Kennesaw State University
DigitalCommons@Kennesaw State University

Dissertations, Theses and Capstone Projects

5-2015

Blocking Negative Influential Node Set in Social Networks: From Host Perspective

Harneet Kaur
Kennesaw State University

BLOCKING NEGATIVE INFLUENTIAL NODE SET IN SOCIAL NETWORKS: FROM HOST PERSPECTIVE

A Thesis
Presented to
The Academic Faculty

by

Harneet Kaur

In Partial Fulfillment
of the Requirements for the Degree
MSCS in the
Department of Computer Science

Kennesaw State University
April 2015

BLOCKING NEGATIVE INFLUENTIAL NODE SET IN SOCIAL NETWORKS: FROM HOST PERSPECTIVE

Committee Members:

Dr. Jing (Selena) He,
Committee Chair
Department of Computer Science
Kennesaw State University

Dr. Hisham Haddad
Department of Computer Science
Kennesaw State University

Dr. Ken Hoganson
Department of Computer Science
Kennesaw State University

Dr. Ying Xie
Department of Computer Science
Kennesaw State University

Dr. Yong Shi
Department of Computer Science
Kennesaw State University

Date Submitted: April 22, 2015

ACKNOWLEDGEMENTS

During the past year, I have developed deep interest in the concept of Influence Maximization in Social Networks. I would like to express my gratitude to all those who gave the encouragement in completing this Thesis successfully. I would like to thank my advisor Dr. Jing (Selena) He for her support and encouragement. She helped me throughout the development of Thesis by offering valuable detailed advices on organization, idea flow and the overall theme of the work and this would not have been that perfect without her guidance. I would also like to thank Dr. Haddad, Dr. Hoganson, Dr. Shi and Dr. Xie for their support as I moved from an idea to a completed study. I also thank them for their encouragement, thoughtful guidance and insightful comments. The technologies I learned in their courses really helped me to conduct the simulations. My parents deserve special mention for their love, encouragement, and support throughout my life. The thesis is impossible without their support. This work is funded in part by the Kennesaw State University in the form of Graduate Research Assistantship (GRA) from Graduate College, the Interdisciplinary Research Opportunities Program (IDROP) from College of Science and Mathematics and Pilot/Seed Grant from Office of the Vice President for Research (OVPR).

TABLE OF CONTENTS

LIST OF FIGURES

SUMMARY

Nowadays, social networks are considered as the very important medium for the spreading of information, innovations, ideas and influences among individuals. Viral marketing is a most prominent marketing strategy using word-of-mouth advertising in social networks. The key problem with the viral marketing is to find the set of influential users or seeds, who, when convinced to adopt an innovation or idea, shall influence other users in the network, leading to large number of adoptions. Therefore, the major problem is to find the initial, well positioned set of individuals who will be able to create word-of-mouth cascades. In our study, we propose and study the competitive viral marketing problem from the host perspective, where the host of the social network sells the viral marketing campaigns to its customers and keeps control of the allocation of seeds. Seeds are allocated based on the budget of the company and in such a way that it is creates the bang for the buck for each company (or Fair Seed Allocation). Our study is to propose a new diffusion model considering the host perspective in Online Social Networks (OSN) where the network model will have both positive and negative edges. We take both negative and positive influences into consideration and propose a novel problem, named Blocking Negative Influential Node Set(BNINS) selection problem, to identify the positive node set such that the number of negatively activated nodes is minimized for all competitors from host perspective. In other words, we try to block the negative influence propagation in social networks from host perspective. We first provide our newly proposed diffusion model, define the novel BNINS problem, propose a solution to the problem (BNINS-GREEDY) and simulation results to validate the proposed solution. We also compare our work with the related work [16] to check the performance of BNINS-GREEDY under different

metrics and we observed that BNINS-GREEDY outperforms the others algorithm. In Random Graph, on average, BNINS-GREEDY blocks the negative influence 17.22% more than CLDAG. At the same time, it achieves 7.6% more positive influence propagation than CLDAG. We also analysed the BNINS-GREEDYś performance by testing it on various performance metrics that includes total number of positive and negative activations with varying number of nodes, varying number of companies and varying number of time iterations etc. The results clearly shows that BNINS-GREEDY achieved its objective of minimizing the negative influence with great proficiency.

CHAPTER I

INTRODUCTION

A social network is a graph of interactions as well as relationships among a group of individuals. Social networks also play an important role as a medium to spread influences, ideas and information among its users. An innovation or idea can either die out quickly or can make considerable impacts upon the population, for example, rise of political movement in unstable society, cell phone usage among college students or adoption of new drug in medical profession etc. In order to understand the degree to which such ideas are adopted, it is very critical to have the understanding of how the dynamics of adoption are likely to unfold within the underlying social network: the degree to which individuals are expected to be affected or influenced by the decisions of their colleagues and friends, or the degree to which the word-of-mouth effects will take hold.

Recently Domingos and Richardson motivated by the applications to marketing posed a fundamental algorithmic problem for such frameworks [19, 20]. Let us assume that we have some data on the social network. Therefore, by estimating the extent to which individuals are influenced by each other, we would like to put a new product in the market hoping that it will be adopted by large number of individuals in the network. The premise of viral marketing is to conduct product promotions using social influences between individuals cycles friends, families, or co-workers. The products are promoted by giving free or discounted items to highly influential individuals (initial adopters) and the product adoptions are believed to be improved through word-of-mouth effects. This technique is considered very effective due to trusted relationships and is also gaining huge amount of popularity these days. This is called as viral marketing. When we have more than one player competing with each other for similar products in the same social network then it is called competitive viral marketing.

Motivated by this background, the community of researchers has recently studied the aspect of influence maximization in social networks for viral marketing [1, 2, 3, 4, 5, 6, 7, 8, 9]. Influence Maximization is a fundamental data mining problem concerning the propagation of ideas, opinions, and innovations through social networks. Kempe et al. [1] formulated influence maximization as a problem in discrete optimization: Given a network graph G with pair wise user influence probabilities on edges, and a positive number k, find k users, such that by activating them initially, the expected spread of influence is maximized under certain propagation models. Two basic influence cascade models are used in these works: Independent Cascade (IC) Model and Linear Threshold (LT) Model. Kempe et. al. in [1] originally defined these models. In both the models, social network is modelled as a graph that starts from an initial set of vertices. The stochastic process states how influence is propagated from this initial set to their neighbors and neighbors of neighbors. This goes on until the process ends and a some part of the social network is activated. Therefore, the influence maximization problem is defined as finding an optimal seed set of size at most k such that the expected number of vertices activated from this seed set, referred to as its influence spread, is maximized. But the point of consideration is: How to choose the few key individuals to use as the initial sets for this process? In [19, 20] Richardson and Domingos considered this question in a probabilistic model of interaction; heuristics were also given to choose the customers having large overall effect on the network, and methods were also developed to infer the influence data necessary for posing such kind of problems.

On the other hand, none of the above works considered one critical aspect of influence maximization that we usually come across in the real world. That is, not only positive opinions propagate through the networks but the negative opinions do too. And also, negative opinions are often more contagious and stronger in affecting peoples decisions than positive opinions. For example, if we hear from one of our friend that he/she found a cockroach in his/her meal in the nearby restaurant then more likely we will try to avoid that restaurant for sometime. Furthermore, we are likely to convey our other friends regarding this incident, discouraging them to patronize the restaurant even though we did not have this bad experience ourselves. On the contrary, if we hear some good words about some restaurant, then we are more likely to visit there but probably we will spread the good words about it only after experiencing a good meal there ourselves. Therefore negative opinions have much more impact on an individual than positive impact.

The impact of negative opinions and its asymmetry with positive opinions have long been studied in the social psychology literature [21, 23, 24, 25]. In these studies, researchers proved that negative impact is mostly much stronger as well as dominant than positive impact in shaping peoples decisions. Negative influences are also explicitly addressed in marketing literatures: individuals spreading negative opinions are called detractors whereas people spreading positive opinions are called promoters [22]. Hence, while studying the problem of influence maximization, it should be critically considered to incorporate the emergence and propagation of negative opinions into the IC model as well as LC model and study the impact of negative influence together with positive influence. This is exactly the goal of our study.

On the other hand, in social networks, it is often a case when there are different campaigns and opposite ideas, information or products competing for their influence in the social network. Motivated by this observation, we concentrate on the problem of how to block the diffusion of an opposing company as much as possible. For example, when there is a negative rumour spreading about a company, then that company wants to react quickly by selecting seed nodes to inject the positive opinions about itself to fight against the negative rumour. X. He et al. [16] considered the influence blocking maximization with only two competing companies but in our work we will generalize that model to k number of companies. And hence we proposed a novel problem called Blocking Negative Influential Node Set Problem (BNINS) in social networks where we have a negative seed set and we have to find all k positive seed sets to minimize the effect of negative influences for k companies.

Another observation is that, in most of the research works, the seeds (initial adopters) are selected based on some criteria or algorithm but this is actually not the case. The social networks are owned by third party like Twitter, LinkedIn, Facebook etc. The proprietary of the social graph is kept secret for privacy as well as company benefits. The owner of the social network is called "host" and companies trying to run the viral campaigns are called "clients" for the hosts. Clients cannot access the social network directly and hence they cannot choose seeds for their campaign by themselves. Clients would need the host permission and privilege to run. Motivated by this observation, we propose and study the naive problem of competitive viral marketing from the host perspective. In this study, we consider a business model with host offering viral marketing as a paid service to its clients. The clients will hence be able to run the campaigns by specifying the seed budget *i.e.* the number of seeds desired. The host of the social network controls the seed selection and allocation to companies. The seeds will be allocated in such a way that guarantees the bang for the buck for all companies is nearly same. The bang for the buck for a company is the cost benefit ratio between the expected number of adopters of its product over its number of seeds. We call this the amplification factor, as it reflects how investing in a small number of seeds gets amplified by the network effect. If the host allocates the seeds carelessly to its clients, it can result in a wide variance in the amplification factors, leading to resentful clients.

The rest of the Thesis is organized as follows: Chapter II gives the details about related works. Chapter III discusses the problem definition. This involves three sections, Section 3.1 introduces the network model and then in Section 3.2 proposes the new diffusion model. Section 3.3 formally defines the problem. BNINS-GREEDY algorithm is presented in chapter IV. Chapter V discusses the simulation setting and the results. Chapter VI confirms the Validation of Simulation. Finally Chapter VII concludes the Thesis work and discusses the future work.

CHAPTER II

RELATED WORK

This section summarizes previous related research work on diffusion models, influence maximization problem and influence blocking maximization.

2.1 Network Model

Social network is usually modelled as a graph $G = (V, E)$, where V is the set of nodes representing individuals and E is the set of edges representing relationships between each pair of individuals. Every edge is associated with weight value representing social influences on pair of individuals. In many research works, two kinds of Network Models are considered: Signed and Unsigned Network Model. These are discussed as follows:

2.1.1 Unsigned Network Model

In Unsigned Network Model, the edges are neutral. They do not have positive or negative weights associated with their edges. Both Independent Cascade (IC) model [1, 7, 13, 14, 28] and Linear Threshold (LT) [1] model use unsigned networks. Kempe et al. [1] studied the problem of identifying the influential set of nodes in order to maximize the spread of influence. The greedy algorithm proposed by Kempe et al.[1] and its improvements are too slow and unscalable. Therefore, W. Chen [7] considered scalability factor and designed a new heuristic algorithm which is easily scalable to billions of nodes as well as edges in the experiments. S. Bharthi [13] studies the influence maximization problem when multiple companies are competing to promote their products or services using viral marketing. C. Budak [14] considered the case of limiting the spread of misinformation in social networks. Z. Wang [28] proposed a new influence propagation model as the extension of the classic IC model in which he added the propagation probability of each node in order to distinguish the influence and propagation in social networks.

