q + \Delta t \leq T$ then
14                 Insert $v_i$ into $Q$ and $t_q + \Delta t$ into $Q_t$ ;
15 Return $\varepsilon_j$.
```

time stamp queue Q_t (line 4), then we repeat a loop until Q is empty. In each iteration, we extract the first node in Q as v_q and the corresponding time stamp t_q in Q_t, then add v_q to ε_j (line 6–8). Next, for every edge e_{iq} that can reach v_q, we flip a coin with bias $w(e_{iq})$ to determine if it's *live* (line 9). And for every live-edge e_{iq}, if it hasn't been selected before, we flip a coin with bias $m(e_{iq})$ until v_i meets v_q and we record the number of coin-flips with Δt. If $t_q + \Delta t \leq T$ which means not reach the time deadline, we insert v_i and $t_q + \Delta t$ into Q and Q_t respectively (line 10–14). Finally, when Q is empty, we can get a hyperedge ε_j (line 15).

As for **BS-LT-M** algorithm, it's designed for LT-M model to generate a random hyperedge $\varepsilon_j \subset V$. We also do some initialization and pick the source node with probability $P(v_j) = b(v_j)/\Omega$ (line 1–2). According to the sampling graph with meeting events for LT-M model, we iteratively select node to structure ε_j. Firstly we add source node v_j into ε_j. Then we select at most one of its incoming edges at a random, such that the edge is selected with probability $w(e_{ij})$, and no edge is selected with probability $1 - \sum_{v_i} w(e_{ij})$ (line 5). If edge e_{ij} is selected, then we flip a coin with bias $m(e_{ij})$ until v_i meets v_j and also record the number of coin-flips using Δt and refresh v_j and t (line 6–9). The iteration breaks until v_j has been selected already or $t > T$ (line 3), as well as we choose no edge (line 10–11). Finally, we get a hyperedge ε_j under LT-M model (line 12).

Algorithm 4. BS-LT-M

```
Input: $G = (V, E, m, w), T$.
Output: A random hyperedge $\varepsilon_j \in V$.
$\varepsilon_j \leftarrow \emptyset, t \leftarrow 0, \Delta t \leftarrow 0$ ;
Pick a node $v_j$ with probability $\frac{b(v_j)}{\Omega}$ ;
while $v_j \notin \varepsilon_j \wedge t \leq T$ do
    Add $v_j$ to $\varepsilon_j$ ;
    Attempt to select at most one edge $e_{ij}$ with probability $w(e_{ij})$ or no edge with probability $1 - \sum_{v_i} w(e_{ij})$ ;
    if edge $e_{ij}$ is selected then
        Flip a coin with bias $m(e_{ij})$ until $v_i$ meet $v_j$ and record the number of coin-flips with $\Delta t$ ;
        Set $v_j \leftarrow v_i$ ;
        $t \leftarrow t + \Delta t$ ;
    else
        Break;
Return $\varepsilon_j$.
```

5 Approximation Analysis

In this section, we prove that **BCT-M** returns a $(1 - 1/e - \epsilon)$-approximate solution for uniform cost version of CTVMT problem and a $(1 - 1/\sqrt{e} - \epsilon)$ solution for the arbitrary cost version under IC-M and LT-M model using corresponding benefit sampling algorithm.

Our proof is following the same way of [3]. They prove the approximation ratio of **BCT** algorithm which is the framework of our algorithm. But with the new benefit sampling algorithms, namely, **BS-IC-M** and **BS-LT-M** algorithm, we need to prove that we can following the same way to get the same approximation ratio. The foundation of their proof is that each hyperedge generated by their sampling algorithm is equivalent to a random sampling using sample graph model to estimate influence spread. So we need to prove the equivalence of random hyperedges generated via **BS-IC-M** or **BS-LT-M** under IC-M or LT-M model respectively.

