BROKER-BASED AD ALLOCATION IN SOCIAL NETWORKS

A THESIS

SUBMITTED TO THE DEPARTMENT OF COMPUTER ENGINEERING

AND THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

İzzeddin Gür

August, 2013

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Hakan Ferhatosmanoğlu(Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Buğra Gedik(Co-advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Cevdet Aykanat

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Savaş Dayanık

Approved for the Graduate School of Engineering and Science:

Prof. Dr. Levent Onural
Director of the Graduate School

ABSTRACT

BROKER-BASED AD ALLOCATION IN SOCIAL NETWORKS

İzzeddin Gür
M.S. in Computer Engineering
Supervisors: Assoc. Prof. Dr. Hakan Ferhatosmanoğlu, Assist. Prof. Dr. Buğra Gedik
August, 2013

With the rapid growth of social networking services, there has been an explosion in the area of viral marketing research. The idea is to explore the marketing value of social networks with respect to increasing the adoption of a new innovation/product, or generating brand awareness. A common technique employed is to target a small set of users that will result in a large cascade of further adoptions. Existing formulations and solutions in the literature generally focus on the case of a single company. Yet, the problem gets more challenging if there are a number of companies (the *advertisers*), each one aiming to create a viral advertising campaign of its own by paying a set of network users (the *endorsers*). The endorsers are asked to post intriguing and entertaining ad messages that contain the content selected by the advertising company. The advertiser has a predefined budget on how much it is going to spend on this effort. Also each endorser has a limit on the number of companies for which it serves as an endorser. In this thesis, we design a *broker* system as an intermediary between advertisers and endorsers. We seek to maximize the spread of advertisements over regular users (the *audience*), while considering the budget constraints of advertisers. Our system avoids *overburdening* of the endorsers and *overloading* of the audience. We model the problem through a combinatorial optimization framework with budget constraints. We develop a cost-effective algorithm called CEAL, which is designed for solving the problem with close to optimal performance on large-scale graphs. We also revisit the traditional Independent Cascade Model (ICM) to account for overloaded users. We propose an extension of ICM called Independent Cascade Model with Overload (ICMO). We study the influence maximization problem on variations of this model. We perform experiments over multiple real-world social networks and empirically show that the proposed CEAL algorithm performs close to optimal in terms of coverage, yet is sufficiently lightweight to execute on large-scale graphs.

Keywords: Social Networks, Submodular Welfare Problem, Influence Maximization

Problem, Ad Allocation, Viral Marketing.

ÖZET

SOSYAL AĞLARDA ACENTA TABANLI REKLAM ATAMA

İzzeddin Gür
Bilgisayar Mühendisliği, Yüksek Lisans
Tez Yöneticileri: Assoc. Prof. Dr. Hakan Ferhatosmanoğlu, Assist. Prof. Dr. Buğra Gedik
Ağustos, 2013

Hızla gelişen sosyal ağ servisleri sayesinde, viral pazarlama alanında yapılan araştırmalarda bir patlama yaşandı. Sosyal ağlarda yeni bir fikrin, ürünün benimsenmesi veya marka bilinirliğinin geliştirilmesi için çokça uygulanan bir yöntem, küçük bir çekirdek kullanıcı kümesi seçerek, sonraki benimsenmelerin maksimize edilmesidir. Literatürdeki çözüm ve formülasyonlar genel olarak tek bir şirket durumunu göz önüne alır. Fakat, herbiri ağdaki bir kısım kullanıcılara (*reklam aktarıcı*) belli bir ücret ödeyerek bir viral pazarlama kampanyası oluşturmak isteyen birden fazla şirket (*reklam veren*) olduğu durumda problem daha zor bir hal almaktadır. Reklam aktarıcıların amacı ağ üzerinde reklam verenin seçtiği içeriği de barındıran ilgi çekici ve eğlenceli mesajlar göndermektir. Herbir reklam verenin önceden belirlenmiş belli bir bütçesi bulunmaktadır. Ayrıca, herbir reklam aktarıcının kaç tane reklam veren tarafından kullanılabileceğini sınırlayan bir limiti bulunmaktadır. Bu tezde, reklam verenler ile reklam aktarıcılar arasında bir *aracı* sistem tasarlamaktayız. Amacımız, reklam verenlerin bütçelerini muhafaza ederek, reklamların ağdaki sıradan kullanıcılar (*son kullanıcılar*) arasında yayılımını maksimize etmektir. Tasarladığımız sistem, reklam aktarıcılarının aşırı yüklenimine ve son kullanıcıların de reklamlarla boğulmasına engel olmaktadır. Bu problemi kombinatoryal optimizasyon problemi üzerinden bütçe kısıtlarını entegre ederek tasarlıyoruz. Bu problemin çözümü için büyük çaptaki ağlarda optimale yakın performanslı, maliyet-etkili bir algoritma tasarladık. Ayrıca son kullanıcıların aşırı-yüklenimini modellemek için klasik Bağımsız Yayılım Modelini (BYM) tekrar gözden geçirip, bu modele bir eklenti sunuyoruz: Aşırı-Yüklenim etkili Bağımsız Yayılım Modeli (ABYM). Yayılım maksimizasyon problemini bu model üzerinde çalışıyoruz. Birkaç gerçek büyük sosyal ağ verisi üzerindeki deneylerimizde gösteriyoruz ki sunulan algoritma optimal performansa yakın ve büyük sosyal ağlar üzerinde de çalıştırılabilecek kadar zaman açısından verimli.

Anahtar sözcükler: Sosyal Ağlar, Altmodüler Refah Problemi, Benimsenme Maksimizasyon Problemi, Reklam Atama, Viral Pazarlama.

Acknowledgement

I would like to first thank my advisor Hakan Ferhatosmanoğlu for his advise and support. He has given me the freedom to explore on my own while sharing his wisdom on how to take the next step. He has provided me a great vision on how to do science.

I have been very fortunate to work with Bugra Gedik as my co-advisor. I am very greatful for his ideas and discussions. I have learned a lot from him during this time.

I am also thankful for Data Mining group at Bilkent University. Particularly, I would like to thank Mehmet Güvercin for his patience and friendship.

I would also like to thank my thesis commitee members, Cevdet Aykanat and Savaş Dayanık, for their suggestions and for sparing time on this thesis.

Finally, I would like to thank my parents Kamile and Ebubekir and my sisters, Büşra, Ayşe, and Fatma. They have given me their endless love and years of support for my education and success in life. And I am very greatful to them for constantly reminding me what really matters in this life.

This work was partially supported by the Turkish Academy of Science (TÜBA) and The Scientific and Technological Research Council of Turkey (TÜBİTAK).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Over the last decade, with the enormous growth in the adoption of social networking services among people, viral marketing has gained newfound interest. Social networks, such as Facebook and Twitter, allow us to explore how ideas, information, and innovations spread. Companies spend more and more money each year to exploit these opportunities to gain attention, increase sales, and generate brand awareness. Consider a company that wants to initiate a large cascade of adoptions (or attention, awareness) by a careful selection of seed users. Selecting the seeds to maximize the cascade is a fundamental problem in viral marketing research and is referred to as the *influence maximization problem* [1]. It is defined on a directed graph, where vertices represent users, edges represent friendships among users, and weights associated with the edges represent the influence weights for the friendships. The goal is to target a set of k seed users that will maximize the expected *spread of influence*, also referred to as the *coverage*. A propagation process is started from the seed users and the coverage of these seed users is defined as the number of regular users (the *audience*) in the network that are activated at the end of the process.

Broker-based ad allocation. The problem gets more challenging when there are a number of companies (the *advertisers*) that want to initiate viral *advertisements* (ads). This calls for need of a *broker* system where the advertisers can sign-up to find seed users (the *endorsers*) for their ads, and endorsers can sign-up to find advertisers whose

ads they will spread. Suppose each advertiser has a predefined budget and each endorser has an associated *coverage* and *cost* (price they request for their service). An advertiser wants to pay a set of endorsers to achieve high coverage, considering its budget. As such, we want to assign each advertiser a set of endorsers to maximize its total coverage, without leaving any advertiser unassigned. Furthermore, we want to limit the number of advertisers an endorser is assigned to. This ensures that no endorser is *overburdened* or perceived as a spammer. Finally, we want to avoid the users getting *overloaded* due to too many ad messages.

Research challenges. There are several challenges in addressing this problem. First, the propagation model used should consider the *overloading* of the audience. There are several propagation models in the literature, Independent Cascade Model(ICM) and Linear Threshold Model(LTM) being the most widely studied ones [1]. These models mostly assume that the chance of a user to get influenced increases as more of her neighbors spread the information. In ICM, when a user becomes activated it has a chance to influence its inactive neighbors. If a neighbor is activated, then it further attempts to activate its inactive neighbors. In LTM, a user chooses an activation threshold uniformly at random and becomes active if the fraction of its active neighbors is greater than this threshold. For example, in Twitter, if a user is fond of a mobile phone, she can post tweets to her followers, recommending or praising the phone. If one of her neighbors buys or has already bought the phone, he may in turn recommend the phone to his followers. This may trigger a cascade of recommendations. Both propagation models (ICM and LTM) assume that as additional neighbors of a user become active, the aptitude of the user to adopt the product increases. This is in contrast to our problem setup, where overloading is a concern. Consider a user in Twitter who has received the same ad from a large number of her neighbors. Then the user may get overloaded by the same type of message. This may in turn cause the user to have a bad opinion about the ad, or unfollow several of her followers, or even churn. This is a serious problem in viral marketing that needs addressing.

Second, the broker-based ad allocation problem needs to be formalized as an optimization problem, where the coverage provided to the advertisers is to be maximized, while considering the budget constraints of the advertisers as well as the advertisement limits of the endorsers. Furthermore, if possible, no advertisers should be left without

coverage and the broker system should be fair to the advertisers in the sense that the budget used per coverage provided to ads with similar budgets should be similar.

Last but not the least, we need fast heuristic algorithms that can match advertisers to endorsers with close to optimal coverage. Ideally, the matching algorithm should be fast enough to be executed on large-scale social network databases.

Solution approach. We address each of these challenges in our work. First, in order to model the audience overload problem, we propose a novel propagation model and revisit influence maximization in the context of this model. An active user u tries to activate a neighbor v with a probability $p_{u,v}$ as in ICM. But the user gets overloaded if she receives too many ads. We model this by a probabilistic function where the user becomes overloaded with a probability $p_u(S)$ with every attempt, where S is the set of active neighbors. We assume that $p_u(S)$ is nondecreasing. We call this model *Independent Cascade Model with Overload (ICMO)*. We consider both *progressive* and *non-progressive* variations of this model. The progressive model preserves desirable properties such as submodularity and monotonocity, which help prove approximation bounds for the greedy algorithms. The non-progressive model aims to model also the overloading of previously activated users.

Second, we formalize the problem through a combinatorial optimization framework. We generalize the Maximizing Submodular Welfare problem to several linear constraints. We show that the generalized problem is NP-hard.

Last, we propose a cost-effective greedy algorithm called *CEAL* to solve the broker-based ad allocation problem. The algorithm operates by searching through the set of advertisements and endorsers to find the best pair that results in maximum marginal gain per cost. We successively pick (*ad, endorser*) pairs according to cost-effective marginal gain until we consider all endorsers. If there are more than one ads achieving maximum gain for the chosen endorser, we pick the ad that has the least remaining budget. Furthermore, the algorithm is guaranteed to assign at least one endorser for each ad, if there are enough number of endorsers. Our results show that the CEAL algorithm can provide high coverage at low computational cost, can use up most of the ad budgets given sufficient number of endorsers, and be fair to the advertisers with respect to the cost charged per coverage provided.

Contributions. This thesis makes the following contributions:

- We formalize the problem of *broker-based ad allocation* in social networks.
- We introduce the *ICMO propagation model* that captures the ad overload problem.
- We develop the *CEAL algorithm* – a fast greedy algorithm that can perform ad allocation with high accuracy and low computational cost.
- We provide an evaluation using real-world networks, giving insights on the problem of broker-based ad allocation and illustrating the effectiveness of our techniques.

The rest of this thesis is organized as follows. In Chapter 2 we give the necessary background and related work with their analysis. Especially the Ad Allocation Problem, the Submodular Welfare Problem and Influence Maximization and Propagation in social networks is given. In Chapter 3, we study the introduced propagation models in detail, give a formulation for the broker-based ad allocation problem, and explain the proposed algorithms. We present the results of our evaluation on several real-life networks in Chapter 4. We conclude the thesis in Chapter 5.

Chapter 2

Background and related work

In this chapter, we review the basic concepts and notational conventions used throughout this thesis. Next, we survey the works on influence maximization, ad allocation and submodular welfare problem.

2.1 Basic concepts and definitions

We briefly define necessary concepts and definitions and introduce some terminology used throughout this thesis.