2.1.2 Signed Network Model

In Signed Network Model, the edges can hold both positive as well as negative influence. Both Linear Threshold (LT) model [16, 17, 26] and Independent Cascade (IC) model [27] use signed networks. X. He [16] studied the competitive influence propagation problem and he proposed a model in which one entity tries to block the influence propagation of the opposing entity as much as possible. This is done by strategically selecting a number of seed nodes that could initiate its own influence propagation whereas W. Lu [17] considered maximizing the spread of information from the host perspective without considering negative influences. Y. Ganjali [26] considered the influence maximization problem from a different perspective. Instead of identifying the most influential individuals, he generalized the problem of most influential groups. In [27], Y. Li investigated the influence diffusion as well as influence maximization in OSNs. He also considered the friend and foe relationships among individuals. These are modeled using positive and negative edges on signed networks. J. He [29] discussed Minimum sized Positive Influential Node Set (MPINS) selection problem, to identify the minimum set of influential nodes, such that every node in the network can be positively influenced by these selected nodes no less than a threshold θ. In our research work, we use the signed network model.

2.2 Diffusion Model

Kempe et al. [1] summarized two extensively studied influence diffusion models: Independent Cascade (IC) Model and Linear Threshold (LT) Model based upon the previous works [10, 11, 12]. M. Granovetter [10] studied the models of collective behavior, based on behavioral thresholds, which account for collective outcomes by simple principles of aggregation whereas T. Schelling in his book [11], discussed about how small and seemingly meaningless decisions and actions by individuals often lead to significant unintended consequences for a large group. Kempe et al. [1] also proved that the generalized versions of these two models are equivalent. Then they proposed a greedy algorithm to solve the influence maximization problem under these two influence diffusion models. In both the models, we have a social graph G = (V, E) with edges (u, v) ϵ E labelled by influence weights $p_{(u,v)}$ ϵ (0, 1]. Each node is either active or inactive. An active node never becomes inactive. At time 0, a set of nodes called seeds become active.

2.2.1 Linear Threshold Model

In the LT model [1, 16, 17, 26], we have to consider a node which is influenced by its neighbors. This model derives that the sum of the weights of all neighboring nodes must be less than or equal to one. Each node v chooses a threshold θ_v uniformly at random from [0, 1]. If at time t, the total incoming weight from active in-neighbours of v is greater than or equal to θ_v, then v becomes active and this node tends to adopt the tendencies of its neighboring nodes. The important constraint here is that the node becomes active from inactive but not vice versa. So, the major study of this paper is to work on the influence diffusion and influence maximization taking the LT model concept. The model assumes that a node becomes active from inactive but not vice versa.

2.2.2 Independent Cascade Model

Independent Cascade Model (IC) model [1, 7, 13, 14, 27, 28] describes that an active node has the probability of p to activate one of its neighboring nodes. The IC model is very basic and well-studied diffusion model. In the IC model, a process is initiated with an initial set of active nodes A_0 or seeds, and then the process unfolds in discrete steps. When a node v becomes active in step t, then it has one chance to activate one of its inactive neighboring node w. The node v succeeds in activating w with probability $p_{(v,w)}$. If v succeeds, then node w becomes active in step $t + 1$. However, whether v succeeds in its attempt or not , it cannot make any further attempts in next rounds. This process continues until no other activations are possible. In case, if node w has many incoming edges from numerous newly activated nodes, then their attempts are sequenced in an arbitrary order.

2.3 Influence Maximization Problem and its Variations

2.3.1 Influence Maximization Problem

Given a propagation model (e.g., IC or LT) and a seed set $S \subseteq V$, the expected number of active nodes at the end of the process, or the (expected) spread, is denoted by $\sigma(S)$. The influence maximization problem asks for a set $S \subseteq V$, $\mid S \mid = k$, such that $\sigma(S)$ is maximized, where k is an input parameter. In LT and IC model, Influence Maximization is considered as is NP-hard [1]. Kempe et al.[1], also proved that the function $\sigma(S)$ is monotone as well as submodular. With these properties, the simple greedy algorithm at each iteration greedily extends the current seed set S with the node w providing the highest marginal gain value $\sigma(S \cup \{w\}) - \sigma(S)$, gives a $(1 - 1/e - \epsilon)$- approximation to the optimum [1, 18] (for any $\epsilon > 0$). In the later works, more efficient as well as scalable influence maximization algorithms were developed [4, 7, 9]. A. Krause [4] presented a new methodology to select the nodes in order to detect outbreaks of dynamic processes spreading over a graph. Prior solutions to the influence maximization problem, such a greedy algorithm of Kempe [1] and its modifications are slow and unscalable whereas other heuristic algorithms do not provide consistently good performance on influence spreads. C.Wang in [7] designed a new heuristic algorithm that is easily scalable to millions of nodes and edges. L. Zhang in [9] showed that computing the exact influence spread in the LT model is NP-hard, even if there is only one seed in the network. Also, based on the fast influence computation for directed acyclic graphs (DAGs), he proposed the first scalable heuristic algorithm tailored for influence maximization in the LT model called local DAG algorithm (LDAG).

2.3.2 Influence Maximization from Host Perspective

The host of the social network keeps the proprietary of social graph secret due to various reasons like privacy legislation and company benefits etc. The authors in [17] considered the host as the owner. Companies that intend to run viral marketing campaigns are considered as clients of the host. Usually the clients cannot directly access the network and hence they cannot select the seeds for the campaigns by themselves. For any client to run viral marketing campaign, the host permission and run privileges are needed for example, Twitter. Business owners or the clients of the social network who wants to promote their products or services through Twitter can create their own Twitter webpage, create display ads or promoted posts to reach users. However, clients are not permissible to effectively run the viral marketing campaign to reach the users. This is because of lack of access to the network graph as well as privacy issues. W. Lu [17] proposed and studied the naive problem of competitive viral marketing from the host perspective and considered a new business model in which the host offers viral marketing as a service, for a price to its clients. The host also allows its clients to run the campaigns by specifying a seed budget, *i.e.*, number of seeds desired. The host also keeps control over the selection as well as allocation of seeds to the companies. After the seeds are allocated to the companies, various companies with similar products or services compete for adopters on the shared network. In traditional non-competitive influence maximization problem, the objective is to select the seeds in such a way that maximizes the expected number of final adopters. But, in a competitive setting, from the hosts perspective, it is critical not just to select the seeds to maximize the collective expected number of adoptions across all companies, but also to allocate the seeds to companies in a way that guarantees the bang for the buck for all companies is nearly the same. A. Goyal in [17] discussed the diffusion model from the host perspective but he did not considered the competitive setting with multiple companies competing to promote their products/services. However their work did not considered the negative influences. That is why we would like to study the problem of influence maximization from the host perspective by considering the negative influences.

2.4 Influence Blocking Model

Whenever a company sees a negative rumor spreading against its products or services then the company may decide to react to it quickly by choosing the seed nodes to inject the positive opinions in order to fight against the negative rumor. Similar type of situations may arise when public officials try to abort rumors regarding public safety and health, terrorist threat and a political candidate tries to do everything to stop that negative rumor about him or her etc. The authors of [16] identify this problem of choosing positive seeds in social network in order to lessen the effect of negative influence diffusion or to maximize the blocking effect on negative influence, the influence blocking maximization (IBM) problem [16]. Motivated by this, we want to consider the blocking of negative influence in our study to solve the influence maximization problem.

Recently there had been studies on competitive influence diffusion [13, 14]. The commonality among these studies is that they concentrate on the clients perspective as opposed to the host perspective. Bharathi et al. [13] and Carnes et al. [15] studied the influence maximization problem from the followers perspective. The follower is also a player who tries to bring in a new product into the social network where a competing product already exists. Both of these studies show the problem for the follower maintains the desired properties of monotonicity and submodularity and hence the greedy algorithm can be applied to provide approximation guarantees. Goyal et al. [17] considered the influence diffusion but the objective of their study was to maximize the influence spread. Budak et al. [14] and Chen et al. [16] studied the influence blocking maximization problem, where one entity tries to block the influence propagation of its competitor as much as possible, under extended IC and LT models, respectively.

Hence, in this work, we propose an influence diffusion model from the host perspective, which has both positive and negative influence propagating. We try to maximize the positive spread and conversely minimize the negative spread. We name it as Blocking Negative Influence Node Set (BNINS) Selection problem where the positive influence tries to block the negative influence.

CHAPTER III

PROBLEM DEFINITION

3.1 Network Model

A social network can be modeled as a weighted directed graph $G = (V, E, W)$ as shown in Figure 1 where V is the set of nodes representing individuals and E is the set of edges representing influential relationships among individuals. An edge can be represented as follows:

$$e = \begin{cases} e_{(v_i,v_j)} = \{ \text{ represents an edge from node } v_i \text{ to node } v_j \ \} \\ e_{(v_j,v_i)} = \{ \text{ represents an edge from node } v_j \text{ to node } v_i \ \} \end{cases} \tag{1}$$

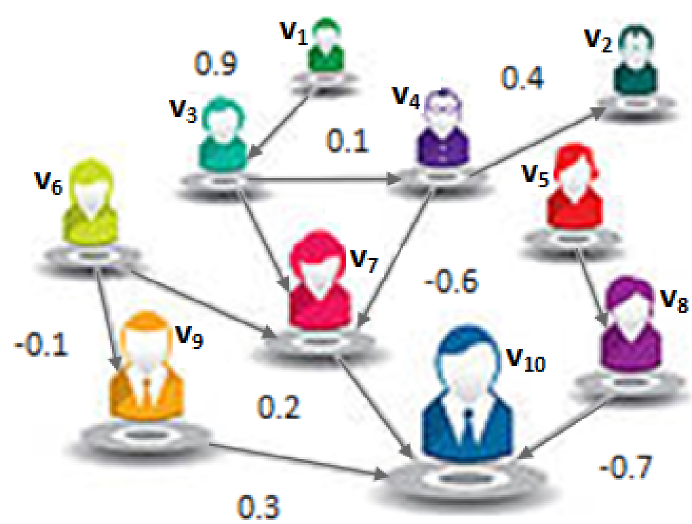


Figure 1: A social network with social influences on edges.

In the context of influence spread, $\forall v_i \in V$ can be viewed as a user of the social network and i is the user ID. In addition to this, W is a set of weights. Each edge $e_{(v_i,v_j)} \in E$ is assigned a weight value representing the direct influence node v_i has on node v_j. *i.e.*, $W = \{w_{ij} \mid \forall e_{(v_i,v_j)} \in E\}$ and each edge $e_{(v_i,v_j)} \in E$ has two weights:

w_{ij}^{+} and w_{ij}^{-}.

$$w_{ij} = \begin{cases} w_{ij}^{+} = \{v_{ij} \mid v_i \text{ has positive opinion on } v_j\ \} \\ w_{ij}^{-} = \{v_{ij} \mid v_i \text{ has negative opinion on } v_j\ \} \end{cases} \tag{2}$$

We can think of it as $e_{(v_i,v_j)}$ splitting into two virtual edges, one positive edge propagating positive influence and one negative edge propagating negative influence. As shown in Figure 1. $v_1, v_2, ..., v_{10}$ represents the different nodes. $e_{(v_1,v_3)}$ represents an edge from node v_1 to node v_3 and w_{13} is 0.9, that represents the positive opinion that node v_1 has on node v_3.

3.2 Diffusion Model

In a social network, a node can have either an active or inactive status. Every node in the network holds one of the two opinions (positive or negative). So, we further divide the active status into *positive active* and *negative active*. Every company will have an initial positive active and an initial negative active seed set assigned by the host. There will be one of the following conditions:

Definition II.1. *Company (C_k)*. In a social graph with m competing companies, C_k is the *kth* company for $k \in \{1, 2, ..., m\}$ and k is the company ID.

Definition II.2. *Budget (b_k)*. Budget is defined as a number of seeds or initial adopters for every company. Each company has to specify its budget in order to introduce their products or services into the social network.

Definition II.3. *Active Seed Set ($A_k(t)$)*. For a network G = (V, E, W), the Active Seed Set is defined as the set of all *positive* and *negative* active nodes in Company C_k at time *t*.

$$A_k(t) = \{v_i \mid v_i \in V \text{ and is active with Company } C_k \text{ at time } t\ \}$$

Definition II.4. *Positive Active Seed Set ($A_k(t)^+$)*. For a network G = (V, E, W), the Positive Active Seed Set is defined as the seed set with all *positive active* nodes in Company C_k at time *t*. When *t=0*, $A_k(0)^+$ represents the initially positive seed set for company C_k.