Lemma 2. *Any hyperedge ε_j generated via **BS-IC-M** or **BS-LT-M** is equivalent to structure a random hyperedge using sample graph with meeting events X under IC-M or LT-M model respectively.*

Proof. We only prove the situation under LT-M model and the other one can easily get using the same method. It's sufficient to prove that for any hyperedge ε_j generated via **BS-LT-M**, there exist at least one sample graph with meeting events X that can also get the same hyperedge with the same source node. It's obviously that we fix the edges in ε_j *live* and for every node $v_j \notin \varepsilon_j$, we select at most one of its incoming edges at a random, such that the edge is selected with probability $w(e_{ij})$, and no edge is selected with probability $1-\sum_{v_i} w(e_{ij})$. And we also fix the meeting events time stamp in ε_j and flip coin with bias $m(e_{ij})$ for others in time range $t \in [1, T]$. Hence we get a sample graph with meeting events X in which using the same source node of ε_j, we can get the same random hyperedge as ε_j.

We have proved that generate hyperedges via our algorithms is equivalent to structure hyperedges using sample graph with meeting events model. The lemma above clarify that each hyperedge generated by sampling algorithm can be regarded as a random sampling to estimate $\mathbb{B}_T(S)$. And now we prove that selecting source node u with probability $P(u) = b(u)/\Omega$, we can use these hyperedges to estimate $\mathbb{B}_T(S)$ using following equation.

Lemma 3. *Given a fixed seed set $S \subseteq V$, for a random hyperedge ε_j,*

$$\Pr[\varepsilon_j \cap S \neq \emptyset] = \frac{\mathbb{B}_T(S)}{\Omega} \tag{7}$$

Proof.

$$\begin{aligned}
\mathbb{B}_T(S) &= \sum_{u \in V} \Pr_{X \sqsubseteq G}[u \in \sigma_T^X(S)] b(u) \\
&= \sum_{u \in V} \Pr_{X \sqsubseteq G}[\exists v \in S \textit{ such that } v \in \varepsilon_j(u)] b(u) \\
&= \Omega \sum_{u \in V} \Pr_{X \sqsubseteq G}[\exists v \in S \textit{ such that } v \in \varepsilon_j(u)] \frac{b(u)}{\Omega} \\
&= \Omega \Pr_{X \sqsubseteq G, u \in V}[\exists v \in S \textit{ such that } v \in \varepsilon_j(u)] \\
&= \Omega \Pr_{X \sqsubseteq G, u \in V}[\varepsilon_j \cap S \neq \emptyset]
\end{aligned} \tag{8}$$

Lemmas 2 and 3 clarify that our benefit sampling algorithms under IC-M and LT-M model have the properties: **(1)** equivalent to random sampling graph with meeting events model and **(2)** can estimate $\mathbb{B}_T(S)$ effectively. These properties are the same as **BSA** algorithm in [3]. Based on Lemmas 2 and 3 above, we can following the same way of the proof of approximation ratio in [3] directly to get the approximation guarantee of **BCT-M** under two diffusion models. One can get the details by reviewing [3].

Theorem 3 **([3]).** *Given a budget B, $0 \leq \epsilon \leq 1$ and $0 \leq \delta \leq 1$. **BCT-M** for **uniform** cost CTVMT problem under IC-M or LT-M model using corresponding benefit sampling algorithm returns a solution $\hat{S}$,*

$$\mathbb{B}_T(\hat{S}) \geq (1 - 1/e - \epsilon) OPT_T \tag{9}$$

with probability at least $(1 - \delta)$.

Theorem 4 ([3]). *Given a budget B, $0 \leq \epsilon \leq 1$ and $0 \leq \delta \leq 1$.* ***BCT-M*** *for* ***arbitrary*** *cost CTVMT problem under IC-M or LT-M model using corresponding benefit sampling algorithm returns a solution $\hat{S}$,*

$$\mathbb{B}_T(\hat{S}) \geq (1 - 1/\sqrt{e} - \epsilon)OPT_T \tag{10}$$

with probability at least $(1 - \delta)$.

