Minimum Cost Seed Set for Competitive Social Influence

Yuqing Zhu*, Deying Li,†§ and Zhao Zhang‡
*Department of Computer Science, California State University, Los Angeles. Email: yuqing.zhu@calstatela.edu
†College of Information, Renmin University of China. Email: deyingli@ruc.edu.cn
‡Department of Computer Science, Zhejiang Normal University. Email: zhaozhang@zjnu.cn
§Corresponding Author.

***Abstract*—We wonder that in a competitive environment, how an influence uses the minimum cost to choose seeds such that its influence spread can reach a desired threshold under thwarting from its competitors. At first we take a simple fact into account: the information arriving first has heavy impact, and present *Competitive - Independent Cascade (C-IC)* model to characterize how different influences competing with others in a social network. We have found that a specific influence's spread is monotone and submodular, and these nice properties make algorithm performance tractable. We then propose *Minimum Cost Seed Set problem (MinSeed)* to answer our original concern and give a greedy algorithm. We analyze the ratio of greedy algorithm, and give result significantly better than similar ones analyzed by others. Noticing that the computation of real information spread is hard to compute and simple greedy is too time consuming, we design an effective method for estimating information spread in C-IC model, and devise scalable algorithm applying for large social networks. Through simulation on real world datasets, we confirm that, our scalable algorithm outputs seed set with small total cost comparable to that given by simple greedy, with very fast computation.**

***Index Terms*—Competitive influence, Seed minimization, IC model, submodularity.**

I. Introduction

Social networks are graphs consist of individuals and their relationships. The extreme boom of online social networking sites like Facebook, Twitter, Weibo, etc. has been witnessed in the past decade, and lots of real social network data have been obtained thanks to these online websites. In social networks, the most significant phenomenon is "diffusion", which contains the diffusion of news, innovations, and product adoptions, etc. Diffusion is a pervasive characteristic of social networks, and people commonly regard such diffusion as influence propagation.

Influence propagation is the basis of *viral marketing* [1], [2], which is marketing via "word of mouth" recommendations. How to identify "opinion leaders" to bring large further recommendations, i.e., propagated information is an essential issue. To address this issue, the first step is to model information propagation in social networks.

Single source influence propagation model has been studied extensively. In [3] Kempe et al. proposed the two most popular and basic single-influence diffusion models including independent cascade (IC) model and linear threshold (LT) model. Then, noticing that binary opposition widely exist in our life, such as ruling party vs. opposition party, truth vs. rumor, people proposed two-influence propagation models [4], [5], [6]. Nevertheless these models are still not general enough, since it is common where multiple influences exist and compete with others. To this end, in [7], [8] the authors studied the multiple-influence propagation modeling.

Although some effort has been put into modeling real-world influence competition, several issues remain unresolved. First, all models assume that the influence propagation start with some seeds, i.e., the nodes that are initially belonged to the influence and spread its information in the beginning. In reality, multiple influences may use same nodes to propagate their information, which means seeds may overlap. But very few models considered this point. Second, when different influences compete against each other, the timeliness plays a significant role. It goes without saying that a node is more inclined to accept the information that comes first. However, people have been overlooking this obvious fact for a long time: they assume that a node will wait for a period, and all influences coming within that period have equal impacts on the node's final decision. Therefore, in this paper, we propose *Competitive - Independent Cascade (C-IC)* model for competitive influence propagation in social networks, we believe it is more natural since it is the first model that takes above two common but neglected phenomena into account.

The ultimate goal of modeling influence propagation is to correctly spread the information. So far, people paid most attention to influence maximization problem, which is to maximize the information spread when the number of seeds are given. However, sometimes the influence propagators, i.e., companies, agents, etc., have no idea of how to decide the budgets for seeds. Instead, they only have expectations of how many people should know them, which can be approximately quantified as the information spread. Therefore, to make the information spread reach a critical threshold is desirable by companies and agents trying to promote products through viral marketing campaigns in social networks. Since seeds are the nodes that initiate the information propagation, and companies usually give seeds free samples or pay them advertising fees, to minimize the number of seeds, is greatly needed. Hence, choosing minimum number of seeds while guaranteeing that the information spread reach some expected threshold, should be studied due to its significance. However, this problem is

978-1-4673-9953-1/16/$31.00 ©2016 IEEE

not as widely studied as influence maximization problem.

To address the issues mentioned in above two paragraphs, we study how to choose the seeds with minimum cost for a influence competing with others in a social network, such that information spread of this influence reaches a preset threshold. We name our problem *Minimum Cost Seed Set (MinSeed)*. *MinSeed* formulation departs from previous optimization problems like [3], [9], [10], [11] and [12] etc. basing on social influence diffusion, in which: 1) *MinSeed* seeks to minimize the cost when given an objective value, however [3] [9] [10] seeks to maximize the objective when given the cost. Our problem considers the information spread optimization in a totally opposite direction. 2) *MinSeed* thinks the information spread in completely competitive environment, which differs from [11] and [12] that ignores the potential competitors that may hinder a specific influence's information spreading.

Our contributions are as follows:

1) We consider the fact that information arriving first has greater impact than that of the information arriving later, and propose a more natural competitive influence propagation model: C-IC model, which allows overlapping seeds. We prove that under C-IC model, the information spread is a monotone and submodular set function. These two nice properties give the clues for solving C-IC related optimization problems, since with them the performance of simple greedy algorithm becomes tractable.
2) We propose *MinSeed* problem that minimizes the cost of seeds while guaranteeing that the information spread of these seeds under C-IC model reaches a given threshold. As far as we know, this is the first time in its kind when the competitors' existence and interruption are considered. We analyze the performance ratio of simple greedy, this ratio applies not only for our problem but also for all its kind. Our ratio significantly bests existing ones. Observing that even simple greedy is too time costly for large social graphs, we design a scalable algorithm, *LSS-Greedy* that employs an effective estimation of real information spread to reduce the computation cost.
3) We have tested our algorithms on real-world social networks of different scales: relatively small data are used to compare the performances of *LSS-Greedy* and simple greedy, and we have found that the performance of *LSS-Greedy* is very comparable. Large data are used to test the scalability of *LSS-Greedy* since simple greedy becomes infeasible, and we have found that *LSS-Greedy* still outputs results in a quick manner for large networks.

II. Related Work

Many social influence propagation models have been proposed. Kempe et al. in [3] based on work [2] proposed their single influence propagation IC and LT models, and proved that the influence maximization under both IC and LT models are submodular which leads to a $(1-1/e)$ ratio greedy algorithm. Many follow-up research [13], [9], [14], [15] focused on the scalability improvement since the influence computation is too costly. Several studies on competitive influence diffusion have emerged recently. [4], [16], [17], [5], [6] mainly studied two-influence model. Further, [10], [18], [19], [7], [20], brought forward some general competitive influence propagation model. Bharathi et al. [10] proposed a new IC based model for the diffusion of multiple innovations, studied the 2 player game and proved that the last mover can obtain $(1-1/e)$ of the best response. Kostka [18] classified the node into k states, their basic model is a special case of IC model. They studied a 2 player case and showed that the first mover does not always have the advantage. Pathak [19] presented an extension of LT model for multiple cascades allowing nodes to switch between them. Three LT based competitive models were proposed by Borodin [7], which include the Weight-Proportional Competitive Linear Threshold Model, the Separated-Threshold Model and the Competitive Threshold Model with Forcing. The authors also showed that influence maximization under none of these models is submodular. However, so far, none models have considered the importance of timeliness in influence competition.