$A_k(t)^+ = \{v_i \mid v_i \in$V and is positively active for Company C_k at time $t\}$

Definition II.5. *Negative Active Seed Set ($A_k(t)^-$).* For a network G = (V, E, W), the Negative Active Seed Set is defined as the seed set with all *negative active* nodes in Company C_k at time t. When *t=0*, $A_k(0)^-$ represents the initially negative seed set for company C_k.

$A_k(t)^- = \{v_i \mid v_i \in$V and is negatively active for Company C_k at time $t\}$

The relationship of the pre-defined three sets can be summarized as:

$$A_k(t) = A_k(t)^+ \cup A_k(t)^-$$

Definition II.6. *Active Node Set (A(t)).* For a network graph G = (V, E, W), A(t) is the set of nodes activated by any Company C at time t.

A(t)= $\{v_i \mid v_i \in V$ and is active with any Company C at time t $\}$

i.e., A(t)= $\sum_{k=1}^{m} A_k(t)$

Definition II.7. *Positive Active Node Set ($A(t)^+$).* For a network G = (V, E, W), the Positive Active Node Set is defined as the set of all *positive active* nodes in any Company C at time t. When *t=0*, $A(t)^+$ represents the initially positive seed set for any company C.

$A(t)^+ = \{v_i \mid v_i \in$V and is positively active for any Company C at time $t\}$

Definition II.8. *Negative Active Node Set ($A(t)^-$).* For a network G = (V, E, W), the Negative Active Node Set is defined as the set of all *negative active* nodes in any Company C at time t. When *t=0*, $A(t)^-$ represents the initially positive seed set for any company C.

$A(t)^- = \{v_i \mid v_i \in$V and is negatively active for any Company C at time $t\}$

The relationship of the pre-defined three sets can be summarized as:

$$A(t) = A(t)^+ \cup A(t)^-$$

Definition II.9. *Neighbouring Set ($N_{v_i}(t)$).* At time t, for a network graph G= (V, E, W), Neighbouring Set is defined as the set of all the nodes approaching $v_i \in V$.

$$N_{v_i}(t) = \{v_j \mid e_{(v_j,v_i)} \in E, w_{ji} \neq 0\}$$

Definition II.10. *Positively Active In-Neighbours($N_{v_i}(t)^+$).* At time t, for a network graph G = (V, E, W), Positively Activated In-neighbours are defined as the nodes approaching $v_i \in V$ which are positively activated. It is defined as:

$$N_{v_i}(t)^+ = \{v_j \mid e_{(v_j,v_i)} \in E, w_{ji} > 0 \text{ at time } t\}.$$

Definition II.11. *Negatively Active In-Neighbours($N_{v_i}(t)^-$).* At time t, for a network graph G = (V, E, W), Negatively Activated In-neighbours are defined as the nodes approaching $v_i \in V$ which are negatively activated.

$$N_{v_i}(t)^- = \{v_j \mid e_{(v_j,v_i)} \in E, w_{ji} < 0 \text{ at time } t\}.$$

The relationship of the above three sets can be summarized as:

$$N_{v_i}(t) = N_{v_i}(t)^+ \cup N_{v_i}(t)^-$$

The diffusion model will work as follows:

For each company, the seed set $A_k(t)$ is further divided into $A_k(t)^+$ and $A_k(t)^-$ where $A_k(t)^+$ is the positive seed set of Company C_k and vice versa. Every node $v_i \in V$ picks two activation thresholds θ_i^+ and θ_i^- uniformly at random from [0, 1].

$$\theta_i = \begin{cases} \theta_i^+ \text{ represents the positive threshold for node } v_i \\ \theta_i^- \text{ represents the negative threshold for node } v_i \end{cases} \tag{3}$$

Initially all the nodes are inactive i.e., $A_k(t) = \phi$.

At time *t=0*,

For each Company C_k, a seed set $A_k(0)^+$ and $A_k(0)^-$ is assigned by the host. This means that if $v_i \in A_k(0)^+$ or $A_k(0)^-$, then v_i becomes positively activated or negatively activated with Company C_k respectively.

At time $t \geq 1$,

The activation of node will take place in two phases:

In the diffusion model, a node will be *influenced* if it possess some positive or negative influence but a node will be *activated* if it is influenced as well as associated with some company.

PHASE 1:

An inactive node $v_i \in V$ becomes influenced when the total incoming influence weight from its in-neighbours ($N_{v_i}(t)^+$ and $N_{v_i}(t)^-$) which are active (regardless of Company) reaches v_i 's threshold:

$$\begin{cases} \text{if} \sum_{v_j \in N_{v_i}(t)^+} |w_{ji}| \geq \theta_i^+, \text{ then } v_i \text{ will be positive influenced} \\ \text{if} \sum_{v_j \in N_{v_i}(t)^-} |w_{ji}| \geq \theta_i^-, \text{ then } v_i \text{ will be negative influenced} \end{cases} \tag{4}$$

In sum, if the total positive or negative influence is greater or equal to the corresponding positive or negative activation threshold, then the node will become influenced with that influence. If there is a tie between the total positive and negative influence then the negative influence is given the priority due to negative dominance rule. This rule reflects the negativity bias phenomenon well studied in social psychology, in which negative opinions usually dominate over positive opinions [21].

PHASE 2:

In the previous phase, the node became positively or negatively influenced. In this phase, a node $v_i \in V$ becomes active by picking a Company out of those of its in-neighbours that activated at time *t-1* with the same influence in which it got influenced in Phase 1. At t, a node v_i becomes active with Company C_k with probability:

$$p_j = \begin{cases} p_j^+ = \sum_{v_j \in A_k(t-1)^+/A_k(t-2)^+} w_{ji}^+ / \sum_{v_j \in A(t-1)^+/A(t-2)^+} w_{ji}^+ \\ p_j^- = \sum_{v_j \in A_k(t-1)^-/A_k(t-2)^-} w_{ji}^- / \sum_{v_j \in A(t-1)^-/A(t-2)^-} w_{ji}^- \end{cases} \tag{5}$$

The node will be activated with p_j^+ or p_j^- based upon the influence in which it got influenced in Phase 1. If it is positively influenced then it will be activated with probability p_j^+ and otherwise with probability p_j^-. $A_k(t-1)^+/A_k(t-2)^+$ means

set of all positive nodes activated by Company C_k at time *t-1* but not at time *t-2*. For example, at *t=1* in Figure 2, the set of positively activated nodes in C_1 *i.e.*, $(A_k(t-1)^+)$ or $(A_1(0)^+)$ is $\{v_1, v_4\}$ and $(A_k(t-2)^+)$ or $(A_1(-1)^+)$ is ϕ, then $A_1(0)^+/A_1(-1)^+ = \{v_1, v_4\}$. Similarly, $A(t-1)^+/A(t-2)^+$ for any Company.
If this probability comes out to be greater than zero than the node will become active with the company C_k where $k \in \{1, 2, ..., m\}$, otherwise the node will stay inactive.

For better understanding of our proposed diffusion model, we use an example of a network to explain it. From Figure 2(a), at *t=0*, the seed set of every company is targeted and $A_1(0)^+ = \{1,\ 4\}$, $A_1(0)^- = \{2\}$ and $A_2(0)^- = \{3\}$. The status of these nodes is changed to active corresponding to the company whose seed set it belongs to. Given two thresholds θ_5^+ and θ_5^- for v_5 are 0.5 and 0.5 respectively. At time *t=1* (Phase 1), the total positive influence from positive active in-neighbours on v_5 is $(0.2{+}0.2) = 0.4$, which is less than v_5's positive threshold *i.e.* 0.5. The total negative influence from negative active in-neighbours on v_5 is $(0.4{+}0.3) = 0.7$, which is greater than v_5's negative threshold. So, the node v_5 will be negatively influenced as shown in Figure 2(b). At time *t=1* (Phase 2), The node v_5 will choose a company out of C_1 and C_2 based on the following criteria:

$$p_j = \begin{cases} p_j^+ = \sum_{v_j \in A_k(t-1)^+/A_k(t-2)^+} w_{ji}^+ / \sum_{v_j \in A(t-1)^+/A(t-2)^+} w_{ji}^+ \\ p_j^- = \sum_{v_j \in A_k(t-1)^-/A_k(t-2)^-} w_{ji}^- / \sum_{v_j \in A(t-1)^-/A(t-2)^-} w_{ji}^- \end{cases} \tag{6}$$

Now we know that v_5 is negatively influenced. Therefore it will be activated using probability p_j^-. For C_1, the set of negative nodes activated by Company C_1 at time *t-1* *i.e.*, $A_1(0)^- = v_2$. Here $A_1(-1)^- = \phi$. So $A_1(0)^-/A_0(-1)^- = v_2$. The total incoming influence from the node v_2 is 0.4. Similarly $A(-1)^- = \phi$ and the set of negative nodes activated by any Company C at time *t-1* *i.e.*, $A(0)^- = \{v_2, v_3\}$. The total incoming influence from all the companies C activated at time *t-1* *i.e.* $A(0)^- = (0.3{+}0.4) =$

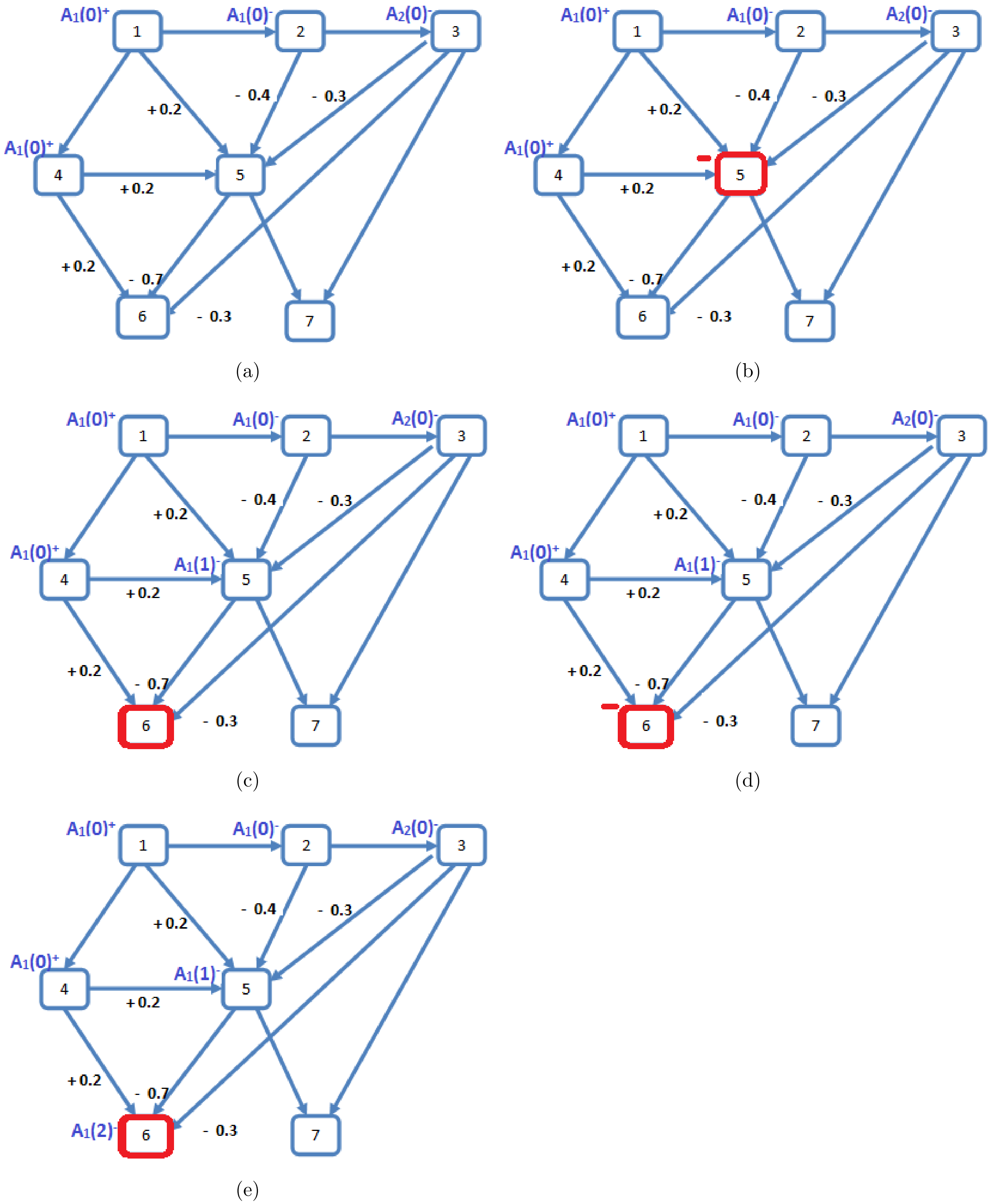


Figure 2: Demonstration of Diffusion Model

0.7. v_5 will be activated with C_1 with probability 0.4/0.7= 0.57. Similarly, for C_2, the probability comes out to be 0.3/0.7= 0.43. So the node v_5 will be negatively activated with C_1 with probability 4/7 and with C_2 with probability 3/7 as shown in Figure 2(c).