6 Performance Study

6.1 Experimental Setup

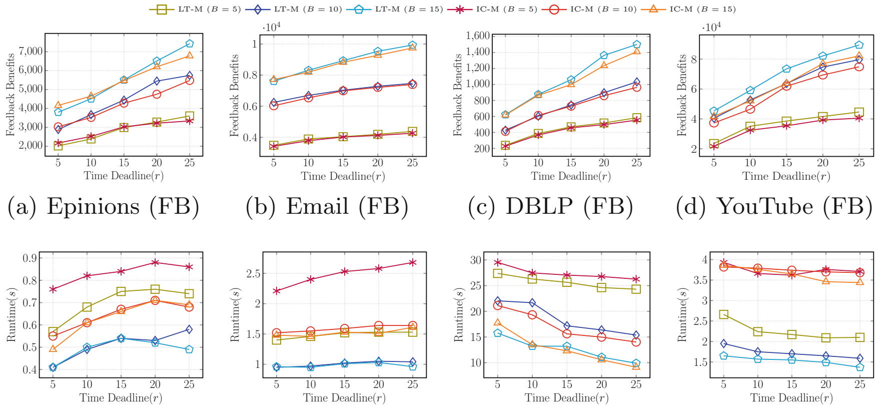
Datasets. We use 4 datasets downloaded from [15] in our experiments and we show their properties in Table 2. **Epinions** was generated from a who-trust-whom online social network site Epinions.com. There is an edge from v_i to v_j if v_j *trust* v_i. **Email** was generated using email data from a large European research institution. There is an edge e_{ij} in the network if person v_i sent person v_j at least one email. **DBLP** construct a co-authorship network where two authors are connected if they publish at least one paper together. In the **YouTube** social network, users form friendship each other and there is an edge between friends. **DBLP** and **YouTube** are undirected, so we do some preprocess to divide one undirected edge into two directed edges with opposite directions. Hence, the total number of edges of these two datasets is twice the initial value.

Table 2. Dataset properties

Property	Epinions	Email	DBLP	YouTube
Type	Directed	Directed	Undirected	Undirected
# of nodes	75,888	265,214	425,877	1,157,806
# of edges initially	508,837	420,045	1,049,866	2,987,624
# of edges finally	508,837	420,045	2,099,732	5,975,248

Graph Parameters. we remark that our solutions are orthogonal to the techniques for generating influence probability [16]. Hence, we consider generating propagation probability in case where the probability on edge e_{ij} is set to be $\frac{1}{N_{in}(v_j)}$, where $N_{in}(v_j)$ is the in-degree of v_j. We generate each user's cost and benefit randomly in $[1, 2]$ with two decimal places. As for meeting probability, we consider two methods of genetating it. (a) **Degree.** The meeting probability on edge e_{ij} is set to be $\frac{N_{out}(v_i)}{N_{out}(v_i)+N_{out}^{max}(v)} + 0.1$, where $N_{out}(v_i)$ is the out-degree of v_i and $N_{out}^{max}(v)$ is the maximum out-degree value. The meeting probability is ranging from $(0.1, 0.6]$. (b) **Random.** The meeting probability on edge e_{ij} is chosen uniformly from the set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. In all experiments, we set $\delta = 0.01$ and $\epsilon = 0.1$.

Performance Measures. We evaluate **BCT-M** algorithm with **BS-IC-M** and **BS-LT-M** sampling algorithm under IC-M and LT-M diffusion model respectively. For each dataset, we conduct experiments under two models respectively with the same graph parameters. We consider two performance measures, (a) **Feedback Benefits.** The benefits sum of all activated users finally. (b) **Runtime.** The runtime of algorithm. All experiments are run on Mac OS X EI Capitan system with Intel Core i5 2.6 GHz DUO core CPU and 8 GB memory.