Long et al. in [11] proposed a new problem, called J-MIN-Seed, whose objective is to find out the minimum number of seeds to make at least J users being influenced. This is the first effort about information spread guarantee in IC and LT model from a minimization perspective. In [21], Chen studied seed minimization problem under their fixed threshold model, which is a deterministic model where a node becomes activated if the number of its active neighbors exceeds a fixed threshod. Goldberg and Liu in [22] considered another variant of fixed threshold model, and used linear programming to provide an approximation algorithm. Note that the analysis and algorithms in [21] and [22] can only be employed to their deterministic and nonsubmodular models.

Goyal et al. in [23] proposed a bicriteria approximation algorithm to minimize the size of the seed set such that its expected influence coverage can achieve a given threshold. In [12] Zhang et al. considered the Probabilistic Coverage Guarantee problem, whose task is to select minimum number of seeds so that the number of nodes being impacted reaches a given threshold with a guaranteed probability. They showed that greedy approximates the optimal solution with a multiplicative ratio and an additive error. The results provided in [23] and [12] both use additive errors, which are not small, and our result in this paper bests theirs significantly.

III. IC model and monotone submodular set function

According to the definition in [3], an IC influence graph is a weighted graph $G=(V,E,p)$, where V is a set of nodes whose size is n, $E\subseteq V\times V$ is a set of edges whose size is m, and $p:E\rightarrow[0,1]$ is a function that p_{uv} represents the success probability when node u tries to activate v. The information propagation in IC model starts with a given set S of active nodes. In every step i, the set of active nodes is S_i (let $S_0=S$ and $S_{-1}=\emptyset$), which consists of the nodes in S_{i-1} and the nodes activated by the nodes in $S_{i-1}\setminus S_{i-2}$ in step i. Note that each node u only has one chance to activate each of its neighbors v with probability p_{uv} when u first becomes active, and p_{uv} is a history independent parameter. The process runs until when $S_{t+1} = S_t$ at a step $t+1$.

Information spread of a given seed set is the expected number of nodes activated by this set. In IC cascade process, fixed $\{p_{uv}\}$ does not fasten the final influence, since each activation from u to v is an event successes with probability p_{uv}. The influence is the expectation over the combination of all the events that a node u activates one of its neighbor v.

A set function f on a given set V is a function from 2^V to $\mathbb{R}$, where 2^V denotes the power set of V and $\mathbb{R}$ is the set of real numbers. We introduce following two basic properties.

- **Monotonicity**: f is *monotone increasing* if $f(X) \leq f(Y)$ for any two sets $X \subseteq Y$.
- **Submodularity**: f is *submodular* if for every $X \subseteq Y \subseteq V$ and $z \in V \setminus Y$,

$$f(X \cup \{z\}) - f(X) \geq f(Y \cup \{z\}) - f(Y).$$

IV. C-IC Model and Seed Minimization Problem

We introduce our C-IC (Competitive - Independent Cascade) model here. This is the first competitive influence propagation model that allows overlapping seeds. The social network is represented as a directed graph $G = (V, E)$, and each edge is associated with a positive weight. For any two nodes u and v, if there is a directed edge from u to v, then we call u as v's *in-neighbor* and v as u's *out-neighbor*.

In C-IC model, the seeds are *active* nodes that initially propagate the influences. For different influences, we don't require their seeds being disjoint. In other words, since overlapped seeds are allowed, a single seed may serve multiple influences.

Each non-seed node, on the other hand, has three possible states: *inactive*, *active* and *decided*. If a node receives no information from any existing influence, then we call the node *inactive*. When a node becomes exposed to some influence, this node becomes *active*. A *decided* node is the one that is determined to one influence.

To be more specific, suppose there are k different influences $\mathcal{I} = \{I_1, ..., I_k\}$ diffusing in the social network, and there are m seeds $s_1, ..., s_m$. Since each seed s_i may propagate multiple influences, we associate s_i with a set $\mathcal{I}_i$ of all the influences that use s_i as a seed ($\mathcal{I}_i \subseteq \mathcal{I}$). In time 0, all non-seeds are inactive and each seed s_i starts to propagate the influences to its non-seed out-neighbors through its out-edges. The propagation rule is as follows: denote as $p_{s_i v}$ the weight of edge (s_i, v), then the chance that v successful accepts the influences from s_i is $p_{s_i v}$. At time 1, every v accepting some influences from the seeds becomes active, and tries to propagate all influences it received to its out-neighbors. After propagating influences, v becomes decided to one of the influences it received with equal probability, still at time 1. Generally speaking, at time $t+1$, node v firstly accepting any influence in time t becomes active and propagates all influences it accepted to its neighbors. Denote as $\mathcal{I}'_v$ the set of all influences v has accepted, then still at time $t+1$, v becomes decided to any influence $I \in \mathcal{I}'_v$ with equal probability $1/|c_v|$ where $|c_v|$ is the cardinality of $\mathcal{I}'_v$ or the number of influences. Note that all nodes propagate the influences once, at and only at the time they turn active. The influence diffusion ends when no more nodes becomes active. We use σ_i to denote the expected number of nodes decided to influence I_i at last, and call σ_i the spread of I_i. Note that σ_i is the expectation but not a certain number, and σ_i is a function w.r.t. a seed set S, i.e., $\sigma_i(S)$.