2.1.1 Graph theoretic concepts and social networks

We represent a social network by a *graph*. A graph $G = (U, L)$ is a set of vertices U and a set of edges L between vertices. We interchangeably use the word vertex, node, user, individual and the word edge, link, relationship.

For each nodes $u, v \in U$, an edge from u to v is represented by a tuple $(u, v) \in L$ if any exists.

Next, we define several basic terminology and their relationships with a social network:

Directed and undirected graph: A graph is *undirected* if $(u, v) \in L \iff (v, u) \in L$. A graph is called *directed* if the order of the pairs for each edge is important. We can also use the same convention for each edge seperately. An edge $(u, v) \in L$ is undirected if the pairs are unordered, and directed otherwise. An undirected edge may reflect the friendship between two users which generally is a mutual relationship. A directed edge on the other hand may reflect that one user is following the interests of another user.

Subgraph: A subgraph $G_z = (V_z, E_z)$ of a graph $G = (U, L)$ is a graph where $E_z \subseteq L$ and $V_z = \{u, v : (u, v) \in E_z\}$.

Node degree: For directed graphs, we divide the degree of a node u into two: indegree $\delta_{in}(u)$ of u is the number of nodes where there is a directed edge from v to u, outdegree $\delta_{out}(u)$ of u is the number of nodes where there is a directed edge from u to v. For undirected graphs there is only one type of degree, and the degree of a node u is the number of nodes $\delta(u)$ where there is an edge between u and v.

We make a distinction between a social network and an influence network. Different from a social network, an influence network also represents relationships between individuals as an influence relationship.

Influence network: An influence network $G = (U, L, p)$ is a social network where p is the influence function. For each edge $(u, v) \in L$, $p_v(u)$ reflects the influence of node v over u.

2.2 Cascading behavior and influence maximization in social networks

Information cascades are the processes of wide spread transmission of an idea, innovation, advertisement, or disease due to the peer-to-peer influences of individuals. The phenomena of how innovations spread in a social environment is first studied by sociologists [2]. More recently, the phenomenon has attracted different researchers from

various fields including finding seed users for viral marketing [3, 1], finding inoculation targets in case of spread of an epidemic [4], and explaining evolution of networks in blogosphere [5].

There are two basic models that try to capture the spread of influence in a social network: Linear Threshold Model (LTM) [6] and Independent Cascade Model (ICM) [7].

2.2.1 Linear Threshold Model

The process is based on node-specific *thresholds*. Given an influence network $G = (U, L, p)$, a node u is influenced by her neighbors v according to a weigth $p_v(u)$ where $\forall u \in \mathcal{M}(u), \sum_v p_v(u) \leq 1$. Then, each node u in the network picks a random threshold value $\theta_u \in [0, 1]$ uniformly at random. This threshold value indicates the weighted fraction of neighbors of u that has to be *active* in order for u to get *active*. Given the threshold distribution over each node in the network and initial set of seed users S_0 (initally active nodes) at time t_0, the diffusion process unfolds according to the following process: at time t each *active* node remains *active* at time $t + 1$ and each *inactive* node u becomes *active* if the sum of the weights of the edges between u and her *active* neighbors is at least the threshold θ_u that u set for herself:

$$\sum_{v \in \mathcal{M}u} p_v(u) \geq \theta_u \tag{2.1}$$

The process continues as such there is no new *active* node. Threshold values reflect the tendency of users to adopt the idea when their friends start to get *active*. The reason why the thresholds are randomly chosen is because of the fact that our the lack of knowledge of the respective values of the thresholds.

The weights of the edges indicates the tendency of a user to get influenced by her neighbors. It shows the authority, influence, or friendship of a neighbor $v \in \mathcal{M}(u)$ over u.

2.2.2 Independent Cascade Model

ICM captures the dynamics of diffusion of an idea in a step-by-step manner. Given an influence network $G = (U, L, p)$, a node u is influenced by her neighbors v according to a weigth $p_v(u)$ where $\forall u \in \mathcal{M}(u), \sum_v p_v(u) \leq 1$, and $p_v(u)$ is the probability that v is succesful in the attempt of activating u. Let's assume that at time t a node v gets *active*. Then, v has a one chance of activating her previously inactive neighbor u with probability $p_v(u)$. If she succeeds, then v is *active* at time $t + 1$, otherwise v stays *inactive*. If multiple neighbors of v gets active at t, then their attempts are sequenced arbitrarly. Each of these neigbors independently attempts to activate u at time t with respective probabilities and if any of them is succesful then v gets *active* at time $t + 1$. Given the set of *active* neighbors $\mathcal{S}_t(u)$ of u at time t, the probability that u is successful at $t + 1$ is

$$Pr(u \textit{ is active at t+1 }) = 1 - \prod_{v \in \mathcal{S}_t(u)} (1 - p_v(u)) \tag{2.2}$$

The process continues as such there is no new *active* node.

Similar to LTM, the weights of the edges indicates the tendency of a user to get influenced by her neighbors. Again, it shows the authority, influence, or friendship of a neighbor $v \in \mathcal{M}(u)$ over u.

Decreasing Cascade Model (DCM). The probability $p_v(u)$ that u is influenced by v depends only on the individual authority of v possesses over u. But as the failed attempts to influence u increases, u may get *marketing-saturated*. Thus the probability of further attemps drops with each unsuccessful attempt. This natural restriction is modeled by generalizing the probability functions $p_v(u)$ as nondecreasing functions $p_v(u, S)$ $(p_v(u, S) \geq p_v(u, T))$ for $S \subseteq T$ where S is the set of nodes that already attempted to influence u but failed.

2.2.3 Influence Maximization

One of the most fundamental question in viral marketing is how do we select a set of nodes in a social network for an ad campaign so that the number of users in the network

that received the ad is maximized? The problem is referred to *influence maximization problem* and there has been a wide-spread work on different variants of the problem.

Informally, influence maximization is the problem of finding a set of seed nodes in a network so that the spread of influence initiated by these seeds is maximized. The problem is first introduced by Domingos et al. [3] and formalized as a discrete optimization problem by Kempe et al. [1, 8].

Influence Maximization Problem. Given an influence network $G = (U, L, p)$, the influence maximization problem is to find a set of nodes S of size k where if the cascade process is initiated by these nodes, then the number of final active set of nodes is maximized. We use $\sigma_G(S)$ to denote the coverage of the set S over the network G. We omit G when the context is clear.

The influence maximization problem is NP-hard [1]. The most fundamental result on the problem is based on the submodularity of the influence function.

Submodular set functions. Submodular set functions are discrete analogs for convex functions on real space. A set function $f(.)$ is called submodular if it satisfies the so called *diminishing returns* property [38]: the marginal gain from adding an element to a set S returns at least as high as the marginal gain from adding an element to any set T where $S \subseteq T$. More formally, a function f is called submodular iff

$$f(S \cup u) - f(S) \geq f(T \cup u) - f(T) \tag{2.3}$$

From influence maximization point of view, submodular set functions have a desirable property. Consider the case where $f(.)$ is submodular and also *monotone* $(f(S \cup u) \geq f(S))$. The purpose is to find a set S of size k such that $f(S)$ is maximized. This problem is NP-hard but it has been shown that the following simple greedy hill-climbing algorithm gives a $(1 - 1/e)$ approximation to the optimum : start with empty set and iteratively add one element u to S where the marginal gain $f(S \cup u) - f(S)$ is maximized. It is shown that the resulting influence function for all the models we considered is submodular and monotone [1], [8]:

Theorem 1 *The influence function $\sigma(.)$ for ICM, LTM, and DCM is submodular and monotone.*

The basic idea to find the set S over a social network is to run simulations to obtain the value of $f(S \cup u)$ and $f(S)$ and use greedy algorithm.

Different approaches for cascading behavior on networks are developed since, including data-centric influence propagation [9], minimizing budget and time together [10], budgeted influence maximization [11], emergence of competitive and opposite opinions [12, 13, 14], based on product adoption [15], and topic-sensitive [16, 17, 18, 19]. There has also been work on scalable and parallel influence dissemination [20, 21, 22, 23, 24]. We will briefly explain the idea behind some of the approaches that is used throughout this thesis.

Budgeted influence maximization. In the traditional influence maximization problem, nodes in the network has uniform cost, i.e., each node has unit cost. The problem gets more challenging if nodes may have nonuniform costs and we have a budget constraint on how much we can spend. Instead of selecting k nodes, we now need to select a set of nodes where the total costs of the nodes are not violating the budget. The simple solution would be to use greedy algorithm to select the nodes maximizing the cost-effective marginal gain, i.e., given a cost function $c(.)$, the cost effective marginal gain is $(\sigma(S \cup u) - \sigma(S))/c(u)$. But this simple solution has an unbounded error and a small modification resolves the issue [11]. In the simplest case, the algorithm works as follows: iteratively run cost-effective greedy algorithm and obtain S, also run simple greedy algorithm and obtain S'. Both of the algorithms stop if $\forall u \in U \setminus S, c(S \cup u) > b$. If $\sigma(S) > \sigma(S')$ use S, otherwise use S'. This algorithm is called CEF and provides a $1/2(1 - 1/e)$ approximation to the optimal.

Scalable and parallel influence dissemination. Computing the expected spread given a seed set is ♯P-hard under both LTM [21] and ICM [20] model. One needs to run a large number of simulations to obtain an accurate esimate of $\sigma(S)$. This is cumbersome for even networks of size of thousands and several heuristics are proposed to scale the greedy algorithm to large datasets [20, 21, 22, 23, 24]. The idea is based on restricting computations on the local influence regions of nodes. The size of the regions is variable which enables tunable tradeoff between accuracy and efficiency. The main idea is to use arborescence structures (a tree where edges are directed towards (in-arborescence) or from (out-arborecence) root) to estimate the influence. The

arborescence trees is built using *maximum influence paths (MIP)* which are the paths having maximum influence value for each pair of nodes. MIPs are estimated using *Dijkstra shortest-path algorithm*, and some of the MIPs which have probability smaller than the tunable threshold value are discarded. By unioning MIPs, they build *Maximum Influence In-Arborescence (MIIA)* and *Maximum Influence Out-Arborescence (MIOA)* trees and estimate influence only on these local arborescence structures. Given a set of seed nodes, let the *activation probability* of a node u in an MIIA be the probability that u is activated on this MIIA when the propagation is initiated by these seed nodes. Then the influence of these seeds over network is the sum of the activation probabilities for each node u estimated on the MIIA having u as the root. This model is called *Maximum Influence Arborescence (MIA)*.

The *Degree Discount* method [23] can be considered as a special case of MIA. To build local influence regions, only 1-hop distant neighbors of nodes are used. Then, for each node a score is assigned based on the expected influence of the node over her neighbors. The idea is valid only in case of uniform weights.

The MIOA structures also yields parallel influence estimations [24]. Instead of using activation probabilities, they use MIOA structures and estimate influence using inclusion-exclusion. The process can be run in a parallel setting which provides further scalability.

Information Overload. Information overload in a social network refers to the concept that the number of messages, tweets, notifications reaches a level beyond a user can process in a reasonable time. The closest overload work to our ICMO model is [25], which models the overload of a sequence of different messages by an exponential decay parameter. They focus on activation of messages (messages that are not ignored), whereas we focus on activation of users and their overload due to repeated messages. Our model is significantly different, as the focus is on user overload and once a user is overloaded, no future activations are possible. Also, they do not investigate whether or not the influence function with the proposed overload measure is submodular.

2.3 Ad Allocation and Submodular Welfare Problem

Combinatorial allocation problems refer to the assignment of a set I of m items among n players such that the total utility provided to the players is maximized. The most general case of this problem is to find a partitioning $(I_1, I_2, ..., I_n)$ of the items among players such that the total utility is maximized, i.e., $\sum_{i=1}^{n} w_i(S_i)$ where $w_i : S^I \rightarrow \mathcal{R}$ is the utility function for player i. This general case is NP-hard [29]. The problem is divided into categories based on the properties of the utility functions. If the utility function is submodular, then the problem is called *Maximizing Submodular Welfare Problem (MSW)*.

2.3.1 Maximizing Submodular Welfare Problem

MSW problem is first studied by [26]. They show that MSW problem is NP-hard and the following simple greedy algortihm gives 2-approximate solution to the optimal: start with empty assignments to each player, iterate over items and for each item x pick the player with highest $w_i(I_i \cup x) - w_i(I_i)$ and assign x to player i.

The problem is also studied under different oracle models [27, 28, 29].

- A *value oracle* gives the result of a basic query: *what is the result of* $w_i(S)$?
- A *demand oracle* answers queries of the form: *given the prices* p_x *for each item* $x \in I$*, what is the result of* $\underset{S \cup I}{\operatorname{argmax}} w_i(S) - \sum_{x \in S} p_x$?

A randomized algorithm is proposed for MSW problem in a *value oracle* model which provides a $(1 - 1/e)$ approximation [27]. The algorithm employs a *continuous greedy heuristic* which returns an approximate solution to a *non-linear continuous optimization problem*. The idea is based on obtaining canonical extensions to smooth monotone submodular functions by taking expectation. Then the process unfolds by finding a local optimal value for the extension by only considering local values.