(a) Epinions (FB) (b) Email (FB) (c) DBLP (FB) (d) YouTube (FB)

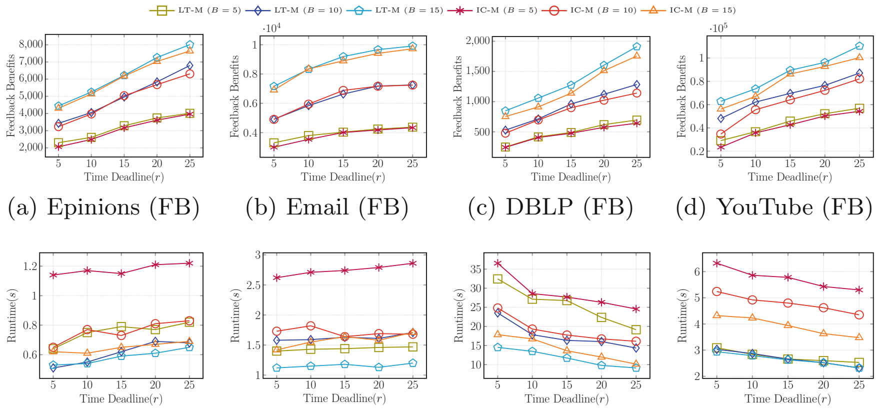
(e) Epinions (RT) (f) Email (RT) (g) DBLP (RT) (h) YouTube (RT)

Fig. 1. Feedback benefits and runtime vs. time deadline and budget with **Degree** meeting probability (FB: Feedback Benefits; RT: Runtime)

6.2 Experimental Results

We set budget to 5,10,15 and for each budget, we range the time deadline from 5 to 25 rounds, denoted by r, to see the value of feedback benefits and runtime under two diffusion models with degree and random meeting probability respectively. We run 50 times for each budget and time deadline combination. Figures 1 and 2 show the average feedback benefits and runtime of the **BCT-M** algorithm under two diffusion models with degree and random meeting probability respectively.

Feedback Benefits. From Figs. 1 and 2, We can see that feedback benefits with more budget is larger than those with less budget, and it always getting bigger with bigger time deadline under all budget settings for all datasets. This phenomenon prove that given more budget to select seed set and given more time to spread influence, we can get more feedback benefits finally. Besides, We can see that feedback benefits under random meeting probability is a little larger than its corresponding instance under degree meeting probability. We think it is because that degree meeting probability is usually small than random meeting probability in those datasets which has so many nodes and edges.



(a) Epinions (FB) (b) Email (FB) (c) DBLP (FB) (d) YouTube (FB)

(e) Epinions (RT) (f) Email (RT) (g) DBLP (RT) (h) YouTube (RT)

Fig. 2. Feedback benefits and runtime vs. time deadline and budget with **Random** meeting probability (FB: Feedback Benefits; RT: Runtime)

Runtime. From Figs. 1 and 2, we can see that runtime of all settings is reasonable. However, there is an interesting phenomenon that when the combination of budget and time deadline gets bigger, the runtime gets smaller. We consider the reason lies in the **BCT-M** algorithm which iteratively generate *hyperedges* to ensure the approximate accuracy. When budget and time deadline get bigger, the number of iteration gets smaller but also can reach the approximate accuracy, hence the total runtime is smaller than before.

Scalability. The runtime of **BCT-M** algorithm is reasonable even with 10^6 nodes and edges in **DBLP** and **YouTube** datasets. We also study the scalable of this algorithm and it scale well with the graph size. We omit detailed results due to space constraint.

7 Conclusion

In this paper, we incorporate the time delay and time deadline of influence diffusion which is concluded as time constraints as well as heterogeneous costs and benefits of users in a social network in IM problem to propose *Cost-aware Targeted Viral Marketing with Time constraints* (CTVMT) problem. We prove that it's NP-hard and we propose a **BCT-M** algorithm to get a solution with approximation guarantee under IC-M and LT-M model using the corresponding benefit sampling algorithm. Our empirical study over several real-world datasets demonstrates the efficiency and effectiveness of our algorithm.

Acknowledgements. This work is partially supported by National Natural Science Foundation of China (NSFC) under Grant No. 61772491, No. 61472460, and Natural Science Foundation of Jiangsu Province under Grant No. BK20161256, and Anhui Initiative in Quantum Information Technologies AHY150300.