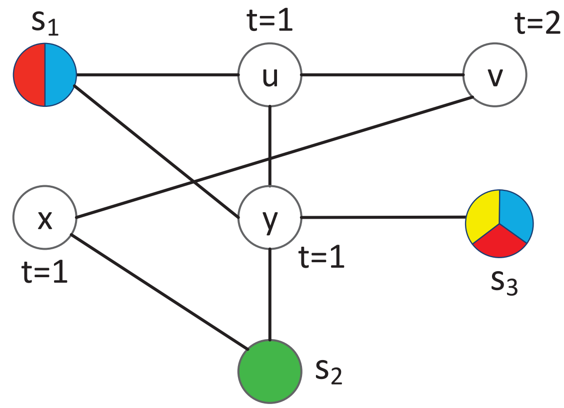


Figure 1: An example of influence propagation under C-IC model

Fig. 1 plots an sample social networks. Assume that all edges are bidirectional and with weight 1. suppose there are 4 influences $\{I_1, I_2, I_3, I_4\}$ representing by red, blue, yellow, green respectively, and they choose $\{s_1, s_3\}$, $\{s_1, s_3\}$, $\{s_3\}$, $\{s_2\}$ as the seed sets respectively. As what has been shown, a node can be the seed for multiple influences. At time 0, the seeds start to diffuse the influences. Each seed diffuses all influences that uses it as a seed (i.e., the set $\mathcal{I}_i$) to its out-neighbors. Since all edge weight is 1, $\{y, u\}$ successfully accept I_1 and I_2 from s_1, $\{y, u\}$ successfully accept I_4 from s_2, and $\{y\}$ successfully accepts I_1, I_2 and I_3 from s_3. Then at the beginning of time 1, nodes x, y, and u become active. After becoming active, a node will turn into decided at the same time step. x is exposed to $\mathcal{I}'_x = \{I_4\}$ from s_2 only, then it is decided to I_4 with probability 1; y is exposed to $\mathcal{I}'_y = \{I_1, I_2, I_3, I_4\}$ from s_1, s_2 and s_3, then it is decided to any influence in $\mathcal{I}'_y$ with probability $1/4$; u is exposed to $\mathcal{I}'_v = \{I_1, I_2\}$ from s_1, then it is decided to I_1 or I_2 with probability $1/2$ respectively. However, no matter which influence a newly active node turns decided to, this node always propagates all influences it accepted to its out-neighbors. Illustrating with the sample social network, at time 1 when x, y and u become active, they respectively diffuses the influences $\mathcal{I}'_x$, $\mathcal{I}'_y$ and $\mathcal{I}'_u$ immediately, and v accepts $\mathcal{I}'_x$ from x, and $\mathcal{I}'_u$ from u, which means $\mathcal{I}'_v = \{I_1, I_2, I_4\}$. At time 2, v becomes active first, and then turns decided, to any influence in c_v, with probability $1/3$. There is no more potential active nodes, therefore the influence diffusion stops and we check the spreads σ_i. For I_1, s_1 and s_3 are the seeds, u is decided to I_1 with probability

$1/2$, v is decided to I_1 with probability $1/3$, y is decided to I_1 with probability $1/4$, therefore $\sigma_1(\{s_1, s_3\}) = 1 + 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} = 3\frac{1}{12}$. Similarly, $\sigma_2(\{s_1, s_3\}) = 3\frac{1}{12}$, $\sigma_3(\{s_3\}) = 1\frac{1}{4}$, $\sigma_4(\{s_2\}) = 2\frac{7}{12}$.

Given a social network $G = (V, E)$, the weights $\{p_{uv}\}$ on edges in E, a target set $U \subseteq V$, a threshold $\mu \leq |U|$, k, the number of competitive influences, and $\{c(v)|v \in V\}$, the cost for making node v as a seed, we define following two variants of minimum cost seed set in a social network where multiple influences exist and compete with each other.

Definition 1 (Overall Minimum Cost Seed Set (*O-MinSeed*)). *To find out the seed set S^* with minimum total cost such that the overall spread $\sigma = \sum_{j=1}^{k} \sigma_j$ of all influences within the target nodes in U satisfies that $\sigma \geq \mu$.*

Definition 2 (Minimum Cost Seed Set (*MinSeed*)). *Given the seed sets $\{S_j | j \neq o, 1 \leq j \leq k\}$ for all other influences, MinSeed for influence I_o is to find out the minimum cost seed set S_o^* such that the spread σ_o of I_o in U satisfies that $\sigma_o \geq \mu$, which is,*

$$S_o^* = \underset{\sigma_o(S) \geq \mu}{\arg\min} \sum_{v \in S} c(v).$$

V. Information spread in C-IC model

A. Computation Complexity

Proposition 1. *For a social network, the overall spread $\sigma = \sum_{j=1}^{k} \sigma_j$ in C-IC model equals to σ_{IC}, where σ_{IC} is the single influence spread of seed set $\bigcup_{j=1}^{k} S_j$ in IC model.*

Theorem 1. *O-MinSeed under C-IC model is NP-hard.*

Theorem 2. *For any fixed $k \geq 1$, MinSeed under C-IC model is NP-hard.*

B. Monotony and submodularity

In this section, we analyze the overall spread and the spread for specific influence. First, according to Proposition 1, and from the proof in [3]:

Note: The overall spread $\sigma = \sum_{j=1}^{k} \sigma_j$ in C-IC model is monotone increasing and submodular w.r.t. the overall seed set.

Theorem 3. *Given the seed sets $\{S_j | j \neq o, 1 \leq j \leq k\}$ for all other influences, the spread σ_o of influence I_o is monotone increasing and submodular w.r.t. the seed set S_o of I_o.*

Proof: Recall that in C-IC model, each node u has a single chance to activate its out-neighbor v at the moment u turns to active. If this attempt succeeds, all influences u accepted will be propagated to v, else v will accept no information from u. So influence propagation in C-IC model may be analyzed based on a set of random graphs: For each edge $(u, v) \in E$, we flip a coin in time 0 of bias p_{uv}. If the coin flip indicates an successful activation then we declare (u, v) a live-edge. Consider the probability space in which each sample point specifies one possible set of outcomes for all the coin flips on the edges, and denote $\mathcal{X}$ one sample point. Our next proof will be based on $\mathcal{X}$, which is a determined graph. Denote by $\sigma_o^{\mathcal{X}}(S_o)$ the spread of I_i on graph $\mathcal{X}$, and we omit $\mathcal{X}$ in $\sigma_o^{\mathcal{X}}(S_o)$ if the context is clear.

We first prove that $\sigma_o(S_o)$ is monotone increasing on graph $\mathcal{X}$. Suppose we are adding s' to S_o. For a node v, denote by $\sigma_o(S_o)[v]$ the potential spread I_o can obtain from v using S_o as the seed set.

If $v \neq s'$, v becomes decided as long as it accepts any successful influence. In other words, if v has the possibility to choose I_o, it must be exposed to I_i the moment it turns active. $\sigma_o(S_o)[v] \geq 0$ and $\sigma_o(S_o)[v] > 0$ iff the length of the shortest path from S_o to v equals to that of the shortest path from $S \setminus S_o$ to v. In following discuss on v, we omit $[v]$ in $\sigma_o(S_o)[v]$ for the sake of simplicity. Adding a new seed s' into S_o may bring following consequences: 1) The length of the shortest path from S_o to v stays the same. In this case, if $\sigma_o(S_o) = 0$ which means the shortest path from S to v are from $S \setminus S_o$, even if adding s' to S_o, the shortest path from S to v are still from $S \setminus S_o$, $\sigma_o(S_o \cup \{s'\}) = 0$. Otherwise if $\sigma_o(S_o) > 0$ which means there is at least one shortest path from S to v coming from S_o, and adding s' to S_o does not change this fact and $\sigma_o(S_o)$ stays the same. 2) The length of the shortest path from S_o to v decreases. In this case, if $\sigma_o(S_o) = 0$ which means the shortest path from S to v are from $S \setminus S_o$, by adding s' to S_o, the shortest path from S to v may come from S_o, that brings $\sigma_o(S_o \cup \{s'\}) \geq 0$. Otherwise if $\sigma_o(S_o) > 0$, $\sigma_o(S_o \cup \{s'\})$ may equal to 1 but it at least is $\sigma_o(S_o)$. Hence, $\sigma_o(S_o)$ is monotone increasing w.r.t. S_o.