At time *t=2* (Phase 1), Given two thresholds θ_6^+ and θ_6^- for v_6 are 0.5 and 0.5 respectively. The total positive influence from positive active in-neighbours on v_6 is 0.2, which is less than v_6's positive threshold *i.e.* 0.5. The total negative influence from negative active in-neighbours on v_6 is (0.7+0.3) = 1.0, which is greater than v_6's negative threshold. So, the node v_6 will be negatively influenced as shown in Figure 2(d). At time *t=2* (Phase 2), $A_1(1)^- = v_5$, $A_1(0)^- = v_2$. So, $A_1(1)^-/A_1(0)^-$ $= \{v_5\}$. The total incoming influence of node v_5 on v_6 is 0.7. $A(1)^- = v_5$, $A(0)^- =$ $\{v_2, v_3\}$. So, $A(1)^-/A(0)^- = v_5$. The total incoming influence here is 0.7. Hence the node v_6 will be activated by C_1 with probability 0.7/0.7=1 as shown in Figure 2(e). Similarly the probability of activating v_6 by C_2 is 0. To be specific, v_6 will be negatively activated with C_1 for sure at *t=2*. This process keeps on repeating until all the nodes get activated.

3.3 Blocking Negative Influential Node Set (BNINS) Selection Problem From the host perspective

As we mentioned before that our goal is to maximize the spread of positive opinion such that the negative opinion is minimized, or equivalently, the reduction in the number of negatively activated nodes is maximized. Now we are ready to define the problem statement for this Thesis work.

Definition III.1. *Expected value of* $A_k(t)^+(\rho_k(t))$.The expected value of Positive Active Seed Set with Company C_k is $\rho_k(t) = \sum_t t \mid A_k(t)^+ \mid$ at time *t*.

Definition III.2. *Blocking Negative Influential Node Set (BNINS) Selection Problem.* Given a graph G = (V, E, W), initial negative seed sets $A_k(0)^-$ for Company C_k and positive integers P_k where $k \in \{1, 2, ..., m\}$, the BNINS selection problem is to find positive active seed sets $A_k(t)^+$ of size at most P_k such that the number of negatively activated nodes is minimized, or equivalently, $\rho_k(t)$ is maximized.

CHAPTER IV

PROPOSED SOLUTION TO THE BNINS PROBLEM

4.1 BNINS-GREEDY Algorithm

This section discusses the BNINS-GREEDY algorithm for BNINS Problem but before introducing BNINS-GREEDY, we first define a useful Amplification Function as follows:

Definition IV.1. *Amplification function($f(v_i, k)$).* For a social network $G = (V,E,W)$, the Amplification function of v_i is defined as follows:

$$f(v_i, C_k)= \frac{(\sum_{v_j \in A_k} w_{ji}^{+})\theta_i^{+}+(\sum_{v_j \in A_k} w_{ji}^{-})\theta_i^{-}}{\theta_i^{+}+\theta_i^{-}}$$

Amplification function denotes the contribution of a node towards the Company. More the value of the Amplification Function, more is the contribution of the node towards the Company. It can be illustrated as follows:

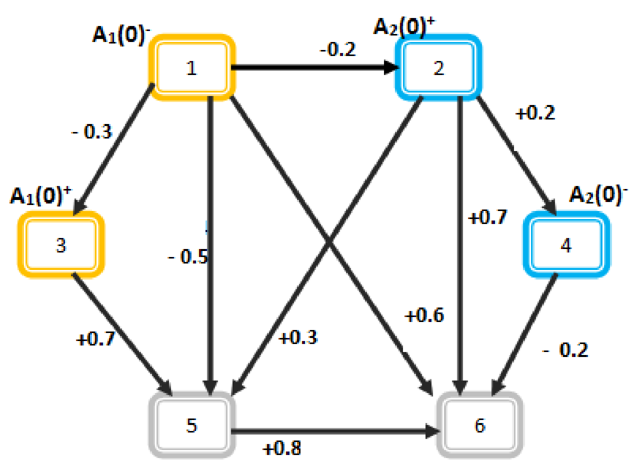


Figure 3: Amplification function for v_5

In the Figure 3, $A_1(0)^- = \{v_1\}$, $A_1(0)^+ = \{v_3\}$, $A_2(0)^- = \{v_4\}$ and $A_2(0)^+ = \{v_2\}$. Let us assume the value for $\theta_5^+ = 0.5$, $\theta_5^- = 0.5$, $\theta_6^+ = 0.5$ and $\theta_6^- = 0.5$. For node v_5, the Amplification Function values can be calculated as follows:

- $f(v_5, C_1) = \frac{(0.7)(0.5)+(-0.3)(0.5)}{0.5+0.5} = 0.20$
- $f(v_5, C_2) = \frac{(0.3)(0.5)+(0)(0.5)}{0.5+0.5} = 0.15$

Likewise, for node v_6, the Amplification Function values are as follows:

- $f(v_6, C_1) = \frac{(0)(0.5)+(-0.6)(0.5)}{0.5+0.5} =$ -0.30
- $f(v_6, C_2) = \frac{(0.7)(0.5)+(-0.2)(0.5)}{0.5+0.5} = 0.25$

From the above results, we infer that node v_5 has more contribution for C_1 and v_6 has more contribution for C_2. This is because the value of Amplification Function for Company C_1 is greater for v_5 than v_6 and similarly for C_2. And hence Amplification Function is very useful for selecting the seeds based upon their contributions to different companies.

Now we will discuss the BNINS-GREEDY Algorithm to find the positive seed set so as to minimize the negative activated nodes.

Algorithm 1: BNINS-GREEDY Algorithm

Input: $A_k(0)^-$ and b_k, $\forall k \in \{1, 2, ..., m\}$,

To find: $A_k(0)^+$ where $|A_k(0)^+| + |A_k(0)^-| = b_k$, $\forall k$,

1. Initialize $A_k(0)^+ = \phi$

✍ Initialize all positive seed sets to be empty.

2. $T \leftarrow \{k \mid k \in \{1, 2, ..., m\}, |A_k(0)^+| < b_k - |A_k(0)^-|\}$;

✍ T is the set of companies for which the budget has not been exhausted.

3. **for** *each* $t \in T$

4.   **for** *each* $v_i \in V - (A(0)^- + A(0)^+)$ **do**

5.   **Calculate** $f(v_i, C_t)$ and *store* it in a MAP$< v_i, f(v_i, C_t) >$;

6. **End for loop**

7. **Traverse** MAP and find out one node that has maximum amplification function value for Company C_t.

8. ***then*** that node will be assigned to Company C_t.

9. **Repeat** step 2-9 until the budget of all the companies has been exhausted.

As shown in Algorithm I, we first set $A_k(0)^+$ to ϕ (line 1), then we check all the companies whose budget has not been exhausted yet and store it in a set T (line 2). After this, we loop through all the Companies in T (line 3) and all the nodes in $V - (A(0)^-$ $+ A(0)^+)$ (line 4). And we calculate the value of Amplification Function $(f(v_i, C_t))$ of each node $v_i \in V - (A(0)^- + A(0)^+)$ for Company $C_t \in T$ and store it in the MAP data structure (line 5,6). Then we traverse the MAP to check the contribution of each node towards each Company $C_t \in$ T and pick one node having the maximum Amplification value for that Company (line 7). In case when more than one node is having the same and maximum value of the Amplification Function value then we use a tuple $< out - degree, nodeID >$ to break the tie in order. We calculate the out-degree of those nodes and higher out-degree node wins and selected as a seed node. Now there can be a case, when its a tie on the higher out-degree, then node ID is used to break the tie. The lower node ID wins and will be assigned to the respective Company as illustrated in Definition 4.1. Now we repeat the steps from 2 through 9 (line 9). We will notice that the size of set T will reduce as the budget of Companies is exhausted. The algorithm will terminate when the budget of all the Companies will be exhausted.

4.2 BNINS-GREEDY Example

To better understand the proposed heuristic algorithm, we use the following example:

We use the social network represented by the graph shown in Fig.4(a). In the figure, blue color nodes represent Company C_1, green color nodes represent Company C_2 and grey nodes represent inactive nodes. The selection procedure is illustrated as follows:

Input: $A_1(0)^- = \{v_1, v_2\}$, $A_2(0)^- = \{v_3\}$, T $= \{C_1, C_2\}$, $b_1 = 4$, $b_2 = 2$, $V - (A(0)^- + A(0)^+) = \{v_4, v_5, v_6, v_7\}$, $\theta_4^+ = 0.4$, $\theta_4^- = 0.5$, $\theta_5^+ = 0.6$, $\theta_5^- = 0.7$, $\theta_6^+ = 0.3$, $\theta_6^- = 0.2$, $\theta_7^+ = 0.5$, $\theta_7^- = 0.4$.

We will first set $A_k(0)^+ = \phi$. We know that the Companies whose budget has not been exhausted yet are C_1 and C_2. We will process all the nodes in $V - (A(0)^- + A(0)^+)$ for each Company.

Iteration 1:

Initially we start to process all the nodes in $v_i \in V - (A(0)^- + A(0)^+)$. We calculate the Amplification function value of v_4, v_5, v_6, v_7 for C_1 and C_2. The results are stored in the MAP data structure. MAP will have the following results for C_1:

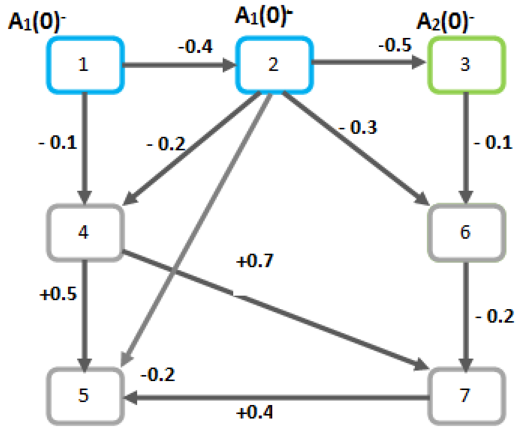
	v_4	v_5	v_6	v_7
C_1	-0.17	-0.11	-0.12	0.00

Table 1: Iteration 1: Company 1

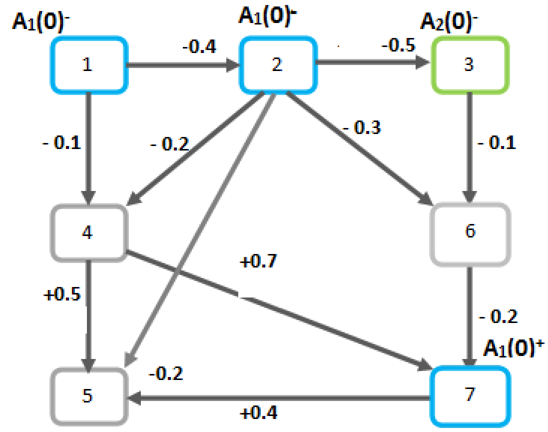
After traversing MAP, we observe that v_7 has more contribution towards C_1 than any other node as seen in Table 1. So, v_7 will be assigned as a positive seed to Company C_1 as shown in Figure 4(b).

Now we calculate the Amplification function of rest of the nodes for C_2 in the same way and we get the following results.