References

1. Kempe, D., Kleinberg, J., Tardos, É.: Maximizing the spread of influence through a social network. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 137–146. ACM (2003)
2. Chen, W., Lu, W., Zhang, N.: Time-critical influence maximization in social networks with time-delayed diffusion process. arXiv preprint arXiv:1204.3074 (2012)
3. Nguyen, H.T., Dinh, T.N., Thai, M.T.: Cost-aware targeted viral marketing in billion-scale networks. In: INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications, pp. 1–9. IEEE (2016)
4. Chen, W., Wang, C., Wang, Y.: Scalable influence maximization for prevalent viral marketing in large-scale social networks. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1029–1038. ACM (2010)
5. Chen, W., Yuan, Y., Zhang, L.: Scalable influence maximization in social networks under the linear threshold model. In: 2010 IEEE 10th International Conference on Data Mining (ICDM), pp. 88–97. IEEE (2010)
6. Jung, K., Heo, W., Chen, W.: IRIE: scalable and robust influence maximization in social networks. In: 2012 IEEE 12th International Conference on Data Mining (ICDM), pp. 918–923. IEEE (2012)
7. Ohsaka, N., Akiba, T., Yoshida, Y., Kawarabayashi, K.-I.: Fast and accurate influence maximization on large networks with pruned Monte-Carlo simulations. In: AAAI, pp. 138–144 (2014)
8. Goyal, A., Lu, W., Lakshmanan, L.V.S.: SIMPATH: an efficient algorithm for influence maximization under the linear threshold model. In: 2011 IEEE 11th International Conference on Data Mining (ICDM), pp. 211–220. IEEE (2011)
9. Leskovec, J., Krause, A., Guestrin, C., Faloutsos, C., VanBriesen, J., Glance, N.: Cost-effective outbreak detection in networks. In: Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 420–429. ACM (2007)
10. Chen, W., Wang, Y., Yang, S.: Efficient influence maximization in social networks. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 199–208. ACM (2009)
11. Borgs, C., Brautbar, M., Chayes, J., Lucier, B.: Maximizing social influence in nearly optimal time. In: Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 946–957. SIAM (2014)
12. Tang, Y., Xiao, X., Shi, Y.: Influence maximization: near-optimal time complexity meets practical efficiency. In Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data, pp. 75–86. ACM (2014)
13. Liu, B., Cong, G., Xu, D., Zeng, Y.: Time constrained influence maximization in social networks. In: 2012 IEEE 12th International Conference on Data Mining (ICDM), pp. 439–448. IEEE (2012)
14. Khuller, S., Moss, A., Naor, J.S.: The budgeted maximum coverage problem. Inf. Process. Lett. **70**(1), 39–45 (1999)
15. Leskovec, J., Krevl, A.: SNAP datasets: stanford large network dataset collection, June 2014. http://snap.stanford.edu/data
16. Goyal, A., Bonchi, F., Lakshmanan, L.V.S.: 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)

17. Nguyen, H.T., Thai, M.T., Dinh, T.N.: Stop-and-stare: optimal sampling algorithms for viral marketing in billion-scale networks. In Proceedings of the 2016 International Conference on Management of Data, pp. 695–710. ACM (2016)
18. Nguyen, H., Zheng, R.: On budgeted influence maximization in social networks. IEEE J. Sel. Areas Commun. **31**(6), 1084–1094 (2013)
19. Ohsaka, N., Yamaguchi, Y., Kakimura, N., Kawarabayashi, K.-I.: Maximizing time-decaying influence in social networks. In: Frasconi, P., Landwehr, N., Manco, G., Vreeken, J. (eds.) ECML PKDD 2016. LNCS (LNAI), vol. 9851, pp. 132–147. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46128-1_9
20. Mossel, E., Roch, S.: On the submodularity of influence in social networks. In: Proceedings of the Thirty-Ninth Annual ACM Symposium on Theory of Computing, pp. 128–134. ACM (2007)
21. Nguyen, H.T., Nguyen, T.P., Vu, T.N., Dinh, T.N.: Outward influence and cascade size estimation in billion-scale networks. Proc. ACM Meas. Anal. Comput. Syst. **1**(1), 20 (2017)
22. Tang, Y., Shi, Y., Xiao, X.: Influence maximization in near-linear time: a martingale approach. In: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pp. 1539–1554. ACM (2015)
23. Song, C., Hsu, W., Lee, M.L.: Targeted influence maximization in social networks. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 1683–1692. ACM (2016)