Else if $v = s'$, since $\sigma_o(S_o \cup \{s'\})[s'] = 1$, $\sigma_o(s')$ will not decrease after adding after s' to S_o. Therefore, $\forall v \in V$, $\sigma_o(S_o)[v] \leq \sigma_o(S_o \cup \{s'\})[v]$, and since $\sigma_o(S_o) = \sum_{v \in V} \sigma_o(S_o)[v]$, σ_o is monotone increasing w.r.t. to S_o.

We then prove that σ_o is submodular w.r.t. S_o on graph $\mathcal{X}$. In other words, for every $X \subseteq Y \subseteq V$ and $z \in V \setminus Y$, $\sigma_o(X \cup \{z\}) - \sigma_o(X) \geq \sigma_o(Y \cup \{z\}) - \sigma_o(Y)$. Still, we consider the spread on a single node v, and omit $[v]$ in $\sigma_o(S_o)[v]$ for simplicity.

If $v \neq z$, then we consider three shortest paths in $\mathcal{X}$: i) P_z, the shortest path between z and v; ii) P_X, the shortest path from X to v; iii) P_Y, the shortest path from Y to z. Besides above three paths, we consider $\Gamma_{S_{-o}}$, the set of all shortest paths from $\cup_{j \neq o, 1 \leq j \leq k} S_j$ to v, and denote by l the length of these shortest paths. Suppose that the source seeds of paths in $\Gamma_{S_{-o}}$ come from d different influences.

Use $|\cdot|$ to denote path length. Obviously $|P_X| \geq |P_Y|$, therefore we must have $|P_z| \geq |P_X| \geq |P_Y|$, $|P_X| \geq |P_z| \geq |P_Y|$, or $|P_X| \geq |P_Y| \geq |P_z|$.

1) $|P_z| \geq |P_X| \geq |P_Y|$. In this case, adding z into X or Y will not change the spread of I_i on v. $\sigma_o(X \cup \{z\}) - \sigma_o(X) = \sigma_o(Y \cup \{z\}) - \sigma_o(Y) = 0$.
2) $|P_X| \geq |P_z| \geq |P_Y|$. In this case, adding z into Y will not change the spread of I_i on v, $\sigma_o(Y \cup \{z\}) - \sigma_o(Y) = 0$. When adding z into Y, if $l \geq |P_X| \geq |P_z|$ or $|P_X| \geq |P_z| > l$, then $\sigma_o(X \cup \{z\}) - \sigma_o(X) = 0$, else $|P_X| > |P_z|$ and $l \in [|P_z|, |P_X|]$: 1) $l = |P_z|$, $\sigma_o(X \cup \{z\}) = 1/(d+1)$; 2) $l > |P_z|$, $\sigma_o(X \cup \{z\}) = 1$. Hence $\sigma_o(X \cup \{z\}) - \sigma_o(X) > 0$. Therefore $\sigma_o(X \cup$

$\{z\}) - \sigma_o(X) \geq 0 = \sigma_o(Y \cup \{z\}) - \sigma_o(Y)$.

3) $|P_X| \geq |P_Y| \geq |P_z|$. In fact $|P_X| = |P_Y| = |P_z|$ is contained in previous cases, hence we check the three subcases: i) $|P_X| > |P_Y| > |P_z|$, ii) $|P_X| = |P_Y| > |P_z|$, and iii) $|P_X| > |P_Y| = |P_z|$. For i), if $l > |P_X|$, then $\sigma_o(X) = 1$, $\sigma_o(X \cup \{z\}) = 1$, $\sigma_o(Y) = 1$, $\sigma_o(Y \cup \{z\}) = 1$. Else if if $l = |P_X|$, then $\sigma_o(X) = 1/(d+1)$, $\sigma_o(X \cup \{z\}) = 1$, $\sigma_o(Y) = 1$, $\sigma_o(Y \cup \{z\}) = 1$. Else if $|P_X| > l > |P_Y|$, then $\sigma_o(X) = 0$, $\sigma_o(X \cup \{z\}) = 1$, $\sigma_o(Y) = 1$, $\sigma_o(Y \cup \{z\}) = 1$. Else if $|P_X| > l = |P_Y|$, then $\sigma_o(X) = 0$, $\sigma_o(X \cup \{z\}) = 1$, $\sigma_o(Y) = 1/(d+1)$, $\sigma_o(Y \cup \{z\}) = 1$. Else if $|P_Y| > l > |P_z|$, then $\sigma_o(X) = 0$, $\sigma_o(X \cup \{z\}) = 1$, $\sigma_o(Y) = 0$, $\sigma_o(Y \cup \{z\}) = 1$. Else if $|P_Y| > l = |P_z|$, then $\sigma_o(X) = 0$, $\sigma_o(X \cup \{z\}) = 1/(d+1)$, $\sigma_o(Y) = 0$, $\sigma_o(Y \cup \{z\}) = 1/(d+1)$. Else, $|P_z| > l$, then $\sigma_o(X) = 0$, $\sigma_o(X \cup \{z\}) = 0$, $\sigma_o(Y) = 0$, $\sigma_o(Y \cup \{z\}) = 0$. No matter what, in i) $\sigma_o(X \cup \{z\}) - \sigma_o(X) \geq \sigma_o(Y \cup \{z\}) - \sigma_o(Y)$ always stands, and we can prove that this is also true for ii) and iii) similarly.

If $v = z$, then $\sigma_o(X \cup \{z\}) = \sigma_o(Y \cup \{z\}) = 1$, hence $\sigma_o(X \cup \{z\}) - \sigma_o(X) \geq \sigma_o(Y \cup \{z\}) - \sigma_o(Y)$ since $\sigma_o(X) \leq \sigma_o(Y)$ (σ_o is monotone increasing).

Therefore, $\forall v \in U$, $\sigma_o(S_o)[v]$ is submodular, and then $\sigma_o(S_o) = \sum_{v \in U} \sigma_o(S_o)[v]$ is submodular.

We have proved that $\sigma_o^{\mathcal{X}}(S_o)$ basing on sample graph $\mathcal{X}$ is monotone increasing and submodular. Denote by Prob$[\mathcal{X}]$ the probability of getting $\mathcal{X}$ in a "coin flip". Finally,

$$\sigma_o(S_o) = \sum_{\text{sample } \mathcal{X}} \text{Prob}[\mathcal{X}] \cdot \sigma_o^{\mathcal{X}}(S_o)$$

is monotone increasing and submodular w.r.t. S_o. Proof is complete. ■

From above theorem we have seen that spread σ_i under C-IC model has very nice properties. Therefore, its performance may be guaranteed using simple greedy algorithm.

VI. Algorithms and Performance Analysis

In this section, we introduce the algorithm for *MinSeed* and analyze the performance. At first, we propose the greedy algorithm. Note that in Alg. 1, f is a general submodular function.