The approximation ratio of the problem is improved under a *demand oracle* model.

A $(1 - 1/e + \epsilon)$ approximate randomized algorithm is introduced in [29] for some absolute constant ϵ. The algorithm first reduces the problem into a linear programming and resolves conflicts between players using a new technique called *fair content resolution*. Suppose that several players are requesting an item with different probabilities. The idea to resolve the conflict among players is to assign the item to players with the same probability.

Although the approximation ratio of the problem can be improved if a demand oracle is used [29], an $1 - 1/e$-approximation [30] is the best approximation ratio that one can obtain [31] under a value oracle model.

In the online version of the problem, the following greedy algorithm is optimal [32] using coverage valuations in which a valuation function $w : 2^I \rightarrow \mathcal{R}^+$ is a coverage valuation if there is a set system $Y_i : i \in I$ such that $w(S) = |\bigcup_{i \in S} Y_i|$: allocate each incoming item to the player maximizing the marginal gain. The algorithm gives $1/2$ approximation to the problem.

Under a stochastic setting with iid items and valuations satisfying diminishing returns, the same greedy algorithm gives $(1 - 1/e)$ approximation.

Different than our setup, the problem is not studied in a social network environment. Furthermore, unlike our problem, players do not have budget constraints and assignments do not have limit constraints in these problems.

2.3.2 Budgeted Ad Allocation

Budgeted allocation is the problem of maximizing the total profit extracted by the algorithm under a budget constraint for each player. The utility function in this problem is linear, i.e., $w_i(S) = \sum_{j \in S} w_{ij}$.

The problem is studied in [33, 34] assuming an offline setting. Both of the algorithms are based on the linear programming relaxation of the problem. Then, the solution to the original problem is obtained by a rounding schema afterwards. The competitive ratio of the algorithms is $4/3$ and $3/2$ respectively.

In an online setting, where the set of impressions arrive online, the objective is to assign impressions to advertisers whenever an item arrives. An assignment algorithm with *free disposal* is proposed in [35] where players are assigned with items more than the number of items they have in their contract for the online ad allocation problem. The idea is that, players are indifferent to assigning more than they requested because the value of an assignment only considers the items having highest values and have the same amount as requested.

A primal-dual training based algorithm is proposed for online ad allocation [36]. The algorithm provides a $(1 - o(1))$ approximation ratio for the problem. They also consider the efficiency and fairness of the algorithm and show that there is a trade-off between these.

An optimization algorithm for the online bipartite matching problem is proposed in [37]. The idea is to compute two disjoint solutions to the expected instance of the problem in an offline setup and use both of them in the online allocation algorithm.

The objective function in these problems can be explicitly stated as linear formulas. In contrast, our objective function has no open form and depends on coverage computation on the social network, which often requires simulations to estimate its value.

To the best of our knowledge, no previous work has studied the broker-based ad allocation problem we have formulated in this thesis. In addition, the ICMO model we developed, including progressive and non-progressive variants, to realistically model the coverage computation in ad dissemination is not covered elsewhere.

Table 2.1: Table of symbols

SYMBOL	DESCRIPTION
$G = (U, L)$	Regular graph
U	Vertex set
L	Edge set
(u, v)	Edge in the graph
G_z	Subgraph
W	Weight function for the edges
$p_v(u)$	Influence function under ICM and LTM
$\delta_{in}(u)$	Indegree of u
$\delta_{out}(u)$	Outdegree of u
$\delta(u)$	Degree of u
$\mathcal{M}(u)$	Neighbors of node u
θ_u	Activation threshold for node u
$p_v(u, S)$	Influence function under DCM
$\sigma_G(S)$	Coverage of S over G
$c(S)$	Total cost of the set of users S
$w_i(S)$	Utility function for player i
$(\mathcal{C})$	Set of advertisers
$\mathcal{E}$	Set of endorsers
$\mathcal{A}_i$	Set of ad campaigns started by advertiser C_i
a_i	Number of ads started by advertiser C_i
$B(A^i_j)$	Budget of the ad A^i_j
S^i_j	Set of endorsers assigned to ad A^i_j
$R(E_l)$	Reverse assignment of ads to endorser E_l
$p^u_o(S)$	Overload distribution function (ODF)
$G = (U, L, p)$	Influence network
$G = (U, L, p, p_o)$	Influence network with overload
$G_t = (V_t, E_t)$	Graph at time t
K_T	Set of active nodes at time t
D_t	Set of overloaded nodes at time t
$\mathcal{D}$	Entire set of replicated endorsers
$\delta^i_j(E_l)$	Marginal gain for adding E_l into S^i_j
$Q(E)$	Priority queue over endorsers

Chapter 3

Broker-Based Ad Allocation

In this chapter, first we present the design of our broker system. Next, we formalize the ad allocation problem.

3.1 Broker System

Advertisers want to benefit from viral advertising by paying a set of endorsers (seed users) that will spread their ads, and endorsers want to spread ads to make profit. We design an intermediary called the "*broker*" to find a match between the advertisers and endorsers. We consider four different perspectives while designing the system: the advertiser, the endorser, the broker, and the audience.

Advertiser Perspective. The advertiser registers its ads with the system by specifying a maximum budget to be spent for each ad. The broker system provides a simulated coverage of how wide the ad campaign can spread. If the company wants a wider coverage, it may increase the budget. On the other hand, while the total money paid by the advertiser cannot exceed its budget, the entire budget may not be used if the coverage that can be provided by the broker system is not sufficiently high.

Endorser Perspective. The endorser registers with the system and specifies a cost per ad it will endorse. The cost of an endorser is assumed to be dependent, but not necessarily linear, on the number of users from the audience it can reach. The broker

system assigns ads to the endorsers. Thus, the amount earned by the endorser is given by its cost times the number of ads it is assigned. If an endorser's cost per ad is too high, the broker may not assign it any advertisers.

Broker Perspective. The broker plays the role of a matcher between the advertisers and the endorsers. It matches a set of endorsers to each ad considering the budget constraints and the nature of ad propagation on the social network. It can also provide a coverage estimate for potential advertisers. The goal of the broker is to provide the highest coverage possible to the advertisers. The broker can take a fixed percentage of what is paid to the endorsers.

Audience Perspective. There are three kinds of behavior a regular user who received an ad message may exhibit: 1) the user is interested in the content and forwards the message to her friends; 2) the user may or may not be interested but does not propagate the message, yet she may do so for a future message; 3) the user has received too many messages and whether or not she is interested she does not propagate the message and will not do so for future messages. The goal of the system is to gain the interest of users without them getting overloaded due to too many ad messages.

3.2 Problem Formulation

Our purpose is to design the broker considering the following informal objective:

- Provide the highest coverage possible to each advertiser

And with the following informal constraints:

- Avoid violating the budget limits of advertisers
- Avoid overburdening endorsers and avoid endorsers being perceived as spammers
- Avoid overloading of the audience with ads and avoid ads being perceived as spam

We now explain how we incorporate each of the above objectives and constraints into our formal model. We assume that we are given an influence network $G = G(U, L, p)$, where U is the set of users, L is the set of relationships, and $p : L \rightarrow [0, 1]$ is the influence function. A set of advertisers $\mathcal{C} = \{C_1, C_2, \cdots, C_n\}$ and a set of endorsers $\mathcal{E} = \{E_1, E_2, \cdots, E_k\} \subset U$ with corresponding costs $c(E_i)$ for $E_i \in \mathcal{E}$ have enrolled in the system. Here, $n = |\mathcal{C}|$ is the number of advertisers and $k = |\mathcal{E}|$ is the number of endorsers. Each advertiser C_i has started a set of ad campaigns $\mathcal{A}_i = \{A_1^i, A_2^i, \cdots, A_{a_i}^i\}$, with corresponding budgets $B(A_j^i)$ for $A_j^i \in \mathcal{A}_i$. Here, $a_i = |\mathcal{A}_i|$ is the number of ads started by the advertiser $C_i \in \mathcal{C}$, and $a = \sum_i a_i$ is the total number of ads.

Disseminating Influence. We define the coverage of a set $S \subseteq \mathcal{E}$ of endorsers on G by $\sigma_G(S) : 2^{\mathcal{E}} \rightarrow U$. Our primary goal is to assign a set of endorsers $S_j^i \subseteq \mathcal{E}$ for each ad A_j^i such that the total coverage is maximized. Given this assignment, we denote the reverse assignment of ads to endorsers as $R(E_l) = \{A_j^i \mid E_l \in S_j^i\}$ for endorser $E_l \in \mathcal{E}$.

Essentially, we try to maximize:

$$\sum_{i=1}^{n} \sum_{j=1}^{a_i} \sigma_G(S_j^i) \tag{3.1}$$

Handling Budget Constraints. We model the budget constraints using inequality conditions for each ad. That is, for each ad, the total cost of the endorsers assigned should not exceed the budget allocated for the ad. Formally:

$$\sum_{E \in S_j^i} c(E) \leq B(A_j^i), \underset{i \in [1..n], j \in [1..a_i]}{\forall} \tag{3.2}$$

Avoid Spamming and Overburdening of the Endorser. Both spamming and overburdening has the same characteristic. If an endorser is disseminating a lot of ads, then the users will perceive the endorser as a spam account. Also if we assign a lot of ads to an endorser, the endorser may get overburdened. We use a threshold-based approach to prevent overburdening and spamming from an endorser. We assign a threshold value to each endorser that limits the number of ads assigned to it. More formally, we pick a threshold θ_j for an endorser $E_j \in \mathcal{E}$ as its dissemination limit. Then the following

constraint is satisfied:

$$|R(E_j)| < \theta_j, \underset{j \in [1..k]}{\forall} \tag{3.3}$$

Avoid Overloading of a Network User. Consider a case where a user is in the close social network proximity of several endorsers. If a large portion of these endorsers are assigned the same ad, then the user will get a lot of similar ad messages and eventually get overloaded. We propose an extension of the ICM to model the overloading of a user with respect to an ad campaign. We introduce a probabilistic component to ICM that handles overloading, while preserving some desirable properties such as *submodularity* and *monotonicity*. This model impacts the coverage $\sigma_G(S)$, as when a user from the audience gets overloaded, the coverage suffers. As such, overloading is handled as part of the objective function.

3.2.1 Independent Cascade Model with Overload

In ICM, if a user adopts a product and becomes active, she further tries to activate her neighbors. As more of a user's neighbors become active, the tendency that the user will get activated increases. For a uniform ICM where $p(l) = z$, $\forall l \in L$, given that k neighbors of a user adopted an idea, the probability that the node will adopt it is $1 - (1 - z)^k$. As k increases, this probability always increases. However this does not hold in modern social networks where each trial to activate a neighbor is accomplished by posting a message, sending an email, or posting a tweet. If a user has a lot of active friends, then the user will receive a lot of messages which will eventually cause the user to get overloaded.

We now propose the Independent Cascade Model with Overload (ICMO). Let $N_u \in U$ be the list of neighbors of a user $u \in U$ in the social network $G(U, L, p)$. We define an Overload Distribution Function (ODF) $p_o^u : 2^{N_u} \rightarrow [0, 1]$. Given the set of active neighbors $S \subseteq N_u$ of a node u, $p_o^u(S)$ is the probability that the user u is overloaded. The ODF p_o^u is monotonically increasing on $|S|$. An example ODF is given by $p_o^u(S) = 1 - (1 - z)^{|S|}$.

In ICMO, we differentiate between the influence of a user v on u and the activation

of u. If the attempt to influence u is successful then u is interested in the idea but not necessarily activated. There is a chance for an *interested* user u to get *activated* with probability $1 - p_o^u(S)$, and *overloaded* with probability $p_o^u(S)$. If u is activated, then it will attempt to activate her inactive neighbors. We first study a *progressive* model where an active user will always be active and an overloaded user will always be overloaded as sending more messages to influence the user will not have any positive effect. Thus all the subsequent attempts to activate u are unsuccessful. We denote the ICMO model by $G(U, L, p, p_o)$.

We will now study the influence maximization problem on ICMO. Influence maximization problem is to choose k seed users from the network such that the influence propagation is maximized when initiated by these seeds. Finding influential users in ICM is NP-hard. Intuitively the ICMO problem is NP-hard as well. However, as we will show, the influence function is submodular and monotone under ICMO. Thus, for any $\epsilon > 0$, it is possible to find a simple greedy algorithm that will lead to a $(1 - 1/e - \epsilon)$-approximation [38].

Theorem 2 *The influence maximization problem using the ICMO model is NP-hard.*

Proof If we set $p_o^u(.) = 0$ for each user u, then we have a linear time transformation of the ICM problem into ICMO. Since ICM is NP-hard, this completes the proof.