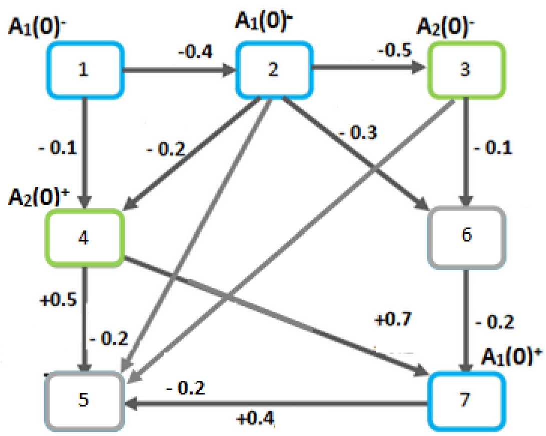
Now from Table 2. we see that, the nodes v_4 and v_5 have maximum and equal



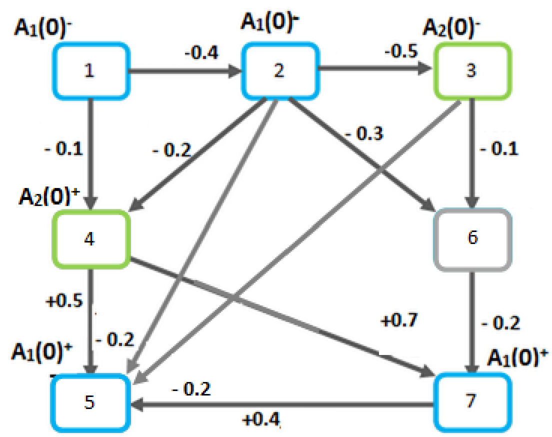
(a)



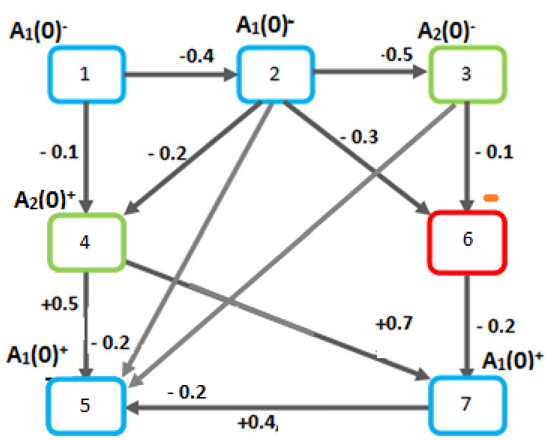
(b)



(c)



(d)



(e)

Figure 4: Amplification Function Demonstration

	v_4	v_5	v_6
C_2	0.00	0.00	-0.04

Table 2: Iteration 1: Company 2

Amplification Function value. So we break the tie by picking the node with maximum out-degree. We notice that the out-degree of v_4 is 1 and out-degree of v_5 is 0. So, v_4 with the more out-degree will be selected as a seed for C_2 as seen in Figure 4(c).

NOTE: *We are considering the maximum out-degree because we want to influence maximum number nodes. But if there are nodes with same out-degree then we use node ID to break the tie. Smaller node ID wins. For example, if node 4 and node 5 have the same out-degree, then node 4 wins.*

Here we notice that the budget of C_2 has been exhausted and we are only left with C_1.

We repeat the steps 2 to 9 of BNINS-GREEDY Algorithm because the budget of all the companies has not been exhausted yet. Now we, have these conditions: $A_1(0)^- = \{v_1, v_2\}$, $A_1(0)^+ = \{v_7\}$, $A_2(0)^- = \{v_3\}$, $A_2(0)^+ = \{v_4\}$, T $= \{C_1\}$, $b_1 = 4$, $b_2 = 2$, $V - (A(0)^- + A(0)^+) = \{v_5, v_6\}$.

Iteration 2:

We check the set T. Then we process all the nodes $v_i \in V - (A(0)^- + A(0)^+)$ for Company C_1 as shown in Figure 4(c). We calculate the Amplification Function value for every node and store it in the MAP$< v_i,\ f(v_i, C_k) >$. The results in the MAP will be as follows:

From the results in Table 3, we infer that node v_5 is contributing more towards Company C_1 than any other node. So, v_5 will be set as a positive seed node for C_1

	v_5	v_6
C_1	0.08	-0.12

Table 3: Iteration 2: Company 1

as shown in Fig.4(d). Here we observe that the budget of all the companies has been exhausted and set T is empty. So, the BNINS-GREEDY Algorithm terminates here.

Now we have received the initial adopters or seeds for every Company C_k. The next step is to implement the diffusion model in rest of the social network. As we can see in the Fig.4(d), we are left with a node v_6. We will implement the diffusion model on this node as discussed in Section 3.2 and find out its status.

PHASE 1:

We check the total positive and negative influence of v_6. If it is greater than or equal to its corresponding positive or negative activation threshold, then v_6 will become influenced with that influence. The total positive influence on v_6 is 0 as v_6 is not having any positive in-neighbours whereas the total negative influence on v_6 is $0.3 + 0.1 = 0.4$ which is greater than its negative threshold *i.e.* 0.2. So, v_6 is negatively influenced in Phase 1 as shown in Fig.4(e).

PHASE 2:

Now we know that v_6 is negatively influenced. Therefore it will be activated using p_6^- as discussed in Equation 5. For C_1, p_6^- is 0.75 and for C_2, p_6^- is 0.25. So, v_6 will be negative activated by C_1 with probability 75%, and be negative activated by C_2 with probability 25%.

CHAPTER V

PERFORMANCE EVALUATION

In addition to verifying the performance of our approximation algorithm, we are also interested in understanding its behaviour in practice, and comparing its performance to other heuristics for identifying influential individuals. We find that our greedy algorithm achieves significant performance gains over several widely-used structural measures of influence in social networks.

5.1 Simulation Setting

5.1.1 Real Online Social Data

For evaluation, it is highly desirable to use a network dataset that exhibits many of the structural features of large-scale social networks. We conduct simulations on Epinions dataset [30]. Epinions dataset has 76K nodes and 509K edges. We pre-processed the dataset to record the results by varying number of nodes. We use the BNINS-GREEDY algorithm to select the seed set S. Every edge is given positive and negative weights which are randomly generated. We randomly provide some initial budget (b_k) and the negative seed set ($A_k(0)^-$) per company. Below simulation results discusses the performance of our algorithm and comparison results.

5.1.2 Random Graphs

To test the performance of our algorithm, we build our own simulator to generate the random graphs based on random graph model $G(n,p) = \{G \mid G$ has n nodes, and an edge between any pair of nodes is generated with probability $p\}$. For $G = (V,E,W)$ $\in$ G(n,p), v_i, $v_j \in$ V, and $(v_i, v_j) \in E$, the associated social influence $0 < p_{ij} \leq 1$ is randomly generated. We also provide some random positive and negative weights to each edge, budget (b_k) per Company and the negative seed set ($A_k(0)^-$) per Company.

5.1.3 Comparison Setting

Currently there is no existing work studying the BNINS selection problem under the Linear Threshold model. The Simulation results of BNINS-GREEDY are compared to a related maximum blocking algorithm titled CLDAG algorithm [16] with our proposed BNINS-GREEDY algorithm. We used two metrics to compare the performance: number of positive activated and number of negative activated nodes with the time iterations. Then we check which Algorithm has more blocking effect.

5.2 Simulation Results

The objective of BNINS is to block the negative influence and hence maximize the positive influence propagation in the social networks. In this section, we use Random Graph and Real Online Social Data to check the performance of the BNINS GREEDY algorithm. We also compare our results with the related work proposed in [16]. We run the algorithm on a network size of 100 to 500 nodes and we obtained the following results.

5.2.1 Simulation Results of Real Social Data

In this section, we first compare our proposed algorithm with the related work and then followed by the comprehensive analysis of the performance of our BNINS-GREEDY algorithm.

Comparison with related work

In this subsection, we compare the performance of BNINS and CLDAG under real social data. We implemented the BNINS-GREEDY and CLDAG on 1000 nodes Epinions dataset with *one* Company (C_1), b_1=100 and $|A_1(0)^-|$=55. The comparison results considering two metrics *i.e.* total negative and total positive nodes with different time iterations are discussed as below:

- *Total Number of Negative Activated Nodes ($|A(t)^{-}|$):*

  Here we analyse the total negative influence using BNINS and CLDAG. From Figure 5, we can see that the number of negative activated nodes for both BNINS and CLDAG decreases when t increases. This is because more negative nodes are getting blocked as the network size increases. As the diffusion process goes on, more and more positive influence is spreading leading to the blockage of negative influence. Additionally, CLDAG produces more number of negative activated nodes. At *t=100*, the number of negative nodes for CLDAG is 50 whereas number of negative activated nodes for BNINS is 45. Similarly, *t=200*, the ratio of CLDAG to BNINS is 45:38. This is because in CLDAG, initially two Local Directed Acyclic Graphs (LDAGs) are constructed: $LDAG^{+}$ and $LDAG^{-}$ based upon the influence of nodes on all $v \in V$. Then, BFS is implemented to find the initial adopters or seeds by traversing $LDAG^{+}$ and $LDAG^{-}$ and one seed is picked per iteration to be put in positive or negative seed set based on its activation probability. Whereas in BNINS, we already have negative seed set and we just have to find positive seeds by analyzing the Amplification Function. So, the diffusion process starts in BNINS earlier than CLDAG and hence BNINS is blocking more nodes as compared to CLDAG at the same time. BNINS achieves the complete blocking effect at *t=900*, whereas CLDAG reaches this level at *t=1100*. So, we see that BNINS always had lesser number of negative activated nodes and also BNINS completely blocks all the negative effect long time before CLDAG. On average, BNINS has 12.5% better performance than CLDAG.

- *Total Number of Positive Activated Nodes ($|A(t)^{+}|$):*

  Here we analyse the total positive influence using BNINS and CLDAG Algorithm. We check the total number of positive nodes activating with the time

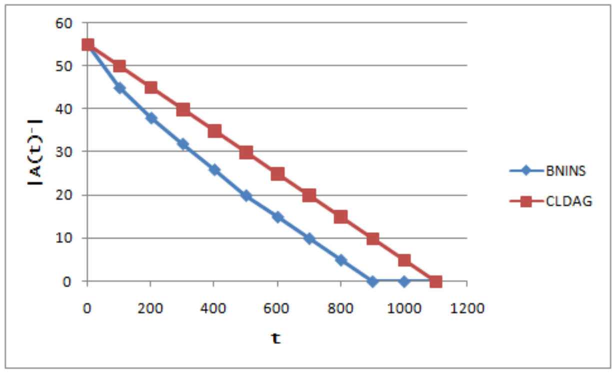


Figure 5: Total negative active nodes with time when n=1000, C_1 and $b_1 = 100$

iterations. In Figure 6, *x-axis* shows the time iteration t and *y-axis* shows the number of positive activated nodes. We see that the number of positive activated nodes for both BNINS and CLDAG increases with the time iterations. This is because more positive nodes are activated with the blockage of negative nodes as the network size increases. The reason is that, both the algorithm selects the most influential seed nodes that keeps spreading influence to more and more nodes. Additionally, BNINS produces more number of positive nodes. At *t=100*, the number of positive nodes for CLDAG is 50 whereas number of positive activated nodes for BNINS is 55. Similarly, *t=200*, the ratio of CLDAG to BNINS is 55:62. This trend goes on till *t=900*, when number of positive activated nodes for BNINS is 100 but CLDAG still have 92 positive activated nodes. So, we see that trend for BNINS is always above CLDAG. That means BNINS always had more positive nodes than CLDAG or we can say that BNINS is blocking more negative nodes than CLDAG. This is because BNINS starts long time before CLDAG, leading to blocking as much negative influence as possible and conversely leading to more and more positive influence. BNINS reaches the maximum number of positive activations at *t=900* by blocking all the negative influence whereas CLDAG do the same at *t=1100*. So, we see that BNINS always had more number of positive activated nodes and also BNINS achieves the maximum positive effect long before CLDAG. On average, BNINS has 7.6% better performance than CLDAG.