Algorithm 1 Greedy

1: $Q \leftarrow \emptyset$;
2: **while** there exists a node $v \in V$ satisfying that $f(Q \cup \{v\}) - f(Q) > 0$ **do**
3: $\quad v^* \leftarrow \arg\max_{v \in V \setminus Q} \{\frac{f(Q \cup \{v\}) - f(Q)}{c(v)}\}$;
4: $\quad Q \leftarrow Q \cup \{v^*\}$;
5: **end while**
6: **return** Q.

Theorem 4. *If f is a monotone increasing and submodular function, suppose that the solution Q produced by Alg. 1 satisfies the condition $\frac{f(Q^{a-1} \cup \{v^a\}) - f(Q^{a-1})}{c(v^a)} \geq \epsilon$ for all $a = 1, 2, ..., h$, where h is the total number of rounds executed by Alg. 1, then*

$$c(Q) \leq \left(1 - \ln \epsilon + \ln \frac{f(Q^*)}{c(Q^*)}\right) \cdot c(Q^*).$$

Note that in [24], Du et al. discussed the case where $\epsilon = 1$, and our result is the generalized version.

For a *MinSeed* instance, we let f in Alg. 1 be $\min\{\sigma_o(S_o), \mu\}$. Since we have proved that σ_o is monotone increasing and submodular, it is easy to obtain that f is also monotone increasing and submodular. Therefore we can employ Theorem 4 to obtain the performance ratio of Alg. 1 on *MinSeed*.

Note that Theorem 4 only asks f being monotone and submodular, and it also applied to the greedy algorithms proposed in many previous work. Note that our result significantly bests the previous one given in [12]: 1) Our result give a strict approximation ratio between the algorithm's result and the optimal result, while the result in [12] has a huge addictive error which impact the accuracy. 2) Even without the addictive errors, the ratio in [12] is much greater than ours. Hence, our theoretical analysis gives the state-of-art best ratio.

Another key issue left for Alg. 1 is the complexity of obtaining $\sigma_o(S_o)$. Unfortunately, computation of $\sigma_o(S_o)$ is hard.

Proposition 2. *The computation of $\sigma_o(S_o)$ in C-IC model is $\#P$-hard.*

Above proposition is not difficult to see since the special case where only one influence exists is the traditional IC model, and Chen et al. has proved that computing influence spread in IC is $\#P$-hard in [9]. Therefore, to obtain $\sigma_o(S_o)$, we may use Monte Carlo method to randomly generate samples and make estimation using these samples.

A. To reduce the time complexity of making greedy choice

However, usually Monte Carlo method is time consuming. In IC model, it takes up to hours to compute the spread of 50 seeds in a social network containing $400K$ edges [9], not mentioning C-IC model which is more complex.

Let us analyze the time complexity for Alg. 1. Denote $\xi(x, n, m)$ the time complexity for computing the spread σ_o of x seeds in a social graph containing n nodes and m edges. Therefore, if at last we choose n_s nodes such that $\sigma_o \geq \mu$, then the overall time of Alg. 1 is

$$\sum_{i=1}^{n_s} (n + 1 - i) \cdot \xi(i, n, m). \tag{1}$$

$\xi(i, n, m)$ is not polynomial since we know the computation of spread is $\#P$-hard. Even when using Monte Carlo method, since usually more than $10,000$ rounds have to be simulated, $\xi(i, n, m)$ is still time consuming. Considering the fact that when the graph is big, n may be up to $100,000$, then $\xi(i, n, m) \cdot (n + 1 - i)$, $i = 1, 2, ...$ is huge, which is the time complexity for making greedy choice, i.e., step 3 in round i

of Alg. 1). Therefore, we see that even simple greedy choice may become infeasible on huge social networks.

To design scalable algorithm for *minSeeds* problem under C-IC model, we consider two points: 1) A scalable method that estimates σ_o. This is to reduce time complexity of ξ in expression (1). 2) A reasonable relatively small subset V_p of V to pick future seeds from. This is to reduce the size of n in expression (1).

In fact the first point is hard to address. To see that, consider a simple graph: star graph with $\tau+1$ nodes $\{u_0, u_1, ..., u_\tau\}$ and u_0 in the center. Assume that for each of nodes outside, there is an edge e_i whose weight is w_i from this node to the center node u_0. Each outside node is the only seed for its influence. Let $U' = \{u_i | 1 \leq i \leq n\}$ be the set of outside nodes, and denote $\mathcal{U}'$ as the family of subsets of U', then for any arbitrary $i \in [1, \tau]$, the exact spread of I_i on node u_0 is computed as:

$$\sum_{U \in \mathcal{U}',\ U \text{ contains } u_i} \frac{1}{|U|} \prod_{u_j \in U} w_j \prod_{u_k \in U' \setminus U} (1 - w_k). \tag{2}$$

The time complexity of computing (2) is $\Omega(2^\tau)$, which is considerable. Note that it is only for the simple star graph, which ordinary social networks are much more complex than. From this example, we see that, unlike IC model, it is still troublesome to compute the exact influence spread on graphs with simple topology, therefore similar techniques like *Maximum Influence Out-Arborescence* in [9] become infeasible for scalably computing information spread. To solve this issue, we adopt the idea of *Linear-Combination Single-Hop Spread (LSS)*. In LSS, only the spread in one hop is considered, but (2) tells us that even single hop spread is hard to compute, therefore, LSS further uses linear combination of edge weights to estimate the spread. In detail, for any non-seed v, let S'_o be the set of v's in-neighbors that are seeds for influence I_o, then the spread on v is estimated as

$$\left[1 - \prod_{s \in S'_o} (1 - p_{sv})\right] \cdot \left(\sum_{s \in S'_o} \frac{p_{sv}}{\sum_{s \in S'_o} p_{sv}} \cdot \frac{1}{|\mathcal{I}_s|}\right) \tag{3}$$

In (3), $1 - \prod_{s \in S'_o} (1 - p_{sv})$ is the chance that v is activated by the seeds within one hop, $\frac{1}{|\mathcal{I}_s|}$ is the spread s can bring to v if s successfully actives v through edge p_{sv}, and $\frac{p_{sv}}{\sum_{s \in S'_o} p_{sv}}$ is the "significance" of s to v. (3) is a reasonable spread estimation that reduces the computation cost of ξ in (1) from exponential time to linear time.

Comparing to reducing the cost of ξ, reducing the size of n in (1) is easier: since many nodes have very few out-edges so that they are impossible for being seeds, we choose a pool of nodes that have the "potential" to become seeds. To be specific, we pick V_p, a group of nodes that have the highest out-degrees. The size of this group is big enough for choosing the right seed, but is much smaller than the number of nodes in original graph. The idea comes from the fact that in reality the seeds are always much much fewer than the original nodes.

LSS uses the original social graph. In other words, Monte Carlo method is not used in LSS. Monte Carlo method is only used to judge if enough seeds have been picked such that the spread has reached the threshold μ. The details are presented in the pseudocode of Alg. 3.