Theorem 3 *The coverage function σ_G using the ICMO model is submodular and monotone.*

Proof We first reduce the problem to a simpler cascade model and show that if influence under this model is submodular and monotone, then influence under ICMO is also submodular and monotone. Borrowing the *live-edge* idea from [1], we flip coins for each edge prior to the propagation. We obtain graph $G' = (V, L', p_o)$ where if $e \in L'$ then e is *live*. Each *live-edge* (v, u) reflects whether or not a neighbor v has been succesful to attract the interest of u. Then the influence of a seed set S on G is equal to $\sigma_G(S) = \sum_{G'} P[G'] \cdot \sigma_{G'}(S)$ where $P[G']$ is the probability of obtaining G'.

It is easy to see that if $\sigma_{G'}(.)$ is submodular and monotone, then $\sigma_G(.)$ is submodular and monotone, as non-negative linear combinations of submodular and monotone functions are also submodular and monotone.

To show the submodularity of $\sigma_{G'}(.)$, we show that the influence propagation on G' is a special case of influence propagation on Decreasing Cascade Model (DCM). We first define a timed version of the diffusion process on G'. Start by activating a set of seed nodes S in G'. If u has any path to $v \in S$ consisting of only *live-edge*s than u gets activated with probability $1 - p_o(Z_t^u)$, otherwise it gets overloaded with probability $p_o(Z_t^u)$, where Z_t^u is the set of *active* neighbors of u at time t. If there is no such path, then u stays *inactive*. Now consider the following diffusion process of DCM on graph $G'' = (U, L'', p'')$: $L'' \subseteq L'$, $\delta_{in}(u) = 1$, and u is *active* with probability $p''(Z_t)$, where $\delta_{in}(u)$ is the in degree of u, and Z_t is the number of *active* neighbors of u at time t. If $p''(.) = 1 - p_o(.)$ then the two processes are equal. Since the influence function $\sigma_{G''}(.)$ under DCM is submodular and monotone, $\sigma_{G'}(.)$ is also submodular and monotone, completing the proof.

3.2.2 Non-progressive ICMO

The ICMO model we just described is a *progressive* model where if a node is *active* then it will remain *active* throughout the process. But in real-life an *active* node may also become *overloaded* if she receives a lot of ad messages. This is different from traditional *non-progressive* propagation models because once a node gets *overloaded*, she may never become *active* again. Figure 3.1 gives an overview of the state transition diagram for non-progressive ICMO.

We can model this problem as a *non-progressive* ICMO on time-stamped subgraphs of the original graph over a time period $[0, T]$. Let $G_t = (V_t, E_t)$ be the graph at time t where $G_o = G$, $K_t \subseteq V_t$ be the set of active nodes at time t, and $D_t \subseteq V_t$ be the set of *overloaded* nodes at time t. We start the dissemination process by activating a set of seed users $S \subseteq V_0$. At time $t = 1$, some of the users in the network may get *overloaded*. These users will not disseminate any ad messages and they will stay overloaded until the end of the process. Thus, we remove the overloaded nodes, that

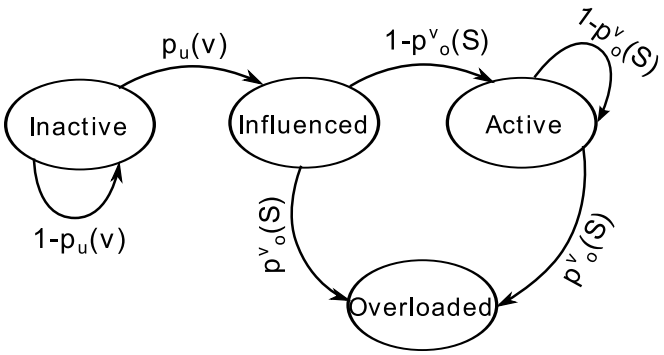


Figure 3.1: State transition diagram of non-progressive ICMO

is $G_1 = G_0[V_0 \backslash D_1]$. Next, the active neighbors $L_u \subseteq K_1$ of an *inactive* node u will try to influence u. If any of them succeeds, then we have $u \in K_2$ with probability $1 - p_o^u(L_u)$, otherwise u is *overloaded*. This process continues similarly, and we are interested in the number of *active* nodes at time $t = T$. This is different from the traditional propagation models [1] that focus on the expected coverage over the entire duration of the propagation process. Differently, we are interested in the users that are active at the end of the process and only those users are counted towards the coverage. While *non-progressive* ICMO is neither monotone nor submodular, our experimental study shows that greedy solutions still provide good results in practice.

3.2.3 Cost Effective Influence Maximization

Consider the case where we have a budget and each user has a cost. Cost effective influence maximization is the problem of finding a set of influential users $\mathcal{I}$ from a set of candidates $\mathcal{N}$ where the coverage of $\mathcal{I}$ is maximized and the budget is not violated. It is proven that CELF (Cost Effective Lazy Forward selection) algorithm achieves a constant rate approximation for the problem [11]. We can easily incorporate our ICMO model into CELF algorithm.

3.2.4 Formalization of Broker-based Ad Allocaton

We are now ready to define our ad allocation problem.

Definition (Broker-based Ad Allocation Problem) Given an influence network $G = G(U, L, p, p_o)$, find a set of endorsers S_j^i for each ad A_j^i, such that:

$$\sum_{i=1}^{n} \sum_{j=1}^{a_i} \sigma_G(S_j^i) \tag{3.4}$$

is maximized, subject to:

$$\sum_{E \in S_j^i} c(E) \leq B(A_j^i), \underset{i \in [1..n], j \in [1..a_i]}{\forall} \tag{3.5}$$

$$|R(E_j)| < \theta_j, \underset{j \in [1..k]}{\forall} \tag{3.6}$$

In what follows, we seek computationally efficient approximations for this problem.

Our objective function (Eq. 3.4) is closely related to the "Maximizing Submodular Welfare (MSW)" problem [26]. In the MSW problem, when adopted to ad allocation, the objective is to allocate endorsers to ads such that Eq. 3.4 is maximized and allocations are disjoint, i.e., $S_{j_1}^{i_1} \cap S_{j_2}^{i_2} = \emptyset, \forall j_1 \neq j_2, i_1 \neq i_2$.

In our problem, allocations can have non-empty intersections, that is the same endorser can be assigned to more than one ads. Furthermore, our problem involves budget constraints. We resolve the non-empty intersection problem by replicating the endorsers. For each endorser E_i, we replicate it by θ_i times, yielding $E_i^1, E_i^2, \cdots, E_i^{\theta_i}$. With this modification, the budget constraints remain as the only additional constraints on top of what can be supported by an MSW formulation.

MSW problem is known to be NP-hard [26]. *Greedy algorithms* [26, 39, 40, 41] are commonly used as heuristics for solving the MSW problem. In the simple case, for each endorser E an ad A is chosen such that the *marginal gain* is maximized:

$$A = \underset{i \in [1..n], j \in [1..a_i]}{\operatorname{argmax}} \sigma_G(S_j^i \cup E) - \sigma_G(S_j^i) \tag{3.7}$$

The algorithm stops when all the endorsers are considered.

3.3 Approximability of the problem

The simple greedy algorithm we outlined provides a near optimal competitive ratio for some special cases, which we look at next.

Unit cost and infinite budget case. In the unit cost and infinite budget case, we can reduce the ad allocation problem to MSW. Let $E_i^1, E_i^2, \cdots, E_i^{\theta_i}$ be the replication of the endorser E_i, and $\mathcal{D} = \{E_1^1, \cdots, E_1^{\theta_1}, \cdots, E_k^1, \cdots, E_k^{\theta_k}\}$ be the entire set of replicated endorsers. Then the ad allocation problem is reduced to MSW with the following optimization problem:

$$\underset{\mathcal{S}=P(\mathcal{D}),|\mathcal{S}|=a}{\operatorname{argmax}} \sum_{S\in\mathcal{S}} \sigma(S), \tag{3.8}$$

where $P(\mathcal{D})$ is a partitioning of $\mathcal{D}$ and $\mathcal{S}$ is a partitioning that has as many partitions as there are ads. The greedy algorithm for the MSW problem provides a 2-approximation [26]:

Theorem 4 *Let ALG be the result returned by the above greedy algorithm and OPT be the optimal solution. Then:*

$$ALG \geq 1/2 \cdot OPT \tag{3.9}$$

Hence, the greedy algorithm is guaranteed to find a solution with an objective value that is at least half of the optimum value. Furthermore there is no polynomial time approximation algorithm for the MSW problem having a competitive ratio larger than $1 - 1/e$ [31].

Non-uniform cost with one ad case. In non-uniform cost with a single ad campaign case, the problem is to find the set of endorsers having maximum coverage without violating the budget. This is called the *budgeted influence maximization problem* [11]. The cost-effective forward selection (CEF) algorithm for this case provides a constant factor approximation [11]:

Theorem 5 *Let ALG be the result returned by CEF and OPT be the optimal solution. Then:*

$$ALG \geq (1 - 1/e)/2 \cdot OPT \tag{3.10}$$

NP-hardness of the problem. Both of the above special cases of the broker-based ad allocation problem are NP-hard, which intuitively suggest that broker-based ad allocation is NP-hard too. We prove it formally as well:

Theorem 6 *The broker-based ad allocation problem is NP-hard.*

Proof If each endorser has unit cost, i.e., $c(E_i) = 1$, and if ads have infinite budgets, the problem is reduced to MSW problem. Following the results of [26], MSW is NP-hard and consequently the ad allocation problem is NP-hard.

Motivated by this result, in the rest of this chapter we describe heuristic techniques for solving the ad allocation problem. Importantly, the heuristic we develop not only provides a near optimal solution to the general problem, but also it satisfies the approximation bounds of the special cases outlined above.

3.4 CEAL Algorithm

We now describe our Cost Effective Ad Allocation (CEAL) algorithm used to solve the broker-based ad allocation problem.

The CEAL algorithm is based on iteratively finding the (*ad, endorser*) pair maximizing the marginal gain considering the cost of the endorsers, while preserving the ad budget and endorser capacity constraints. We cache the marginal gains of (*ad, endorser*) pairs and update the estimations only when necessary.

We start with an empty set of assignments for each ad. Initially the marginal gain of an endorser E over ads is her coverage, i.e., $\sigma(E) - \sigma(\emptyset)$. We choose the endorser that has the maximal marginal gain per cost, i.e., $\mathrm{argmax}_E\, \delta(E)$, where $\delta(E) = (\sigma(E) - \sigma(\emptyset))\,/c(E)$. At this point, any one of the ads having adequate budget for the endorser can be chosen for the assignment. Among these, we choose the ad that has the minimum remaining budget because she has the least likelihood of getting assigned an endorser. As long as there is at least one ad with enough budget that has

Algorithm 1: CEAL-initial-assignment($G, \mathcal{E}, \mathcal{D}$)

$S_j^i \leftarrow \emptyset, \forall i \in [1..n], j \in [1..a_i]$ ▷ Reset endorser to ad assignments

$\beta_i \leftarrow 0, \forall i \in [1..k]$ ▷ Reset used endorser capacities

while $\mathcal{E} \neq \emptyset \wedge \mathcal{D} \neq \emptyset$ **do** ▷ Endorsers and ads remain

  ▷ Compute the marginal gain per cost for all pairs

  $\delta_j^i(E_l) \leftarrow \frac{\sigma_G(S_j^i \cup E_l) - \sigma_G(S_j^i)}{c(E_l)}, \forall i \in [1..n], j \in [1..a_i], l \in [1..k]$

  ▷ Find the best pair among all viable (ad has enough budget)

  $(E_{\bar{l}}, A_{\bar{j}}^{\bar{i}}) \leftarrow \underset{E_l \subset \mathcal{E}, A_j^i \subset \mathcal{D}, c(E_l) \leq B(A_j^i)}{\operatorname{argmax}} \langle \delta_j^i(E_l), -B(A_j^i) \rangle$

  $S_{\bar{j}}^{\bar{i}} \leftarrow S_{\bar{j}}^{\bar{i}} \cup E_{\bar{l}}$ ▷ Make the assignment

  $\beta_{\bar{l}} \leftarrow \beta_{\bar{l}} + 1$ ▷ Increment used capacity for endorser

  if $\beta_{\bar{l}} = \theta_{\bar{l}}$ **then** ▷ Endorser is full

    $\mathcal{E} \leftarrow \mathcal{E} \setminus E_{\bar{l}}$ ▷ Remove endorser

  $B(A_{\bar{j}}^{\bar{i}}) \leftarrow B(A_{\bar{j}}^{\bar{i}}) - c(E_{\bar{l}})$ ▷ Decrease ad's remaining budget

  if $B(E_{\bar{l}}) < \min_{E \in \mathcal{E}} c(E)$ **then** ▷ Ad cannot take more endorsers

    $\mathcal{D} \leftarrow \mathcal{D} \setminus A_{\bar{j}}^{\bar{i}}$ ▷ Remove the ad

return $\{S_j^i\}, i \in [1..n], j \in [1..a_i]$ ▷ Return the assignments

not been assigned with any endorser, the maximum marginal gain of the best endorser will stay the same. Thus, we continue to select the endorser that has the maximum $\delta(E)$ until we assign each ad with one endorser or the endorser's capacity is reached.