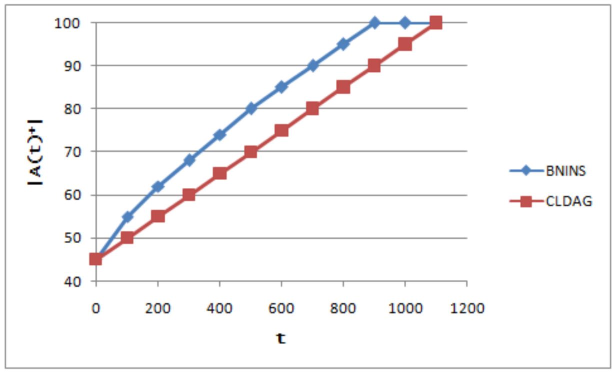


Figure 6: Total positive active nodes with time when n=1000, k=*1* and $b_1 = 100$

- *Performance Analysis of BNINS-GREEDY*

  From the above comparison results, we see that BNINS-GREEDY outperform CLDAG. BNINS-GREEDY always had more blocking effect than CLDAG which leads it to always having more number of positive activations and less number of negative activations than CLDAG. BNINS-GREEDY performed 12.5% better while blocking the negative influence and 7.6% better while maximizing the positive influence. Now we start to analysis our proposed method for multiply companies.

Number of Positive and Negative activated nodes with regard to all Companies

This section discusses the performance of the BNINS-GREEDY algorithm based upon the number of positively as well as negative activated nodes when the number of companies is greater than 1.

- *Number of Positive Activated nodes when k=2:*

  Here we discuss the number of positively activated nodes when all the nodes are processed or there are no more inactive nodes. We are considering 2 Companies

C_1 and C_2 here. The budget of each Company is $b_1 = 15$, $b_2 = 10$, $|A_1(0)^-|$=5, $|A_2(0)^-|$=3. The number of nodes in the network vary from 100 to 500. The results are shown in Figure 7. *x-axis* shows the varying number of nodes in the network and *y-axis* shows the number of positive activated nodes.

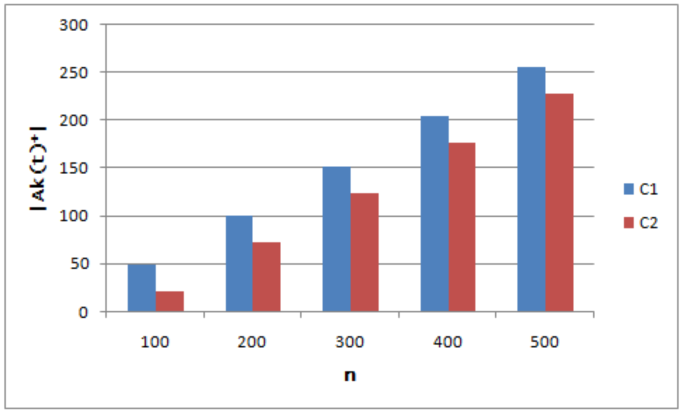


Figure 7: Positive active nodes when n = 100-500, k=2, $b_1 = 15$ and $b_2 = 10$

We see that when the number of nodes in the network is 100, number of positive activated nodes for C_1 is 48 and for C_2, its 20 at time iteration t=100. When n=200, the ratio of number of positive nodes in C_1 to number of positive nodes in C_2 is 100:72 at time iteration t=200. This trend keeps going up as the number of nodes in the network keeps on increasing. This is because, we are using the Amplification function to find the most influential nodes to be used as positive seed set depending upon the budget of the Company. And these positive seeds try to spread as much positive influence as possible. This influence spread keeps increasing as the number of nodes in the network increases. Hence we have the rising trend for positive activated nodes.

We also observe that the time iterations are also increasing with the increasing number of nodes. This is because we need to traverse each node in the network and apply BNINS-GREEDY or diffusion model on it. So more the number of nodes, more is the time to process them.

Not just this, we also see that C_1 always have more positive activated nodes than C_2. This is because the budget of C_1 is 15 which is greater than the budget of C_2 which is 10. Moreover, the number of positive seeds for C_1 which is 10, is also greater than number of positive seeds for C_2, which is 7. Hence C_1 always had more number of positive initial adopters which are influencing more number of nodes.

- *Number of Negative Activated nodes when k=2:*

  Here we discuss the number of negatively activated nodes when all the nodes in the social network are processed or their are no more inactive nodes left. We are considering 2 Companies C_1 and C_2 here. The budget of each Company is $b_1 = 15$, $b_2 = 10$, $|A_1(0)^-|$=5 and $|A_2(0)^-|$=3. The number of nodes in the network vary from 100 to 500. The results are shown in Fig.8. *x-axis* shows the number of nodes (n) and *y-axis* shows the number of negatively activated nodes.

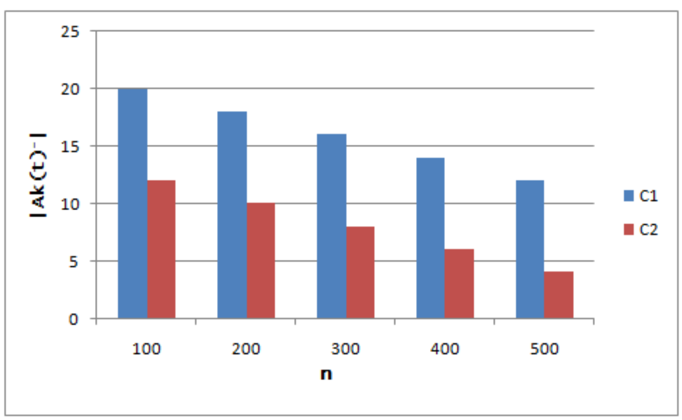


Figure 8: Negative active nodes when n = 100-500, k=*2*, $b_1 = 15$ and $b_2 = 10$

We see that when the number of nodes in the network is 100, number of negatively activated nodes for C_1 is 20 and for C_2, its 12 at time iteration t=*100*.

When $n=200$, the ratio of number of positive nodes in C_1 to number of positive nodes in C_2 is 18:10 at time iteration $t=200$. This trend keeps going down as the number of nodes in the network keeps on increasing. This is because, firstly we are picking the most influential nodes as positive seeds using Amplification function to spread maximum positive influence and block the negative influence. Secondly, in the diffusion model we pick the node to be negative or positive based upon its influence. So, having more positive influential nodes as seeds will mostly lead to positive influence. The negative influence keeps decreasing with every iteration because the positive activations keeps blocking the negative influence and a stage comes where all the negative influence is completely blocked. Hence we have the falling trend for positive activated nodes.

We also observe that the time iterations are increasing with the increasing number of nodes. This is because we need to traverse each node in the network and apply BNINS-GREEDY or diffusion model on it. So more the number of nodes, more is the time to process them. Also the number of negatively activated nodes are decreasing as the time iterations increases. This is because of the blocking effect as discussed above.

We see that C_1 has more negative nodes than C_2, this is because initially at $t=0$, C_1 had 5 negative activated seed nodes and C_2 had 3 negative activated seed nodes. So, C_1 has more negative influence as compared to C_2.

Total activated nodes vs. Time iterations

This section analyzes the BNINS-GREEDY algorithm based upon the number of activated nodes with the time for $n=500$, $k=1$, $b_1 = 100$ and , $|A_1(0)^-|=40$ as shown in Figure 9. *x-axis* represents the time iterations and *y-axis* represents number of activated nodes.

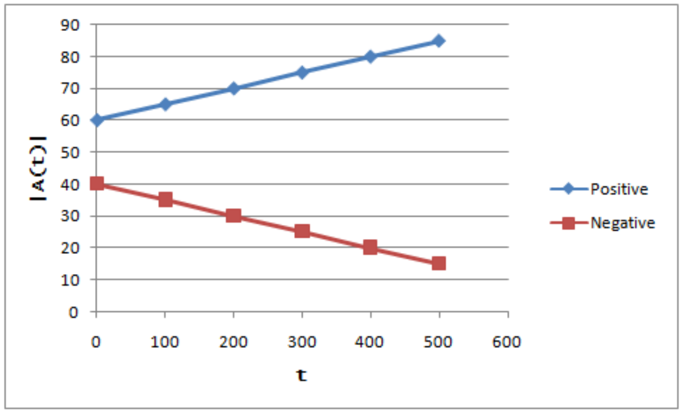


Figure 9: Number of activated nodes with time iterations when n=500, k=*1* and b_1= 100

We observe that, at t=*0*, total number of positive activations is 60 whereas total negative activations is 40. Then at t=*100*, positive activations increase to 65 and negative activations reduce to 35. Then at t=*200*, same trend follows giving us an upwards trend for positive and downward trend for negative. The reason for this is already discussed in the previous sections. We clearly see that the number of positive activated nodes are gradually increasing with the time and number of negative activated nodes is gradually decreasing with the time.

5.2.2 Simulation Results of Random Graph

In this section, we first compare our proposed algorithm with the related work and then followed by the comprehensive analysis of the performance of our BNINS-GREEDY algorithm.

Comparison with related work

In this subsection, we compare the performance of BNINS-GREEDY and CLDAG under Random Graph *G(n,p)*. We implemented the BNINS-GREEDY and CLDAG on 1000 nodes with One Company C_1, b_1=100 and $|A_1(0)^-|$=45. The comparison results considering two metrics *i.e.* total negative and total positive nodes with different time iterations are discussed as below:

- Total Number of Negative Activated Nodes($|A(t)^-|$):

  We analyse total negative influence using BNINS-GREEDY and CLDAG algorithm. From Figure 10, we observe that the number of negative activated nodes is decreasing as the number of time iterations increases. This is due to the blocking effect. More negative nodes are getting blocked as the time passes. We also notice that the number of negative activated nodes for CLDAG is always greater than BNINS. At *t=100*, the number of negative activated nodes for CLDAG is 44 whereas for BNINS-GREEDY, its 40. Similarly, at *t=200*, the ratio of CLDAG to BNINS is 43:37. The trends follows. This is because of the reason we discussed in the previous section. In CLDAG, it takes more time to select the initial adopters than BNINS-GREEDY and hence the diffusion process for BNINS-GREEDY starts early leading to more blocking effect than CLDAG at the same time iterations. We also notice the comparison trends for the real social data (Figure 5) and random graph (Figure 10) have one common thing and that is, falling trends. The differences arises due to the difference in network topology because for random graph, the topology is randomly generated. Random graph is less denser than real social data for which we are using real world topology. On average, BNINS has 17.22% times better performance than CLDAG.

- Total Number of Positive Activated Nodes ($|A(t)^+|$):

  In this section, we analyze the same for positively activated nodes. In Figure 11, *x-axis* represents the time iterations and *y-axis* represents the number of positively activated nodes. We notice the increase in number of positive activated nodes as the time iterations increases and the network size increases. Blocking

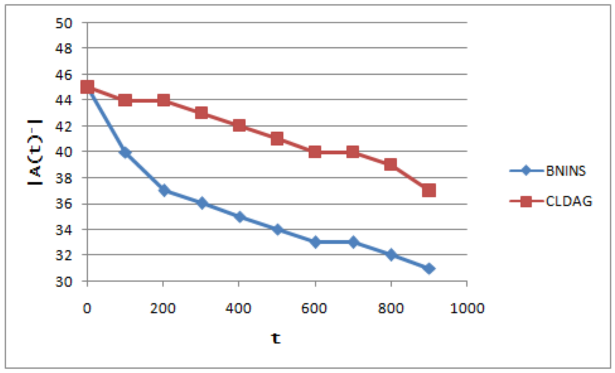


Figure 10: Total negative active nodes with time when n=1000, k=1 and $b_1 = 100$

effect leads to blocking the negative nodes and conversely increase the number of positive activated nodes with every iteration. Also, we see that the curve for BNINS-GREEDY is above the CLDAG curve. At *t=100*, when BNINS-GREEDY has 60 positive nodes, CLDAG has 56 positive activated nodes. Similarly, the trend follows for *t=200*, when the ratio is 63:57. The reason behind this already discussed in the previous section that BNINS-GREEDY starts the diffusion process before CLDAG and hence producing more positive nodes for the same corresponding time iterations. On average, BNINS has 5.9% better performance than CLDAG. If we compare the comparison results of BNINS-GREEDY and CLDAG for real social data in Figure 6 and Figure 11, we observe that number of positively activated nodes are increasing with the increase in time iterations. This is because the algorithm has been efficient in selecting most influential nodes as seed nodes which are increasing the spread influence. There is a slight difference in the number of nodes getting positively activated for both the figures. For *n=1000*, BNINS had 55 positively activated nodes for real input data but 60 for random graph. The reason behind that is the degree of nodes in random graph is greater than the degree of nodes in real online social data graph. For *n=100*, in real online social data, the maximum degree of node for node is 26 whereas for random graph, its 31. On average, BNINS has 5.9% times better performance than CLDAG.