Algorithm 2 LSS(S_o) on original graph G

$\delta \leftarrow 0$, $S'_o \leftarrow \emptyset$;
for $\forall v \notin S_o$ **do**
  $S'_o \leftarrow$ the set of v's in-neighbors in S_o;
  if $S'_o \neq \emptyset$ **then**
    $\delta \leftarrow \delta + [1 - \prod_{s \in S'_o} (1 - p_{sv})] \cdot (\sum_{s \in S'_o} \frac{p_{sv}}{\sum_{s \in S'_o} p_{sv}} \cdot \frac{1}{|\mathcal{I}_s|})$;
  end if
end for
$\delta \leftarrow \delta + |S_o|$;
return δ.

Algorithm 3 LSS-Greedy

$Q \leftarrow \emptyset$;
while there exists a node $v \in V_p$ satisfying that $f(Q \cup \{v\}) - f(Q) > 0$ **do**
  $v^* \leftarrow \arg\max_{v \in V_p \setminus Q} \{\frac{LSS(Q \cup \{v\}) - LSS(Q)}{c(v)}\}$;
  $Q \leftarrow Q \cup \{v^*\}$;
end while
return Q.

VII. Experiments

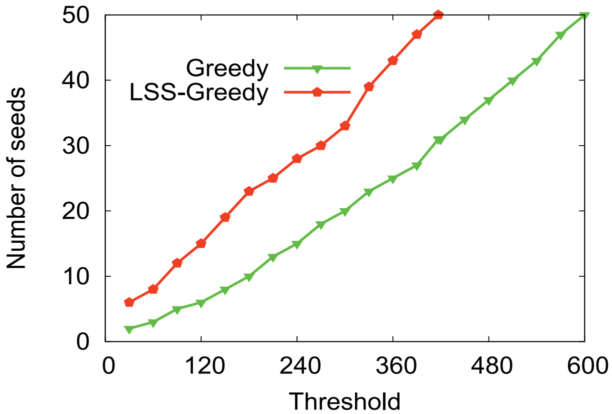
To check the competitive influence spread in C-IC model and the performance of LSS-Greedy algorithm, we have carried through plenty of experiments over real-world datasets.

Table I. Dataset Statistics

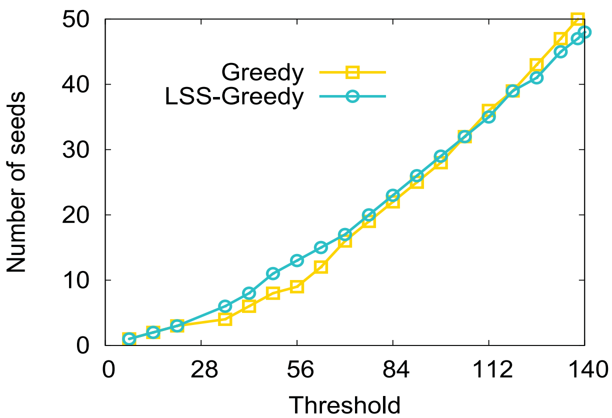
	GR-QC	WikiVote	Enron	Epinion
#Nodes	5.2K	7.1K	36.7K	75.9K
#Edges	14.5K	103.7K	367.7K	508.8K
Avg. OD	5.5	14.6	10.1	6.7
Max OD	81	893	1383	1801

Table I (in which 'OD' means 'outdegree') presents the statistics of our employed datasets, (Arxiv) GR-QC (General Relativity and Quantum Cosmology) [25] collaboration network is from the e-print arXiv and covers scientific collaborations between authors in these areas. WikiVote[26] contains all the Wikipedia administrator voting data from the inception of Wikipedia till January 2008. Enron[27] is a large scale email collection, in which the nodes are email addresses and an edge between u and v means they have exchanged emails. Epinions[28] is a network indicates the trust relationships, which is collected from the Epinions.com web site.

Generating propagation probabilities. We use two traditional models generate these nonuniform propagation probabilities.

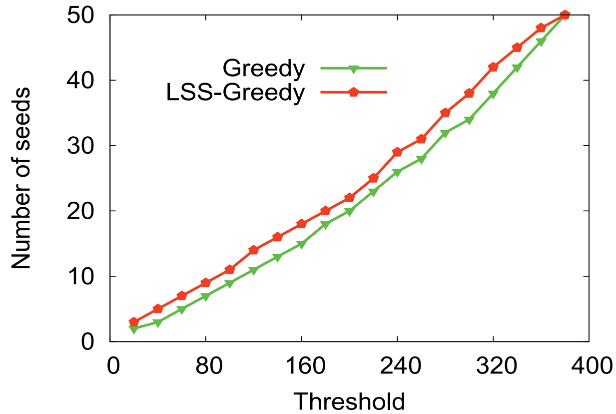


(a) WC probability model

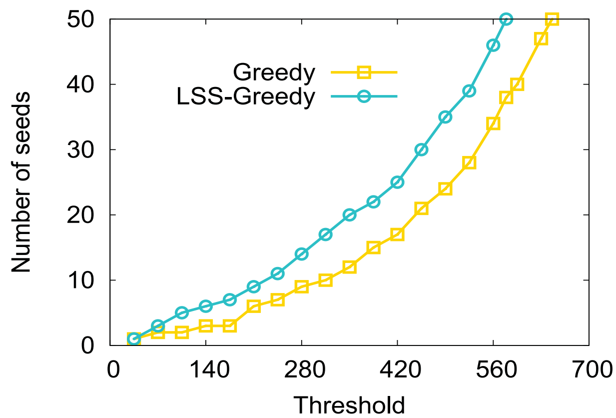


(b) TRIVALENCY probability model

Figure 2: (Arxiv) GR-QC



(a) WC probability model



(b) TRIVALENCY probability model

Figure 3: WikiVote

- WC model: The weighted cascade model proposed in [3]. In WC model, p_{uv} of edge (u, v) equals to $1/d(v)$, where $d(v)$ is the in-degree of v. This model always generates asymmetric and nonuniform propagation probabilities even if the original graph is undirected.
- TRIVALENCY model: This model uniformly at random selects a probability from the set $\{0.1, 0.01, 0.001\}$ (corresponding to high, medium, and low influences) for each edge (u, v).

Seed Cost and Target Set U. In all simulations, we uniformly set the cost of choosing a seed as 1, i.e., $\forall v \in V, c(v) = 1$, and the target set U is set as the original node set V of graph G.

Competitor's seed selection. In all simulations, we set two other competitors and give each of them 10 seeds. The competitors' seeds are chosen as follows: 1) Pick 5 nodes with the maximum out-degree and make the 5 nodes as the seeds for both competitors. 2) In remaining nodes, pick 10 nodes whose out-degrees are maximum, and assign them to the two competitors by turns.