We should note that, in most practical scenarios, the algorithm will assign at least one endorser to each ad. This is because the marginal gains are maximized for ads that have empty assignments. This may not hold in extreme cases, such as when there is an ad that does not have enough budget for a singe endorser, or the number of endorsers are small and they do not have enough capacity for covering all the ads.

We iteratively continue the assignment process and at each step we pick the (*ad, endorser*) pair that maximizes the marginal gain per cost, and re-estimate the marginal gains when they become outdated. The marginal gain computed for a pair is valid as long as the ad is not assigned a new endorser since the time the gain value was last computed. Algorithm 3 gives the pseudocode for the CEAL algorithm. For brevity, we do not show the caching of the marginal gain computations in the pseudocode.

Scaling up the algorithm. Searching for the (*ad, endorser*) pair maximizing the marginal gain per cost, $\delta_j^i(E_l)$, over all ads and all endorser sets is prohibitive in a

Algorithm 2: CEAL-post-processing($G, \mathcal{E}, \mathcal{D}, \{S_j^i\}$)

```
ctrl ← true                                              ▷ Reset loop control variable
β_i ← |R(E_i)|, ∀i ∈ [1..k]                               ▷ Reset used endorser capacities
F ← {E_l | β_l = θ_l ∧ E_l ∈ E}                          ▷ Fully assigned endorsers
while ctrl do                                            ▷ While there are still changes
    ctrl ← false                                         ▷ Reset loop control variable
    for A_j^i ∈ D in incr. order of B(A_j^i) do          ▷ Iterate over ads
        Ē ← E \ F ∪ S_j^i                                ▷ Current candidates for assignment
        for E_l ∈ S_j^i do                               ▷ Endorsers previously assigned to ad
            β_l ← β_l − 1                                ▷ Make previous assignment available
        while Ē ≠ ∅ do                                   ▷ Endorsers remain
            ▷ Compute the marginal gain for all endorsers
            δ_j^i(E_l) ← σ_G(S_j^i ∪ E_l) − σ_G(S_j^i), E_l ∈ Ē
            ▷ Find the best ad among ads with enough budget
            E_l̄ ← argmax_{E_l⊂Ē, A_j^i⊂D, c(E_l)≤B(A_j^i)} δ_j^i(E_l)
            if E_l̄ = ∅ then break                        ▷ No viable endorser
            if β_l̄ < θ_l̄ then                            ▷ Endorser has capacity
                S̄_j^i ← S̄_j^i ∪ E_l̄                      ▷ Assign endorser to ad
                B(A_j^i) ← B(A_j^i) − c(E_l̄)              ▷ Decrease ad budget
                β_l̄ ← β_l̄ + 1                            ▷ Increase used endorser capacity
            if β_l̄ = θ_l̄ then                            ▷ Endorser used all capacity
                Ē ← Ē \ E_l̄                              ▷ Remove endorser
        if σ_G(S_j^i) < σ_G(S̄_j^i) then                  ▷ A better assignment
            S_j^i ← S̄_j^i                                ▷ Use the new assignment
            ctrl ← true                                  ▷ Assignment has changed
return {S_j^i}, i ∈ [1..n], j ∈ [1..a_i]                 ▷ Return the assignments
```

large-scale setup. This is due to the cost of coverage computations on the social network. In particular, each evaluation of $\delta_j^i(E_l)$ requires estimating the coverage of a set of nodes, which in turn requires running a series of simulations. Fortunately, we can avoid estimating $\delta_j^i(E)$ for all pairs (A_j^i, E_l).

We estimate $\delta_j^i(E_l)$ for each (*ad, endorser*) pair during initialization (iteration t_0). But luckily when all nodes are empty, we have $\delta_j^i(E_l)$=$\sigma(E_l), \forall_l$ and thus coverage computation is performed only as many times as there are endorsers. We use a priority queue Q over endorsers, where the value associated with an endorser E_l is taken as the maximum of the marginal gain per cost values over the ads, that is $\max_{i,j} \delta_j^i(E)$. At

Algorithm 3: CEAL

$\{S_j^i\} \leftarrow$ CEAL-initial-assignment$(\mathcal{G}, \mathcal{E}, \mathcal{D})$ $\{\bar{S}_j^i\} \leftarrow$ CEAL-post-processing$(\mathcal{G}, \mathcal{E}, \mathcal{D}, \{S_j^i\})$

iteration t_0, we use the endorser at the top of the priority queue Q and assign it to the ad that provides the maximum marginal gain.

At iteration $t_0 + 1$, we need to find another endorser having maximum marginal gain per cost. Instead of re-estimating marginal gains for each pair, we first re-estimate the maximum marginal gain of the endorser at the top of the queue. If the resulting value is still the highest one in priority queue, then we can use the endorser that is currently at the top of the priority queue without further marginal gain computations. This is because coverage is submodular, and thus the priority queue contains upper bounds for all endorsers at any given time. Updating the values for other endorsers can never result in an increase.

For any step t, we can summarize the complete process as follows: If the marginal gain per cost value of the endorser at the top of the priority queue is up-to-date, use the endorser as the next one to assign to an ad. Otherwise, recompute its marginal gain and readjust its location in the priority queue. Continue until the endorser at the top of the queue has an up-to-date marginal gain value.

As an additional optimization, we can further reduce the number of coverage evaluations by limiting the number of marginal gain computations performed when an update is performed for an endorser E that is currently at the top of the priority queue Q. Normally, such an update requires computing marginal gains for each ad. This can be avoided by keeping a priority queue $Q(E)$ over ads for each endorser E. We recompute the marginal gain for the ad at the top of the priority queue $Q(E)$ and shortcut the computations of further marginal gains if the same ad is still at the top of the queue after the re-computation. Again, this is possible due to the submodularity, which means that the values can only get smaller and thus the $Q(E)$ always contains the upper bounds, just like for the global priority queue Q before.

Boundness of the algorithm. Unfortunately, using only cost effective marginal gain may result in arbitrarily large error. Consider a case where there are two ads A_1 and A_2 with unit budgets, and four endorsers E_1, E_2, E_3, and E_4 having coverages

$\epsilon, \epsilon, 1-\epsilon, 1-\epsilon$ and costs $\epsilon, \epsilon, 1, 1$, respectively. Assume that the coverage sets of the endorsers do not intersect. Then the CEAL algorithm will assign E_1 to A_1 and E_2 to A_2. While E_1 and E_2 have higher marginal gain per cost, they leave a significant portion of the budget unused. Yet this budget is not sufficient to accommodate an additional endorser. In this particular case, the optimal assignment would be to assign E_3 and E_4 to the two ads. We can generalize this situation to arbitrarily large number of ads and endorsers.

We handle this problem by using a post-processing step. After the initial assignments of the endorsers to ads is complete, we consider alternative assignments for each ad. Let $F \subset \mathcal{E}$ be the set of endorsers that the initial assignment phase of CEAL used to their limits, i.e., $\forall E_l \in F, \beta_l = \theta_l$. We start with the ad having the smallest budget and aim to locate the set of endorsers $S \subset \{\mathcal{E} - F \cup E_l\}$ that maximizes the coverage *without considering the endorser costs*. In other words, we have $S = \underset{S \subset \{\mathcal{E}-F \cup E_l\}}{\operatorname{argmax}} \sum_{e \in E} \sigma(e)$. For this purpose, we use the same greedy procedure from earlier, but this time the marginal gains in coverage are not divided by the endorser costs. Thus, we iteratively assign endorsers that provide the highest marginal gains and are not violating the remaining budget of the ad at hand. Let $\bar{S}$ be the resulting set of assignments for the ad. If the coverage $\sigma(\bar{S})$ of this new assignment is larger than that of the original coverage that was provided by the initial phase of the CEAL algorithm, then we replace the assignments for the ad at hand with $\bar{S}$ and adjust the endorser capacities. Otherwise, we continue with the initial assignments for the ad. We continue this process by considering other ads, in increasing order of ad budget and potentially performing multiple scans, until the assignments do not change for any of the ads.

This post-processing step of CEAL is given in Algorithm 2 and the complete CEAL algorithm, which consists of the initial and post-processing steps is given in Algorithm 3.

3.5 An Upper Bound on Coverage

Finding the exact solution to broker-based ad allocation problem requires considering exponential number of assignments of endorsers to ads. This is infeasible even for a small number of ads and endorsers. We show that under certain circumstances, an upper bound on the accuracy of the CEAL algorithm can be provided. Our bound is based on the following theorem [38]:

Theorem 7 *Let $\rho_j = \sigma(S \cup \{j\}) - \sigma(S)$, then the following statement defines a submodular set function σ:*

$$\sigma(T) \leq \sigma(S) + \sum_{j \in T-S} \rho_j(S) \tag{3.11}$$

If we let $S = \emptyset$, then the theorem suggests that the coverage of any set of nodes is smaller than the sum of the coverages of any partitioning of it. Let us assume that the total cost $c(E)$ of endorser E is linear on the coverage it provides, i.e., $c(E) = \alpha\,\sigma(E) + \beta$. Thus the total cost of a set of endorsers S becomes $\sum_{E \in S} c(E) = \alpha \sum_{E \in S} \sigma(E) + |S|\,\beta$. And the total coverage of the set can be expressed using the total cost of the set:

$$\sum_{E \in S} \sigma(E) = \frac{1}{\alpha} \left(\sum_{E \in S} c(E) - |S|\,\beta \right) \tag{3.12}$$

Based on this result and the theorem above, we can give an upper bound on total coverage as follows:

Corollary 1 *Given $c(E) = \alpha\,\sigma(E) + \beta$*

$$\sigma(S) \leq \frac{1}{\alpha} \left(\sum_{E \in S} c(E) - |S|\,\beta \right) \tag{3.13}$$

Then, the total coverage of assignments is bounded from above by the following:

$$\sum_{i,j} \sigma(S_j^i) \leq \frac{1}{\alpha} \sum_{i,j} \left(\sum_{E \in S_j^i} c(E) - |S_j^i|\,\beta \right) \tag{3.14}$$

Chapter 4

Evaluation

We conducted experiments using the ICMO model and the CEAL algorithm. We used three large, real-world datasets for our evaluation and experiment with different workload parameters.

We first present our results on the proposed ICMO model. In these experiments, we investigated influence maximization under our ICMO model, which factors in the potential overloading of the users. We evaluated the performance of the greedy algorithm and other heuristics for influence maximization using ICMO.

We then focus on illustrating the performance of the CEAL algorithm by investigating the following aspects:

1. total coverage provided compared to a baseline algorithm as well as to an upper-bound,
2. total coverage as a function of the skew in the polynomial cost function used for endorsers,
3. the number of endorsers assigned work (fairness to endorsers),
4. the total budgets of companies that are be filled (profitability for the broker),
5. the variation in the cost charged per coverage provided for the ads (fairness to companies/ads).

4.1 Experimental Setup

Datasets. We used three different real-world social network datasets that are publicly available [42, 43].

• **WikiVote Dataset**: The network contains the Wikipedia voting data till Jan 2008. Nodes in the network represent Wikipedia users. A directed edge from i to j denotes that the user i has voted on user j. Thus the reverse edge represents the flow of influence from user j to user i. The network contains 7,115 nodes and 103,689 edges.

• **Epinions Dataset**: This is the who-trusts-whom network of users from the `epinions.com` website. Nodes represent users. A directed edge from i to j in the network represents the trust of user i to user j. Thus the reverse edge represents the influence from user j to user i. There are 75,879 nodes and 508,837 edges.

• **Facebook Dataset**: This is the Facebook friendship network. Nodes in the network represent users and edges represents relationships between user pairs. The original network is undirected, but we convert it into a bidirectional one, where directed edges exist in both directions. There are 63,731 nodes and 1,269,502 edges.

The various workload features are generated as follows:

Propagation probabilities. We used the weighted cascade model [1] to assign propagation probabilities in ICMO. In this model, $p_u(v) = 1/d(v)$, where $d(v)$ is the in-degree of a node v. Thus the propagation probability of each edge is determined by the number of incoming edges of the destination node.

Overload probabilities. We used a trivalency-based approach to generate the overload probabilities of users. Intuitively, a user gets overloaded easier than she gets activated. It may take a few consecutive messages to overload a user, while this may be hardly enough to catch the interest of the user. Thus we used relatively larger probabilities to capture this phenomenon. We selected uniformly at random a probability from $\{0.1, 0.3, 0.5\}$ to represent different levels of overload tendency for users.