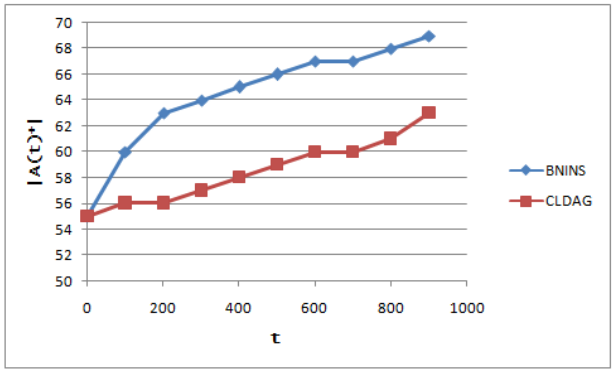


Figure 11: Total positive active nodes with time when n=1000, k=1 and $b_1 = 100$

- Performance Analysis of BNINS-GREEDY

  From the above comparison results on Random Graph, we see that BNINS-GREEDY outperform CLDAG. BNINS-GREEDY always had more blocking effect than CLDAG which leads it to always having more number of positive activations and less number of negative activations than CLDAG. Now we start to analysis our proposed method for multiply companies.

Number of Positive and Negative activated nodes with regard to all Companies

This section discusses the performance of the BNINS-GREEDY algorithm based upon the number of positively as well as negative activated nodes when the number of companies is greater than 1.

- Number of Positive Activated nodes when k=2:

  We consider two Companies C_1 and C_2 with budget b_1=20 and b_2=20, initial negative seed set $|A_1(0)^-|$=7 and $|A_2(0)^-|$=10. The number of nodes in the network varies from 100 to 500. The results are discussed in Figure 12. *x-axis* represents the number of nodes and *y-axis* represents the number of positively activated nodes with time.

We observe that, when *n=100*, number of positive activated nodes in C_1 is 48 and in C_2, its 20 at time iteration *t=100*. At *n=200*, the positive activated nodes ratio for C_1 to C_2 is 99:72 for *t=200*. For *n=300*, its 151:123 at *t=300*. So, we see that the number of positive activated nodes keep on increasing with the increasing size of the network. This is because seed nodes are picked by selecting the nodes having the most contribution towards company. This in turn lead to more positive influence spread.

Also, C_1 is always having more number of positive nodes than C_2 because positive initial adopters for C_1 is 13 which is more than for C_2 which is 10. Hence contributing to more positive activations.

The difference in topology for random graph and real social data has generated the difference in number of activated nodes as seen in Figure 7 and 12. More the density more the chances for number of activations. Regardless of topological difference, we see that the number of nodes is continuously increasing.

We also notice that, with the increase in time iteration, the number of positive activated nodes is gradually increasing. This is because the positive influence increases as the time and number of nodes increasing due to reason already discussed above.

- Number of Negative Activated nodes when k=2:

  Here we discuss the number of negatively activated nodes when all the nodes in the social network are processed or their are no more inactive nodes left. We are considering 2 Companies C_1 and C_2 here. The budget of each Company is $b_1 =$

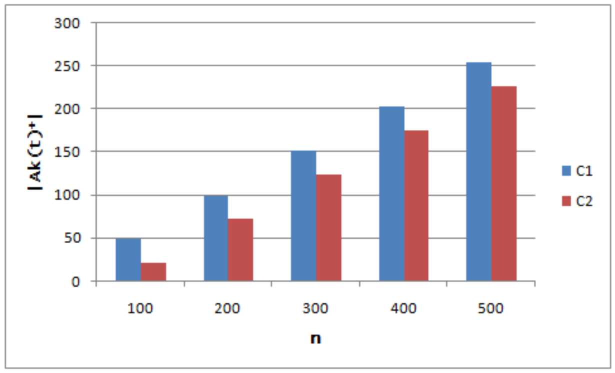


Figure 12: Positive active nodes when n=100-500, k=2, $b_1 = 20$, $b_2 = 20$, $|A_1(0)^-|$=7 and $|A_2(0)^-|$=10

20, $b_2 = 20$, $|A_1(0)^-|$=10 and $|A_2(0)^-|$=7. The number of nodes in the network vary from 100 to 500. The results are shown in Figure 13. *x-axis* represents the number of nodes and *y-axis* represents the number of negatively activated nodes.

We see that number of negative activated nodes keeps decreasing as the size of the network is increasing. At *n=100*, number of negative activated nodes in C_1 to C_2 is 17:15 at time iteration *t=100*. When *n=200*, the ratio is 16:13 at *t=200*. This is because negative influence is getting blocked as the diffusion process follows. The reason is that, BNINS-GREEDY picks the most influential set of nodes to be the positive seed set. The nodes in this seed set try to spread as much influence as possible. Plus the diffusion model activates the nodes based upon the influence on it. So we mostly get the positive influence and followed by the reduction in negative influence.

As compared to Figure 8, Figure 13 has less number of negatively activated nodes. The difference arises because of the topological changes. In the random graph, topology is randomly generated unlike the real online social data. For n=100, for real online social data, C_1 has 20 and C_2 has 12 negatively activated nodes whereas for random graph, C_1 has 17 and C_2 has 15 negatively activated nodes. The reason behind this is already discussed above. The random graph is less denser than the real online social network graph.

We also notice that, with the increase in time iteration, the number of negative activated nodes is dropping. This is because the negative nodes are getting blocked as the time and number of nodes increasing due to reason already discussed above.

Also, we see that C_1 has more negatively activated nodes than C_2. This is because C_1 had 10 negatively activated nodes at t=0 whereas C_2 had 7. 10 nodes have more influence than 7 nodes. Hence the difference arises.

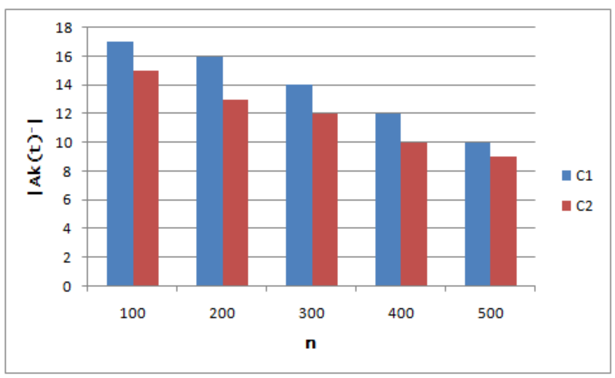


Figure 13: Negative active nodes when n=100-500, k=2, b_1 = 20, b_2 = 20, $|A_1(0)^-|$=10 and $|A_2(0)^-|$=7

Total activated nodes vs. Time iterations

This section analyzes the BNINS-GREEDY algorithm based upon the number of activated nodes with the time for *n=500*, *k=1*, b_1= 100 and $|A_1(0)^-|$=45 as shown in Figure 14. Here *x-axis* represents the different time iterations and *y-axis* represents the total number of activated nodes.

At t=0, total number of positive activated nodes is 55 and total number of negative activated nodes is 45. Then at t=100, positive activations increase to 60 and negative activations reduce to 40. Then at t=200, same trend follows giving us an upwards trend for positive and downward trend for negative. The reason for this is already discussed in the previous sections. We clearly see that the number of positive activated nodes are gradually increasing with the time and number of negative activated nodes is gradually decreasing with the time. For both Real Social Data (Figure 9) and Random Graph (Figure 14), the number of positive nodes increases with time iterations and number of negative activated nodes decreases with time iterations. The only difference is number of activations. This arises due to the density of the graph and the influences on the edges.

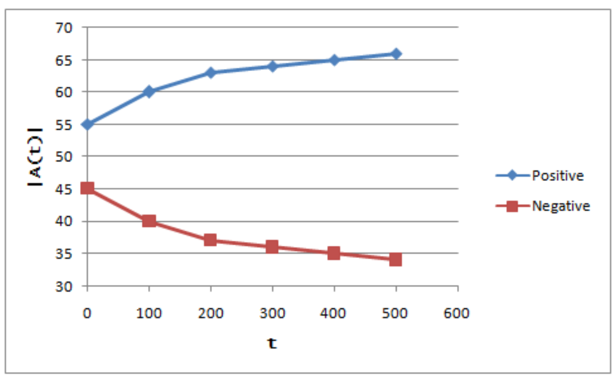


Figure 14: Number of activated nodes with time iterations when n=500, k=1 and b_1= 100

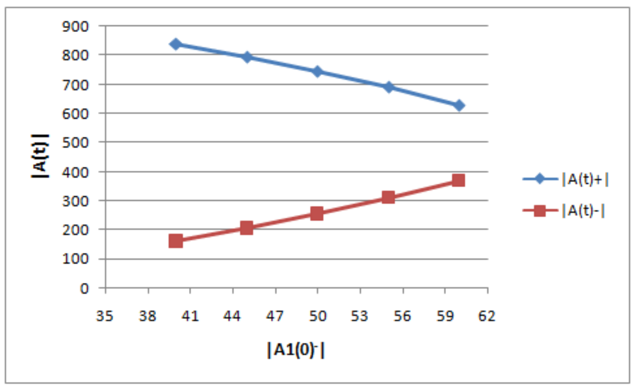
5.3 Performance Under Varying Input Parameters

In this section, we check the performance of BNINS-GREEDY algorithm under different values for the input parameters like budget and number of negative seeds.

5.3.1 Simulation Results for Smaller Network Size

Constant Budget, Changing the size of Negative Seeds set

The performance of BNINS-GREEDY is analysed by keeping the budget of the company constant but varying the size of Negative Seed Set. We perform the simulations for this with n=1000, k=1, b_1=100. Now, for the constant budget value 100, we plot a chart that shows how the different size of negative seed set affects the output results. In the chart 15, *x-axis* represents the number of negative seeds and *y-axis* represents the total number of activated nodes.



(a)

Figure 15: Number of activated nodes when *n=1000*, *k=1* and b_1= 100

From Fig.15, we analyse that, when $A_1(0)^-$ =40, the total number of positive activated nodes is 840 and the total number of negative activated nodes is 160. When $A_1(0)^-$ =45, $A_1(t)^+$ =840 and $A_1(t)^-$ =160. We see that no matter how many number of negatively activated seeds are there, BNINS-GREEDY always achieves maximum positive spread. This is because of effectively picking most influential positive seed nodes. We also notice that when $A_1(0)^-$ =50 and $A_1(0)^+$ =50, then also the positive spread is 745 and the negative spread is 255. So, for equal number of positive and negative initial adopters also, BNINS-GREEDY achieves more number of positive activated nodes than the negative activated nodes because of efficient seed selection.

Constant size of Negative Seed Set, Changing Budget

We discuss the total number of negative and positive activated nodes when all the nodes in the network are processed. Here we keep the size of negative seed set fixed and we vary the budget which will ultimately lead to change in the size of positive seed set as well because the budget is the sum of number of negative and positive seeds. We have the network with n=1000, k=1 and $A_k(0)^{-}$=40. In Fig.16 *x-axis* represents the budget and *y-axis* represents the total number of activated nodes.