For all cases where Monte Carlo simulation is needed, we run 1000 times and get the average. We compare the costs of seeds output by simple Greedy and LSS-Greedy on Axiv GR-QC and WikiVote in Fig. 2 and Fig. 3 respectively. It can be seen that on GR-QC network, LSS-Greedy works approximately 30% worst than Greedy when the probability model is WC, and LSS-Greedy works as good as Greedy on TRIVALENCY model. On WikiVote network, LSS-Greedy works as good as Greedy on WC model and 15% worst than Greedy on TRIVALENCY model. Comparing to Greedy, LSS-Greedy provides competitively small seed set in a very quick running time. For Enron and Epinion, their sizes are too big to make Greedy feasible, therefore we only check the performance of LSS-Greedy. Fig. 4 shows the costs of seeds on Enron and Epinion under WC model, and Fig. 5 shows the costs of seeds on Enron and Epinion under TRIVALENCY model.

We uniformly check the time taken for choosing the first seed in LSS-Greedy and Greedy on all datasets, and the result is plotted in Fig. II. LSS-Greedy runs much faster than Greedy.

Table II. Running time (minutes)

Alg.	GR-QC		WikiVote		Enron		Epinion	
	WC	TRI	WC	TRI	WC	TRI	WC	TRI
LSS	0.08	0.08	0.32	0.7	2.4	13.8	4.6	14.1
Greedy	6.7	5.9	12.4	27.1	N/A	N/A	N/A	N/A

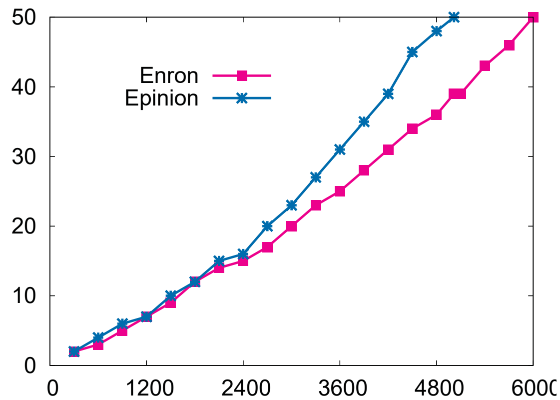


Figure 4: Seed set size in WC probability model

VIII. Conclusion

In this paper we studied how an influence may use the seeds with minimum cost to spread its information to a certain threshold under competitors' hinders. We noticed the fact that: the information arriving first always impact people significantly, and presented C-IC model to characterize the competitive information propagation when different influences in a social network contend each other. C-IC model allows overlapping seeds. We proved that a influence's spread is monotone and submodular if other influences' seeds are fixed, and proposed *MinSeed* and gave a greedy algorithm for it. The ratio of simple greedy algorithm has been given, which significantly bests existing results. Further, we noticed that simple greedy is still too time consuming, therefore designed an effective method for estimating information spread in C-IC model. Based on this method we devised scalable algorithm that applies for large social networks. We confirmed that our algorithm is scalable and outputs seed set with small total cost on real-world networks.

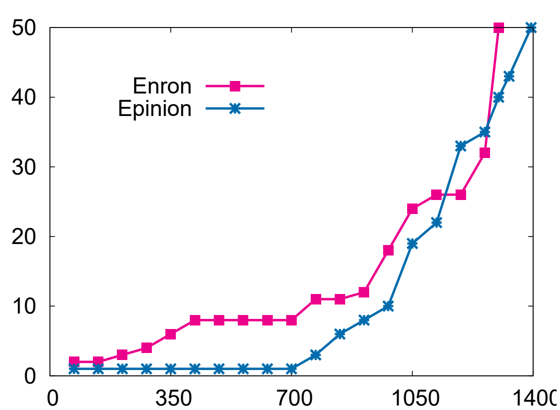


Figure 5: Seed set size in TRIVALENCY probability model

References

[1] P. Domingos and M. Richardson. Mining the network value of customers. KDD '01, pages 57–66. ACM.

[2] J. Goldenberg, B. Libai, and E. Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 2001.

[3] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. KDD '03, pages 137–146. ACM.

[4] T. Carnes, C. Nagarajan, S. M. Wild, and A. Van Zuylen. Maximizing influence in a competitive social network: A follower's perspective. ICEC '07, 2007.

[5] C. Budak, D. Agrawal, and A. El Abbadi. Limiting the spread of misinformation in social networks. WWW '11.

[6] S. Li, Y. Zhu, D. Li, D. Kim, and H. Huang. Rumor restriction in online social networks. IPCCC '13.

[7] A. Borodin, Y. Filmus, and J. Oren. Threshold models for competitive influence in social networks. WINE '10.

[8] W. Lu, F. Bonchi, A. Goya, and L.V.S. Laksmanan. The bang for the buck: Fair competitive viral marketing from the host perspective. KDD '13.

[9] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. KDD '10, pages 1029–1038. ACM.

[10] S. Bharathi, D. Kempe, and M. Salek. Competitive influence maximization in social networks. WINE '07.

[11] C. Long and R. C. Wong. Minimizing seed set for viral marketing. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining*, ICDM '11, pages 427–436, 2011.

[12] P. Zhang, W. Chen, X. Sun, Y. Wang, and J. Zhang. Minimizing seed set selection with probabilistic coverage guarantee in a social network. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (KDD'14)*, pages 1306–1315, 2014.

[13] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance. Cost-effective outbreak detection in networks. KDD '07.

[14] W. Chen, Y. Yuan, and L. Zhang. Scalable influence maximization in social networks under the linear threshold model. ICDM '10.

[15] Y. Zhu, W. Wu, Y. Bi, L. Wu, Y. Jiang, and W. Xu. Better approximation algorithms for influence maximization in online social networks. *Journal of Combinatorial Optimization*, published online.

[16] W. Chen, A. Collins, R. Cummings, T. Ke, Z. Liu, D. Rincon, X. Sun, Y. Wang, W. Wei, and Y. Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. SDM '10, pages 379–390.

[17] D. Trpevski, W. K. S. Tang, and L. Kocarev. Model for rumor spreading over networks. volume 81, page 056102, May 2010.

[18] J. Kostka, Y. A. Oswald, and R. Wattenhofer. Word of mouth: Rumor dissemination in social networks. *SIROCCO, Lecture Notes in Computer Science*, pages 185–196, 2008.

[19] N. Pathak, A. Banerjee, and J. Srivastava. A generalized linear threshold model for multiple cascades. ICDM '10.

[20] Y. Zhu, D. Li, H. Guo, and R. Pamula. New competitive influence propagation models in social networks. In *10th IEEE International Conference on Mobile Ad-hoc and Sensor Networks (MSN'14)*, pages 257–262.

[21] N. Chen. On the approximability of influence in social networks. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '08, pages 1029–1037, 2008.

[22] S. Goldberg and Z. Liu. The diffusion of networking technologies. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '13, pages 1577–1594, 2013.

[23] A. Goyal, F. Bonchi, L. V.S. Lakshmanan, and S. Venkatasubramanian. On minimizing budget and time in influence propagation over social networks. *Social Network Analysis and Mining*, 3(2):179–192, 2013.