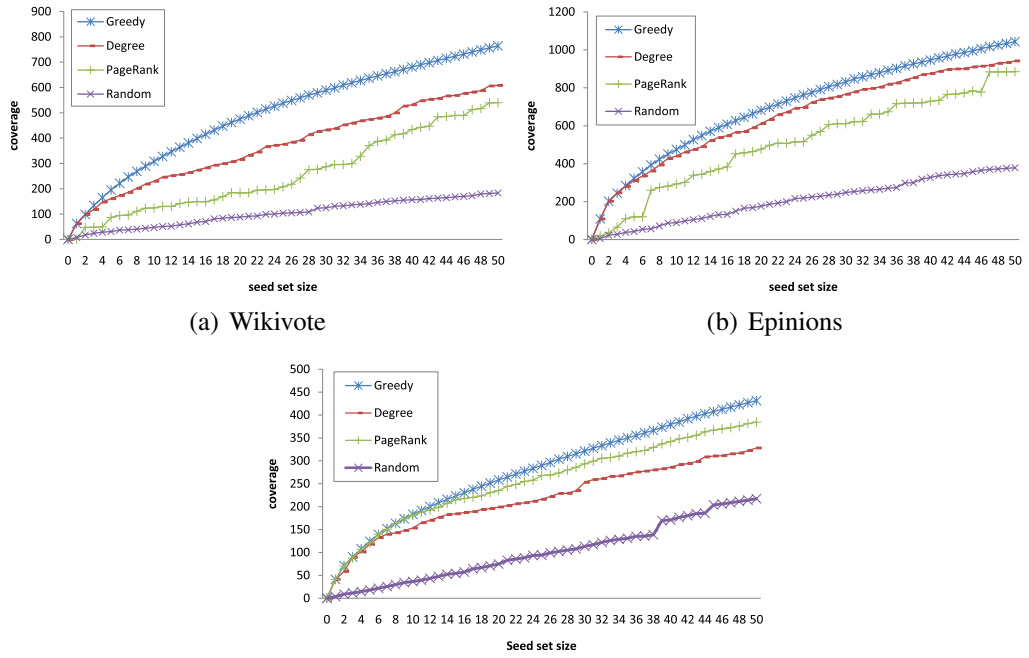
Endorser selection. We selected 500 users as endorsers from each network separately, where the total influence of these users under ICMO model is maximized. We used the same 500 endorsers throughout the experiments and vary the number of companies.

Endorser limits. Since we do not know the exact limits for endorsers that cause them

to get overburdened or perceived as spammers. We selected uniformly at random the limits from the list $\{1, 2, 3, 4, 5\}$ for each endorser.

Endorser costs. To model different costs for endorsers, we used a polynomial $f(x)$ of order α where $0 < \alpha \leq 1$. We used the individual coverages of endorsers as inputs to f and obtained different cost models by changing α, aka the *endorser cost skew parameter*.

Ad budgets. We generated at least one endorser whose cost is less than the budget of an ad. We iteratively picked a random sample from a power law distribution with an exponent of -0.9 and exponential cutoff of *maxcost* $= 0.5$ and added *mincost* $= 0.3$ to it for each ad.

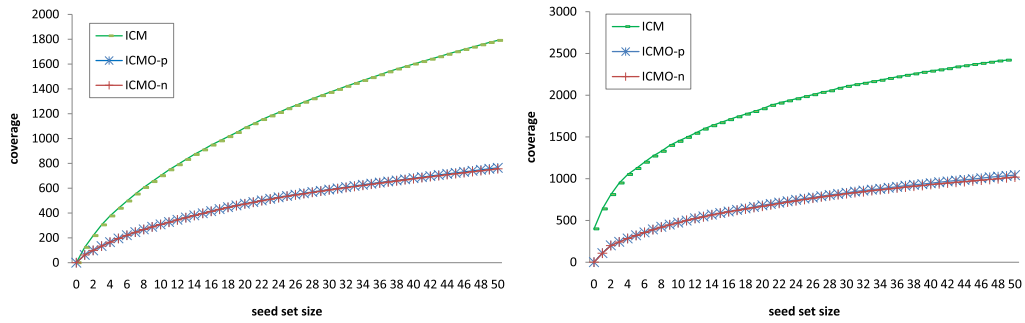


(a) Wikivote

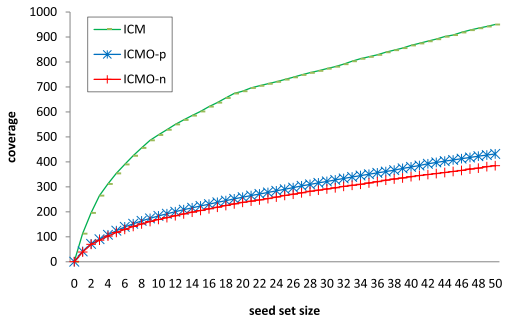
(b) Epinions

(c) Facebook

Figure 4.1: Coverage under the progressive ICMO model



(a) Wikivote (b) Epinions



(c) Facebook

Figure 4.2: Coverage under ICM, progressive and non-progressive ICMO

4.2 Experimental Results

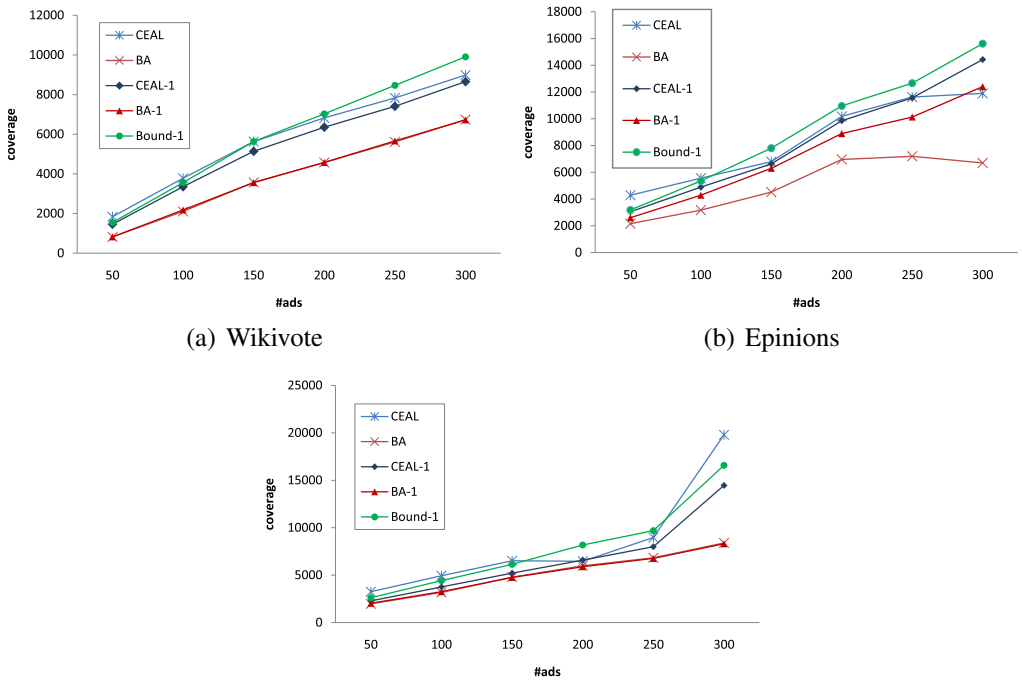
4.2.1 Influence Maximization with ICMO

We compared several centrality-based measures as well as greedy algorithms to investigate the coverage under our ICMO propagation model. We used the following heuristics in our experiments:

• **Greedy**: The greedy algorithm proposed in [1] along with lazy-forward tuning [11]. For each iteration, we ran 10,000 simulations to estimate the coverage. We picked the m nodes that provide the best total coverage based on the greedy selection procedure.

• **PageRank**: The web page ranking algorithm proposed in [44]. We used 0.1 as the restart probability and 0.001 as the stopping margin. We picked the m nodes with the highest PageRank values.

• **Highest Degree**: Degree of a node represents its popularity. We picked the m



(a) Wikivote

(b) Epinions

(c) Facebook

Figure 4.3: Total coverage as number of ads increases

nodes that have the highest probabilities.

• **Random**: We uniformly at random picked m seeds from the network. We did not pick nodes that have 0 out-degree.

We generated subgraphs of size 10,000 and 7,000 respectively by uniform sampling, and used whole Wikivote network to evaluate the influence under ICMO. We generated a seed set with a maximum size of 50. We iteratively increased the size of the seed set by adding the node that has the highest score for the current heuristic.

Figure 4.1 illustrates the results of the algorithms on the three datasets. They plot the coverage achieved as a function of the seed set size. For all three datasets, the greedy algorithm performs better than the others. This is expected as all other heuristics choose clustered seeds whose coverage sets might be overlapping. Thus the marginal profits of seeds in terms of coverage degrades. We expect PageRank and Degree heuristics to perform worse with ICMO. This is because if seeds are clustered, then every node that is close to the clustered seeds will possibly get the same message multiple times. Thus, the overload likelihood of these nodes will increase.

In the Wikivote dataset, the difference between the Greedy algorithm and other heuristics is larger. This dataset is denser than the other two. So it is more likely for the nodes in clustered regions to receive the same message multiple times. This is the reason for the large gap between the Greedy algorithm and other heuristics, compared to other datasets. In the Epinions and Facebook datasets, PageRank and Degree heuristics are closer to the Greedy algorithm. The reason is that, although these heuristics may pick clustered seeds, a node in the clustered regions will receive the same message only a few times because the density of the graphs is lower.

Additionally, we examined the increase in the number of overloaded users as the number of seeds increase. We observed that the increase in the number of overloaded users slows down as the seed set size increases. The reason is that, as users get overloaded, the number of messages that non-overloaded users can receive starts to decrease, which also decreases the probability that additional users get overloaded.

We also estimated the coverage *without* the overloading effect. We ran greedy algorithm to select 50 seeds. As expected, the cascade of influence is overestimated by ICM. On average, the coverage under ICM is 2.34, 2.09, and 2.20 times larger than the coverage under ICMO for Wikivote, Epinions, and Facebook datasets, respectively. This emphasizes the importance of overload effect while estimating the coverage. Also under ICM the coverage has a more longstanding increase compared to under ICMO. Figure 4.2 illustrates the results.

To study the impact of non-progressive ICMO, we applied the greedy algorithm under this model to choose 50 seeds. Intuitively, we expect the coverage under non-progressive ICMO to be significantly smaller than under the progressive model. Surprisingly, the coverage decreases by an insignificant amount. We can explain this phenomenon by the sparseness of the influence propagation. In general, the coverage over a network is sparse and the spread of influence has a fast degrading effect as users get distant from the seed. Considering that the greedy algorithm picks seeds that have small coverage overlap with previously chosen seeds and relatively high coverage, and that users that will receive the same message multiple times are likely to be close to several seeds, such users are not common. Importantly, they are a very small portion of the activated users, and thus the decrease due to the non-progressive model is small.

The small decrease is caused by overloading of active users that are locally central and close to one of the seed users. Thus the user will receive the same message several times which is initiated by the same seed user.

Figure 4.2 illustrates the results. ICMO-p denotes the progressive ICMO and ICMO-n denotes the nonprogressive ICMO. At 50 seeds, the coverage under progressive ICMO is 1%, 2%, and 12% more than the coverage under the non-progressive model for the Wikivote, Epinions, and Facebook datasets, respectively.

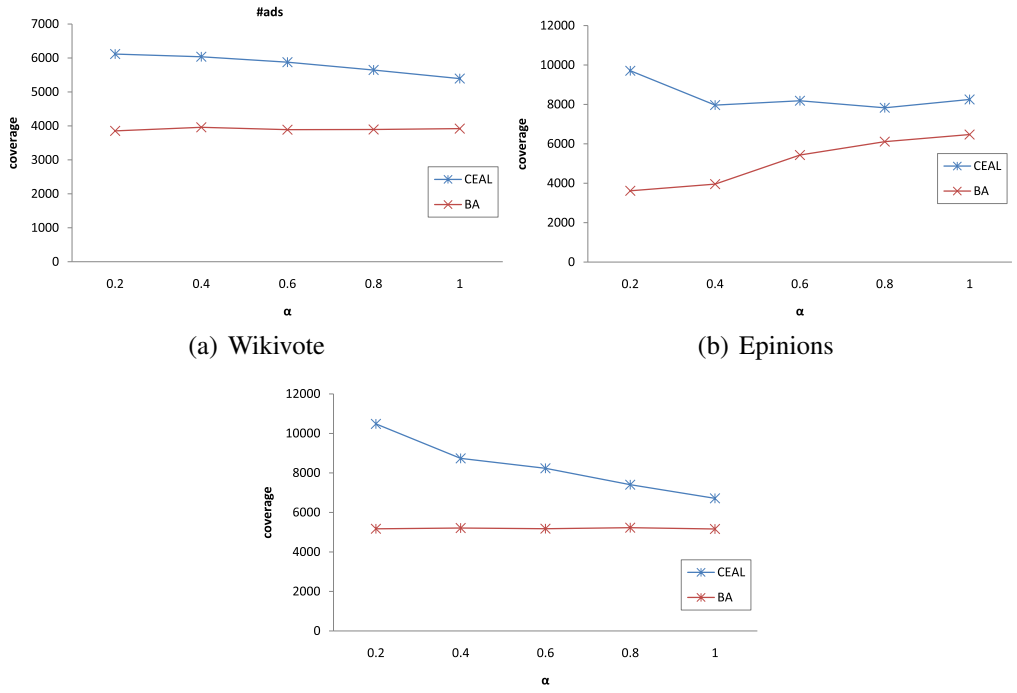
In summary, the coverage under ICMO using the greedy algorithm is significantly smaller than the coverage under ICM. The reason is that users get overloaded as they receive the same ad message several times. This is more realistic since users tend to get overloaded easily than they get influenced. Also, other heuristics suffer more under ICMO than under ICM, because they select seeds that are likely to be clustered which is the main reason why coverage under ICMO decreases compared to ICM. This effect is more clear when coverage is measured under denser graphs. As the density of the graph increases, the users within the clustered regions of seed users will receive the same message multiple times. Finally, we observe that the negative impact of non-progressive ICMO on the coverage is insignificant.

4.2.2 Coverage in Broker-based Ad Allocation

We evaluatde the effectiveness of the CEAL algorithm by comparing its coverage with that of a base algorithm (BA). The base algorithm iteratively selects an (*ad, endorser*) pair at uniform random, where the budget constraints of the ad is not violated, the endorser has capacity for an ad assignment, and the ad was not already assigned to the same endorser. Once a random suitable pair is located, the assignment of the endorser to the ad is made.

Figure 4.3 illustrates the results for total coverage. In particular, it plots the total coverage achieved as a function of the number of ads, for the CEAL and BA algorithms.

We observed a near linear relationship between the number of ads and the total



(a) Wikivote

(b) Epinions

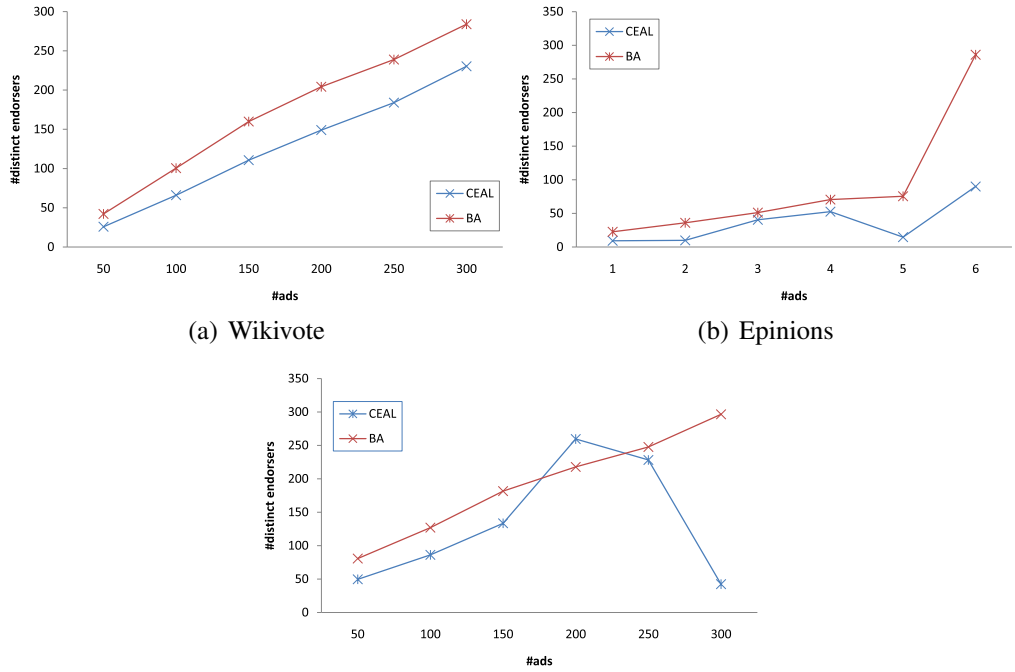
(c) Facebook

Figure 4.4: Total coverage as α increases

coverage provided by the CEAL and BA algorithms. The reason is that the companies in the tail part of the budget distribution are assigned with a few endorsers (very few overlaps exist among the coverage of these endorsers). Thus the submodular coverage function behaves like linear in the tail. And the total coverage of the endorsers assigned to the companies in the tail has almost half of the total coverage. As a result, total coverage linearly increases as the number of ads increases.

The BA algorithm can pick endorsers that have low gain/cost ratio, since it is randomized. Furthermore, some ads that may have higher marginal gains for an endorser may not have enough remaining budget due to earlier assignments. Meanwhile, CEAL picks an (*ad*, *endorser*) pair that is locally optimal. It searches towards the global optimum by making local optimal decisions. At worst, we expect to find a local optimal result. Consequently, CEAL algorithm has higher coverage compared to BA. On average, CEAL improves the total coverage relative to BA by 37%, 68%, and 32% for the Wikivote, Epinions, and Facebook datasets, respectively.

Figure 4.3 also presents the upper-bounds on the coverage results. The upper bound



(a) Wikivote

(b) Epinions

(c) Facebook

Figure 4.5: Number of different endorsers that are assigned to an ad

in Chapter 3 requires linear cost functions, so we provide the upper-bounds only for the case of $\alpha = 1$. CEAL-1, BA-1, and Bound-1 are the coverage results on CEAL and BA algorithms, respectively, for $\alpha = 1$. As the number of endorsers assigned to an ad increases, the difference between the upper bound and the optimal result also increases. This fact comes from the submodularity of the coverage function. So as the number of ads increases, we expect the difference between the upper bound and CEAL to increase. We can observe that when the number of companies is 50, CEAL is almost optimal for the three datasets. The difference from the upper bound is smaller than 6%, 4%, and 2% for 50 companies. But as the number of companies increases, the difference grows up to 13%, 15%, and 20%, respectively. On average, CEAL is within 91%, 91%, and 84% of the upper bound and considering that the optimal lies between the upper bound and CEAL, we conclude that CEAL is very close to the optimal.

4.2.3 Impact of the Endorsers Cost Model

Intuitively, the higher the coverage of an endorser, the lower the increase in her per ad cost as new users are added to the coverage set. For example, let us assume that one endorser has 10K coverage, while another has 1000K. If they start to influence an additional 5K, then it is a noteworthy increase for the first endorser. And the total cost of the endorser will probably increase a lot. But this change may not cause an easily observable increase in the total cost for the second endorser. To study how CEAL and BA behave and how total coverage changes under different cost models, we assign the costs to endorsers using the following function $f(\sigma(E_i)) = (\sigma(E_i))^{\alpha}$, where $E_i \in \mathcal{E}$.

Figure 4.4 presents the relationship between the total coverage of the algorithms and the value of α. For the Wikivote dataset, as α increases, there is a slight decrease in the coverage for CEAL, whereas the coverage is stationary for BA. On the other hand, for the Epinions dataset, we observe an increase in the coverage for BA. As α increases, the cost distribution of endorsers approaches a power law distribution. And the coverage/cost ratio of each endorser is the same. As we assign endorsers to ads, the marginal gain of endorsers slightly degrades and CEAL starts to pick high coverage (and thus high cost) endorsers. But as α decreases, the cost distribution of endorsers approaches a uniform distribution and BA starts to pick high cost endorsers that have low coverage.

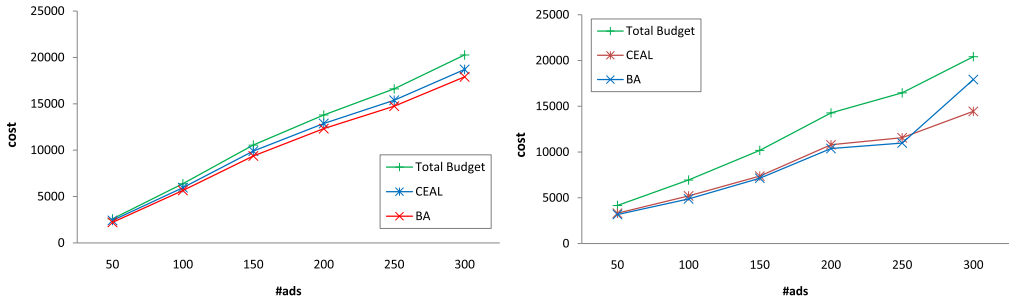
For all datasets and for all α values, the CEAL algorithm outperforms the BA algorithm. On average, CEAL improves total coverage compared to BA by 32%, 38%, and 36% for the Wikivote, Epinions, and Facebook datasets, respectively.

4.2.4 Number of distinct endorsers used

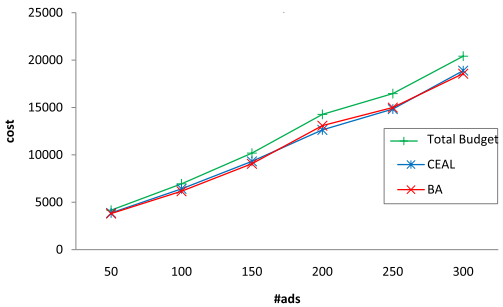
All other things being equal, such as the total coverage provided and the total company budgets used up, a broker that uses less endorsers is desirable to one that uses more, as it will reduce operational overheads. We compared the number of distinct endorsers used by the CEAL algorithm to that of BA.

Figure 4.5 plots the number of distinct endorsers used as a function of the number of ads. We observed that for both of the datasets, CEAL uses less number of distinct endorsers than BA. In particular, BA uses 1.40, 2.13, and 2.23 times more distinct endorsers for the Wikivote, Epinions, and Facebook datasets, respectively.

To understand the intuition behind this, consider the working of the CEAL algorithm. Before any assignments are made, the marginal gain of each endorser will be equal to her coverage. Let us assume that CEAL has assigned the endorser $E_l \in \mathcal{E}$ to the ad A. If the endorser has additional capacity ($\theta_l > 1$) and there exists other ads that have enough budget to accommodate the ad ($\exists A' \neq A \in \mathcal{D}$ s.t. $B(A') \geq c(E)$), then the endorser E_l will still have the highest gain. Then the algorithm will again pick E_l. Similar steps are taken later in the execution of the algorithm as well, as long as the coverage of the best endorser has limited intersection with the coverage sets of viable ads. This results in using the capacities of some endorsers in full, leaving some endorsers with empty assignments. In our broker model, this is not necessarily a problem. In fact, it is an advantage when same or better coverage can be provided by less number of endorsers.

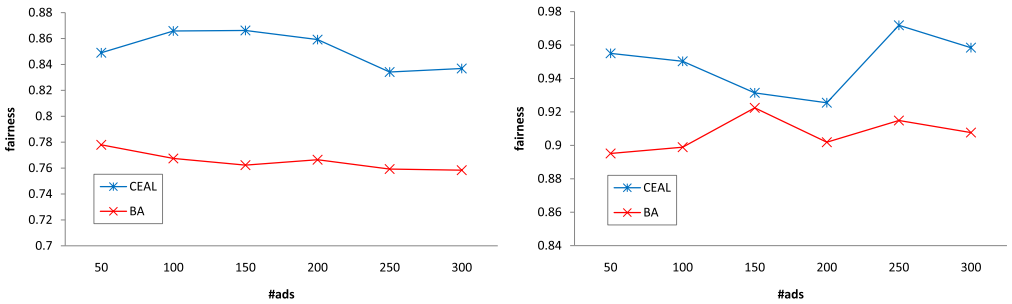


(a) Wikivote (b) Epinions

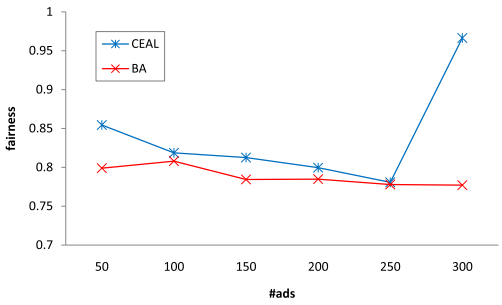


(c) Facebook

Figure 4.6: Total pay and total budget as number of companies increases



(a) Wikivote (b) Epinions



(c) Facebook

Figure 4.7: Fairness to the companies

4.2.5 Total company budgets used

Total company budgets used is another important metric, which also determines how much profit the broker can make. This can be computed as the total cost of the endorsers assigned to ads times a fixed percentage, aka the *profit margin*.

Figure 4.6 plots the total budget of the companies and the total payment made for the ads as the number of ads increases. We observed that for all datasets, most of the budget is used, irrespective of the number of ads. As the coverage of endorsers is power law distributed, we have a lot of low cost endorsers. Thus we can use almost all the budgets of the ads. On average CEAL uses 93%, 74%, and 91% of the budgets of ads for the Wikivote, Epinions, and Facebook datasets, respectively.