[24] D. Du, K. Ko, and X. Hu. Design and analysis of approximation algorithms. In P. Pardalos, editor, *Springer Optimization and Its Applications*. Springer, 2012.

[25] J. Kleinberg J. Leskovec and C. Faloutsos. Graph evolution: Densification and shrinking diameters.

[26] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 641–650.

[27] J. Leskovec, A. Dasgupta K. Lang, and M. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, pages 29–123, 6(1) 2009.

[28] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. ISWC'03.

APPENDIX

Assume that f is a submodular function on subsets of V, define

$$\Delta_Z f(X) = f(X \cup Z) - f(X)$$

for any subsets X and Z of V. When $Z = \{v\}$ is a singleton, we simple use $\Delta_v f(X)$ instead of $\Delta_{\{v\}} f(X)$.

Lemma 1. *For all sets $X, Z \in V$,*

$$\Delta_Z f(X) \leq \sum_{v \in Z} \Delta_v f(X).$$

Use above lemma, now we can prove Theorem 4:

Proof of Theorem 4: Let $q^i = f(Q^*) - f(Q^i)$ for $i = 0, 1, ..., h$. It is obvious that $q^0 = f(Q^*)$ and $\Delta_{v^i} f(Q^{i-1}) = f(Q^{i-1} \cup \{v^i\}) - f(Q^{i-1}) = q^{i-1} - q^i$.

Assume that $Q^* = \{x^1, x^2, ..., x^o\}$, then according to the greedy choice in Alg. 1 and Lemma 1, we have, $\forall j, 1 \leq j \leq h$,

$$\begin{aligned} \frac{q^{j-1} - q^j}{c(v^j)} &\geq \max_{1 \leq i \leq o} \frac{\Delta_{x^i} f(Q^{j-1})}{c(x^i)} \\ \geq & \quad \frac{\sum_{i=1}^{o} \Delta_{x^i} f(Q^{j-1})}{c(Q^*)} \geq \frac{\Delta_{Q^*} f(Q^{j-1})}{c(Q^*)} \\ = & \quad \frac{f(Q^* \cup Q^{j-1}) - f(Q^{j-1})}{c(Q^*)} \\ = & \quad \frac{f(Q^*) - f(Q^{j-1})}{c(Q^*)} = \frac{q^{j-1}}{c(Q^*)} \end{aligned} \tag{4}$$

Thus, we have

$$q^j \leq q^{j-1}(1 - \frac{c(v^j)}{c(Q^*)}) \quad \forall j, 1 \leq j \leq h. \tag{5}$$

Let us then check q^0,

$$\begin{aligned} q^0 &= f(Q^*) = f(Q) \\ = & \quad \sum_{i=1}^{h} \Delta_{v^i} f(Q^{i-1}) \\ \geq & \quad \epsilon \cdot \sum_{i=1}^{k} c(v^i) = \epsilon \cdot c(Q) \\ \geq & \quad \epsilon \cdot c(Q^*), \end{aligned}$$

and $q^h = f(Q^*) - f(Q) = 0$. What is more, $q^i \leq q^{i-1}$ for all $1 \leq i \leq h$, since f is monotone increasing. Therefore, we can find out an integer $r \in [0, h]$ such that $q^{r+1} < \epsilon \cdot c(Q^*) \leq q^r$. Note that from (5)

$$\frac{q^r - q^{r+1}}{c(v^{r+1})} \geq \frac{q^r}{c(Q^*)}. \tag{6}$$

We split the numerator of the left-hand side of inequality 6 into two parts: $q' = \epsilon \cdot c(Q^*) - q^{r+1}$, $q'' = q^r - \epsilon \cdot c(Q^*)$ ($q' + q'' = q^r - q^{r+1}$), and also split the denominator of the left-hand side into two parts proportionally ($c' + c'' = c(v^{r+1})$) such that c' and c'' satisfy following condition:

$$\frac{q'}{c'} = \frac{q''}{c''} = \frac{q^r - q^{r+1}}{c(v^{r+1})},$$

then, since $q'' = q^r - \epsilon \cdot c(Q^*)$:

$$\begin{aligned} \frac{q''}{c''} = \frac{q^r - \epsilon \cdot c(Q^*)}{c''} &\geq \frac{q^r}{c(Q^*)} \\ \iff \quad \epsilon \cdot c(Q^*) &\leq q^r(1 - \frac{c''}{c(Q^*)}). \end{aligned} \tag{7}$$

If we apply (5) repeatedly on the right-hand side of inequality (7), the following can be obtained:

$$\begin{aligned} \epsilon \cdot c(Q^*) &\leq q^0 \cdot \prod_{i=1}^{r} \left(1 - \frac{c(v^i)}{c(Q^*)}\right) \cdot \left(1 - \frac{c''}{c(Q^*)}\right) \\ &\leq q^0 \cdot \prod_{i=1}^{r} \exp(-\frac{c(v^i)}{c(Q^*)}) \cdot \exp(-\frac{c''}{c(Q^*)}) \qquad (*) \\ &= q^0 \cdot \exp(-\frac{c'' + \sum_{i=1}^{r} c(v^i)}{c(Q^*)}). \end{aligned} \tag{8}$$

Note that (*) stands because $1 + x \leq \mathrm{e}^x$. Then by (8),

$$c'' + \sum_{i=1}^{r} c(v^i) \leq c(Q^*) \cdot \ln \frac{q^0}{\epsilon \cdot c(Q^*)}. \tag{9}$$

We should consider v^i chosen after round $(r + 1)$, and

$$\begin{aligned} \sum_{i=r+2}^{h} c(v^i) &\leq \sum_{i=r+2}^{h} \frac{1}{\epsilon} \cdot \Delta_{v^i} f(Q^{i-1}) \\ &= \frac{1}{\epsilon} \cdot (f(Q) - f(Q^{r+1})) = \frac{1}{\epsilon} \cdot q^{r+1}. \end{aligned}$$

Finally,

$$\begin{aligned} c(Q) &= \sum_{i=1}^{h} c(v^i) \\ &= \sum_{i=1}^{r} c(v^i) + c(v^{r+1}) + \sum_{i=r+2}^{h} c(v^i) \\ &= \sum_{i=1}^{r} c(v^i) + c'' + c' + \sum_{i=r+2}^{h} c(v^i) \\ &\leq c(Q^*) \cdot \ln \frac{q^0}{\epsilon \cdot c(Q^*)} + \frac{1}{\epsilon} \cdot q' + \frac{1}{\epsilon} \cdot q^{r+1} \\ &= c(Q^*) \cdot \ln \frac{q^0}{\epsilon \cdot c(Q^*)} + c(Q^*) \\ &= \left(1 - \ln \epsilon + \ln \frac{f(Q^*)}{c(Q^*)}\right) \cdot c(Q^*), \end{aligned}$$

since $q'/c' = (q^r - q^{r+1})/c(v^{r+1}) = \Delta_{v^{r+1}} f(Q^r) \geq \epsilon$, and $q' = \epsilon \cdot c(Q^*) - q^{r+1}$. ■