4.2.6 Fairness to companies

To measure the fairness of CEAL and BA to companies, we calculate their *unit payment*s: the amount of payment companies make (their used budget) per unit coverage we provide to them. A naïve comparison of unit payment will be biased, because the unit payments of small budgeted ads are considerably smaller than the unit payments of larger budgeted ads. This is due to the submodularity of coverage. As we assign more and more endorsers to an ad, the marginal gain per cost of the subsequent assignments will drop.

To provide a reasonable comparison with respect to the fairness metric, we divided ads into segments according to their budgets (full budgets before the assignments are made) and compared ads within their own segment. We first sorted the ads in decreasing order of their budgets and created a new segment with the ad having the highest budget. We iterated over remaining ads and put the ads into the current segment if the difference between the maximum budget in the segment and the budget of the current ad is within h times the overall standard deviation, where h is used to model the homogeneity of the segments. If the budget of the ad exceeds this value, then we created a new segment and put the ad into it. We continued the process until we put all the ads into a segment. We took the weighted average of standard deviation of unit payments for the segments as our fairness metric.

Figure 4.7 plots the fairness metric as a function of the number of ads. Since the fairness metric depends more on the distribution of costs and budgets than the number of ads, there is not a pronounced pattern between fairness and number of ads. In all three datasets, CEAL provides higher fairness than BA. On average, CEAL improves fairness by 11%, 4%, and 6% compared to BA for the Wikivote, Epinions, and Facebook datasets, respectively.

Chapter 5

Conclusion and Future Work

We formulated the problem of broker-based ad allocation in social networks. We modeled the problem through a combinatorial optimization framework and proposed a cost-effective solution. We presented a novel information propagation model that can capture the case of users getting overloaded due to too many messages.

The ICMO propagation model we proposed makes a distinction between a user getting interested and a user getting active. A user becomes interested if she gets influenced by the content of a sent message. A user becomes active if she is interested and the number of messages she received is not too high to cause her to get overloaded. We show that influence under ICMO is more sparse than under general ICM. We also show that several heuristics techniques used for influence maximization have different characteristics on ICMO than they have on ICM. In particular, choosing clustered seeds has a more pronounced degrading effect on the coverage with ICMO, since it increases the likelihood of users getting overloaded.

The greedy CEAL algorithm we developed to perform broker-based ad allocation iteratively searches the locally optimal and viable (in terms of ad budget and endorser capacity) (*ad, endorser*) pair to perform an assignment. At each iteration, CEAL picks the pair that has the highest cost-effective marginal gain. After CEAL assigns endorsers to ads using cost effective marginal gain, it enters a post-processing phase to solve unboundness issues that may result from the use of cost effective gain. During

post-processing, CEAL iteratively picks an ad and assigns endorsers using marginal gain without cost effects. If the newly assigned endorsers have higher coverage than the previous assignments, the new assignments are used. CEAL continues the post-processing step until no further improvements are possible.

We measured the coverage provided by CEAL with varying number of ads and different cost functions, using real-world social networks. Experimental results showed that the algorithm is close to optimal. CEAL is within 91%, 91%, and 84% of the upper bound for the three different datasets we experimented with. We compared it with a baseline algorithm and on average CEAL improves the total coverage by 37%, 68%, and 32% for the three datasets. We showed that under different cost functions for endorsers, total coverage differs. In particular, when the increase in the cost of the endorsers slows down as the coverage of the endorser increases (which models the real-world behavior), then the CEAL algorithm provides even better coverage relative to the baseline. CEAL also consumes close to the entire budgets of the companies and it is fair to the companies with respect to the cost charged per coverage provided for similar budgeted ads.

In the following, we discuss a number of directions for extending this work.

Ad Allocation with User Interests. In a social network environment, the interests of users are different from each other. Users tend to follow others having interests similar to their own. This eventually forms the underlying topology of the network and affects the influence users have over each other. Also, users have different levels of authority on different topics and influence others at different levels based on their expertise. Furhermore, each ad has a dedicated content and appeals only to a subset of users. Consequently, it is more natural and desirable to assign ads to endorsers considering the authority of the endorsers and the interests of the users in their area of influence.

Ad Allocation with Competitive Ads. Consider a scenario where there are two companies from the same market segment and both willing to make a viral advertisement campaign using our system. We need to assign these ads to endorsers such that both receive high coverage, yet none of them die out or get adversely affected because

of the negative effect of the other. Assigning these ads to endorsers in the same region will reduce the effectiveness of the ads and will give unsatisfactory results. The assignment algorithm should take this into account as well.

Ad Allocation with Targeted Endorser Selection. Endorsers are enrolling our system voluntarily. But a targeted selection of endorsers may increase the effectiveness of the assignments. Consider the case where there are bottleneck users which are highly inactive and the propagation of ads cease when these users are reached. On the other hand, if the bottleneck users are activated, the propagation process accelerates significantly. By paying a relatively higher amount of money to these users and attracting them to the system may increase the success of the system.

Ad Allocation with Community Structure. Users with similar interest tend to cluster together. Also, acquaintances have relatively close circles. These communities have distinct behaviors considering the whole network. A targeted selection of communities for assignment of an ad may increase the accuracy and reduce the search space of the algorithm. Also, core users which generally generate ideas and peripheral users that adopt ideas may be targeted more extensively, instead of targeting arbitrary users.

Bibliography

[1] D. Kempe and J. K. Éva Tardos, "Maximizing the spread of influence through a social network," in *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 137–146, 2003.

[2] E. M. Rogers, *Diffusion of Innovations, 5th Edition*. Free Press, 5th ed., Aug. 2003.

[3] P. Domingos and M. Richardson, "Mining the network value of customers," in *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 57–66, 2001.

[4] M. E. J. Newman, "Spread of epidemic disease on networks," *Physical Review E*, vol. 66, pp. 016128+, July 2002.

[5] R. Kumar, J. Novak, P. Raghavan, and A. Tomkins, "On the bursty evolution of blogspace," in *Proceedings of the 12th international conference on World Wide Web*, WWW '03, (New York, NY, USA), pp. 568–576, ACM, 2003.

[6] M. Granovetter, "Threshold models of collective behavior," *The American Journal of Sociology*, vol. 83, no. 6, pp. 1420–1443, 1978.

[7] 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*, pp. 211–223, Aug. 2001.

[8] D. Kempe, J. Kleinberg, and Éva Tardos, "Influential nodes in a diffusion model for social networks," in *International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 1127–1138, 2005.

[9] A. Goyal, F. Bonchi, and L. V. S. Lakshmanan, "A data-based approach to social influence maximization," *VLDB Journal*, vol. 5, no. 1, pp. 73–84, 2011.

[10] 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*, vol. 3, no. 2, pp. 179–192, 2013.

[11] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen, and N. Glance, "Cost-effective outbreak detection in networks," in *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 420–429, 2007.

[12] S. Bharathi, D. Kempe, and M. Salek, "Competitive influence maximization in social networks," in *International Conference on Internet and Network Economics*, pp. 306–311, 2007.

[13] Y. Li, W. Chen, Y. Wang, and Z.-L. Zhang, "Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships," in *ACM International Conference on Web Search and Data mining*, pp. 657–666, 2013.

[14] W. Chen, A. C. 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," Tech. Rep. MSR-TR-2010-137, Microsoft Research, 2011.

[15] S. Bhagat, A. Goyal, and L. V. S. Lakshmanan, "Maximizing product adoption in social networks," in *ACM International Conference on Web Search and Data mining*, pp. 603–612, 2012.

[16] N. Barbieri, F. B. Bonchi, and G. Manco, "Topic-aware social influence propagation models," in *IEEE International Conference on Data Mining (ICDM)*, pp. 81–90, 2012.

[17] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "TwitterRank: Finding topic-sensitive influential twitterers," in *ACM International Conference on Web Search and Data mining*, pp. 261–270, 2010.

[18] L. Liu, J. Tang, J. Han, M. Jiang, and S. Yang, "Mining topic-level influence in heterogeneous networks," in *ACM International Conference on Information and Knowledge Management (CIKM)*, pp. 199–208, 2010.

[19] Y. Zhang, J. Zhou, and J. Cheng, "Preference-based top-k influential nodes mining in social networks," in *International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, pp. 1512–1518, 2011.

[20] W. Chen, C. Wang, and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks," in *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 1029–1038, 2010.

[21] W. Chen, Y. Yuan, and L. Zhang, "Scalable influence maximization in social networks under the linear threshold model," in *IEEE International Conference on Data Mining (ICDM)*, pp. 88–97, 2010.

[22] C. Wang, W. Chen, and Y. Wang, "Scalable influence maximization for independent cascade model in large-scale social networks," *Data Mining and Knowledge Discovery*, vol. 25, no. 3, pp. 545–576, 2012.

[23] W. Chen, Y. Wang, and S. Yang, "Efficient influence maximization in social networks," in *ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, pp. 199–208, 2009.

[24] G. Mega, A. Montresor, and G. Picco, "Efficient dissemination in decentralized social networks," in *IEEE International Conference on Peer-to-Peer Computing (P2P)*, pp. 338–347, 2011.

[25] J. Cheng, A. Sun, and D. Zeng, "Information overload and viral marketing: countermeasures and strategies," in *International Conference on Social Computing, Behavioral Modeling, and Prediction (SBP)*, pp. 108–117, 2010.

[26] B. Lehmann, D. Lehmann, and N. Nisan, "Combinatorial auctions with decreasing marginal utilities," in *ACM Conference on Electronic Commerce (EC)*, pp. 18–28, 2001.

[27] J. Vondrak, "Optimal approximation for the submodular welfare problem in the value oracle model," in *Annual ACM Symposium on Theory of Computing (STOC)*, pp. 67–74, 2008.

[28] V. Mirrokni, M. Schapira, and J. Vondrak, "Tight information-theoretic lower bounds for welfare maximization in combinatorial auctions," in *ACM Conference on Electronic Commerce (EC)*, pp. 70–77, 2008.

[29] U. Feige and J. Vondrak, "Approximation algorithms for allocation problems: Improving the factor of 1 - 1/e," in *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 667–676, 2006.

[30] S. Dobzinski and M. Schapira, "An improved approximation algorithm for combinatorial auctions with submodular bidders," in *Annual ACM-SIAM Symposium on Discrete Algorithm (SODA)*, pp. 1064–1073, 2006.

[31] S. Khot, R. J. Lipton, E. Markakis, and A. Mehta, "Inapproximability results for combinatorial auctions with submodular utility functions," *Algorithmica*, vol. 52, no. 1, pp. 3–18, 2008.

[32] M. Kapralov, I. Post, and J. Vondrák, "Online submodular welfare maximization: Greedy is optimal," in *Annual ACM-SIAM Symposium on Discrete Algorithm (SODA)*, pp. 1216–1225, 2013.

[33] A. Srinivasan, "Budgeted allocations in the full-information setting," in *International Workshop on Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques (APPROX)*, pp. 247–253, 2008.

[34] Y. Azar, B. Birnbaum, A. R. Karlin, C. Mathieu, and C. T. Nguyen, "Improved approximation algorithms for budgeted allocations," in *International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 186–197, 2008.

[35] J. Feldman, N. Korula, V. Mirrokni, S. Muthukrishnan, and M. Pál, "Online ad assignment with free disposal," in *International Conference on Internet and Network Economics*, pp. 374–385, 2009.

[36] J. Feldman, M. Henzinger, N. Korula, V. S. Mirrokni, and C. Stein, "Online stochastic packing applied to display ad allocation," in *Annual European conference on Algorithms (ESA)*, pp. 182–194, 2010.

[37] J. Feldman, A. Mehta, V. Mirrokni, and S. Muthukrishnan, "Offline optimization for online ad allocation (extended abstract)," in *Proceedings of the Ad Auctions Workshop*, 2009.

[38] G. Nemhauser and L. Wolsey, "Maximizing submodular set functions: Formulations and analysis of algorithms," *Annals of Discrete Mathematics - Studies on Graphs and Discrete Programming*, vol. 11, pp. 279–301, 1982.

[39] M. Kapralov, I. Post, and J. Vondrák, "Online and stochastic variants of welfare maximization," *Computing Research Repository (CoRR)*, vol. abs/1204.1025, 2012.

[40] J. Vondrak, "Optimal approximation for the submodular welfare problem in the value oracle model," in *Annual ACM Symposium on Theory of Computing (STOC)*, pp. 67–74, 2008.

[41] M. Feldman, J. Naor, and R. Schwartz, "A unified continuous greedy algorithm for submodular maximization," in *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 570–579, 2011.

[42] J. Leskovec, "Stanford network analysis platform." `http://memetracker.org/data/index.html`, July 2013.

[43] B. Viswanath, A. Mislove, M. Cha, and K. P. Gummadi, "On the evolution of user interaction in Facebook," in *ACM SIGCOMM Workshop on Social Networks (WOSN)*, pp. 37–42, 2009.

[44] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," in *International Conference on World Wide Web (WWW)*, pp. 107–117, 1998.