(a)

Figure 16: Number of activated nodes when *n=1000*, *k=1* and $A_k(0)^{-}$=40

We see when the budget is 80, $A_1(t)^{+}$ =800 and $A_1(t)^{+}$ =200. When budget is 100, $A_1(t)^{+}$ =840 and $A_1(t)^{+}$ =160. So we observe that the number of positive activated nodes gradually keep on increasing and the number of negative activated nodes keeps on decreasing. This is because the size of negative activated seed set remains constant but the budget keeps on increasing which impact the size of positive seed set leading to the increase in the number of positive initial adopters. This in turn lead to more positive influence and also have more blocking effect because of the reason already discussed in the previous sections. The results are concluded in the following table:

n	b_1	$\lvert A_1(0)^- \rvert$	$\lvert A_1(0)^+ \rvert$	$\lvert A_1(t)^+ \rvert$	$\lvert A_1(0)^- \rvert$
1000	100	40	60	840	160
1000	100	45	55	795	205
1000	100	50	50	745	255
1000	100	55	45	690	310
1000	100	60	40	630	370
1000	80	40	40	800	200
1000	100	40	60	840	160
1000	120	40	80	866	134
1000	140	40	100	885	115
1000	160	40	120	900	100

(a)

Figure 17: Number of activated nodes when *n=1000*, *k=1*

5.3.2 Simulation results for Larger Network Size

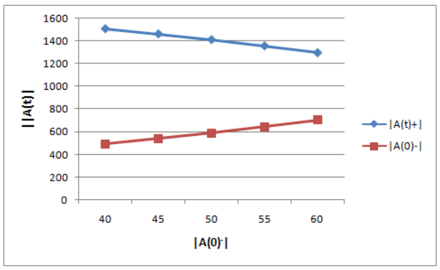
In this section, we show the results of simulations for large size network by varying the values of input parameters. Figure 5.3.2 shows the output results for network size varying from 2000 to 20,000 nodes. After observing all the results, it confirms that no matter how big the network size is, we always receive the maximum positive spread. Additionally, the negative influence is always blocked resulting in lesser number of negative activated nodes.

5.4 Performance Summary

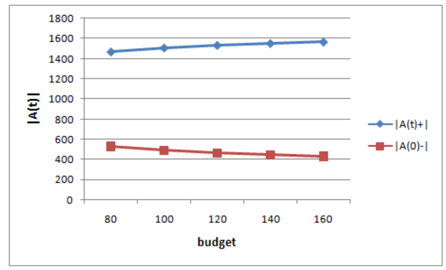
After conducting the simulations and testing the algorithm on different parameters, we observe that BNINS-GREEDY achieves its objective by minimizing the number of negative activated nodes and hence maximizing the number of positive activated nodes. Also, BNINS-GREEDY has been generalized to run for k competing companies. We compared the results of BNINS-GREEDY with CLDAG [16]. On average, in terms of total number of negative activated nodes, for real online social data, BNINS-GREEDY achieved 12.5% better performance than CLDAG and for random graph, BNINS-GREEDY achieved 17.22% better performance than CLDAG. Similarly, for total number of positive activated nodes, on average, BNINS-GREEDY achieved

n	b_1	$\|A_1(0)^-\|$	$\|A_1(0)^+\|$	$\|A_1(t)^+\|$	$\|A_1(0)^-\|$
2000	100	40	60	1508	492
2000	100	45	55	1463	537
2000	100	50	50	1413	587
2000	100	55	45	1358	642
2000	100	60	40	1298	702
2000	80	40	40	1468	532
2000	100	40	60	1508	492
2000	120	40	80	1534	466
2000	140	40	100	1553	447
2000	160	40	120	1568	432

(a)



(b)

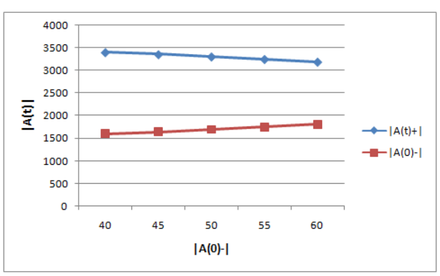


(c)

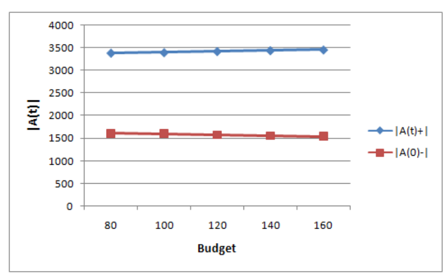
Figure 18: n=2000, k=1

n	b_1	$\|A_1(0)^-\|$	$\|A_1(0)^+\|$	$\|A_1(t)^+\|$	$\|A_1(0)^-\|$
5000	100	40	60	3399	1601
5000	100	45	55	3354	1646
5000	100	50	50	3304	1696
5000	100	55	45	3249	1751
5000	100	60	40	3189	1811
5000	80	40	40	3389	1611
5000	100	40	60	3399	1601
5000	120	40	80	3425	1575
5000	140	40	100	3444	1556
5000	160	40	120	3459	1541

(a)



(b)

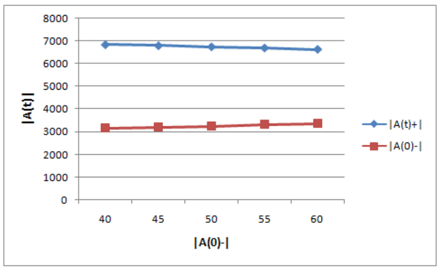


(c)

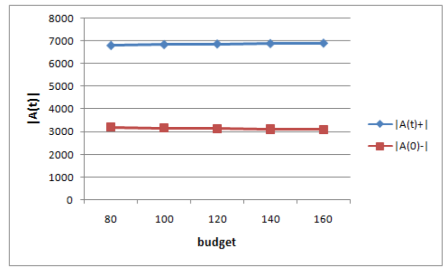
Figure 19: n=5000, k=1

n	b_1	$\|A_1(0)^-\|$	$\|A_1(0)^+\|$	$\|A_1(t)^+\|$	$\|A_1(0)^-\|$
10000	100	40	60	6841	3159
10000	100	45	55	6796	3204
10000	100	50	50	6746	3254
10000	100	55	45	6691	3309
10000	100	60	40	6631	3369
10000	80	40	40	6801	3199
10000	100	40	60	6841	3159
10000	120	40	80	6867	3133
10000	140	40	100	6886	3114
10000	160	40	120	6901	3099

(a)



(b)

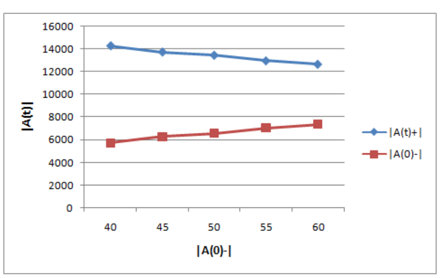


(c)

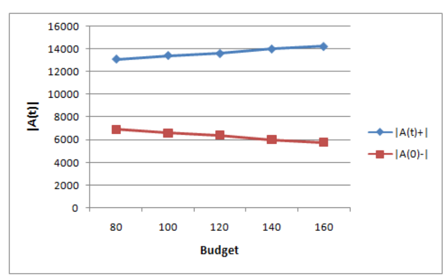
Figure 20: n=10,000, k=1

n	b_1	$\|A_1(0)^-\|$	$\|A_1(0)^+\|$	$\|A_1(t)^+\|$	$\|A_1(0)^-\|$
20000	100	40	60	13131	6869
20000	100	45	55	13086	6914
20000	100	50	50	13036	6964
20000	100	55	45	12961	7039
20000	100	60	40	12901	7099
20000	80	40	40	13091	6909
20000	100	40	60	13131	6869
20000	120	40	80	13157	6843
20000	140	40	100	13176	6824
20000	160	40	120	13191	6809

(a)



(b)



(c)

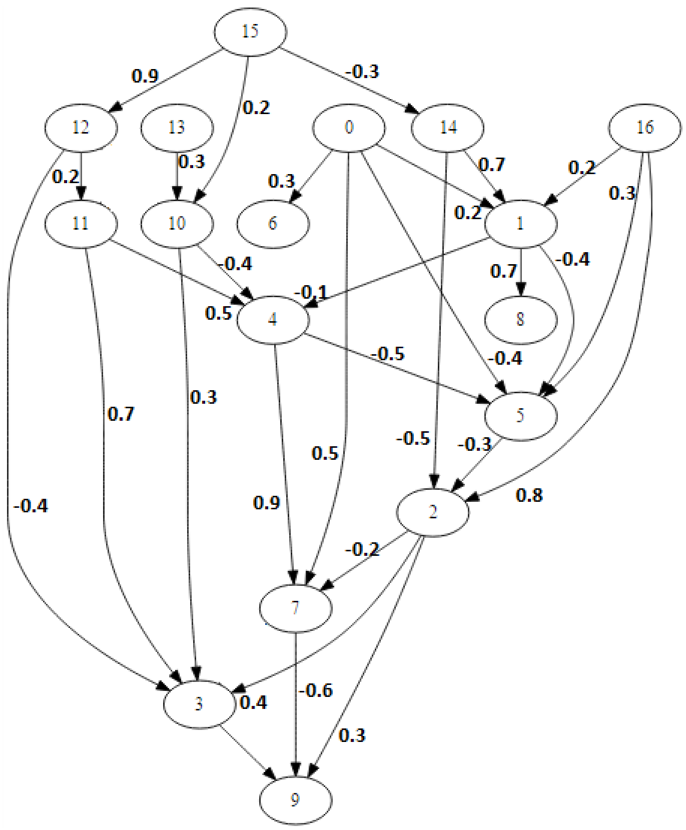
Figure 21: n=20,000, k=1

5.9% better performance than CLDAG for real online social data and 7.6% for random graph. We also evaluated the performance of BNINS-GREEDY for multiple companies and achieved the desired results. Apart from this, we analysed BNINS-GREEDY's performance with varying number of nodes from 100-500 in the networks and we observed that, the number of positive activated nodes are increasing with the time iterations and number of negative nodes is gradually decreasing with the time iterations. So, BNINS-GREEDY achieved its objective by minimizing the number of negative activated nodes and hence maximizing the number of positively activated nodes.

CHAPTER VI

VALIDATION OF SIMULATION

This section applies the proposed work on the graph and finds out the status of every node. The graph is shown as follows:



(a)

Figure 22: BNINS Application

Input: n=17, T = $\{C_1, C_2\}$, $b_1 = 3$, $b_2 = 4$, $|A_1(0)^-| = 1$, $A_1(0)^- = \{0\}$, $|A_2(0)^-| = 2$, $A_2(0)^- = \{v_{15}, v_{16}\}$. Table 4 shows the threshold values for every node.

Now we will first find out the most influential nodes in the network. We process every inactive node and perform BNINS-GREEDY on it.

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
θ_i^+	0.3	0.4	0.7	0.3	0.4	0.2	0.4	0.3	0.5	0.6	0.4	0.1	0.3	0.6	0.2	0.1	0.3
θ_i^-	0.8	0.2	0.6	0.6	0.3	0.6	0.3	0.5	0.6	0.7	0.4	0.3	0.5	0.1	0.4	0.8	0.7

Table 4: Threshold Values

ITERATION 1:

We first find out the amplification values for every node for Company C_1. Table 5 shows the results.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$f(v_i, C_1)$	0.13	0.37	0	0	-0.08	0.13	0.31	0	0	0.1	0	0.56	0	-0.2

Table 5: Iteration 1, Company C_1

We see that v_{12} has most contribution towards Company C_1, so it will be chosen as a positive seed node for C_1. Now, we check the budget of C_1. Now we calculate the same for C_2.

	1	2	3	4	5	6	7	8	9	10	11	13	14
$f(v_i, C_2)$	0.13	0.37	-0.13	0	-0.08	0.13	0.31	0	0	0.1	0.05	0	-0.2

Table 6: Iteration 1, Company C_2

We observe that v_2 has most contribution towards Company C_2, so it will be chosen as a positive seed node for C_2. Now, we check the budget of C_1 and C_2. None of themś budget has been exhausted. So we perform the next iteration.

ITERATION 2:

The new Amplification function values for C_1 are shown in Table 7.

We observe that v_7 has more contribution towards C_1 than any other node. So, it will be selected as a positive seed for C_1. Now we check the same for C_2 and we get the results shown in Table 8.

	1	3	4	5	6	7	8	9	10	11	13	14
$f(v_i, C_1)$	0.13	0	0	-0.08	0.13	0.39	0	0.14	0.1	0.05	0	-0.2

Table 7: Iteration 2, Company C_1

	1	3	4	5	6	8	9	10	11	13	14
$f(v_i, C_2)$	0.13	0	0	-0.08	0.13	0	-0.28	0.1	0.05	0	-0.2

Table 8: Iteration 2, Company 2

v_1 will be selected as a positive seed for C_2 because of maximum contribution and more out-degree.

Now the budget of Company C_1 and C_2 has been exhausted and we have received the initial adopters or seeds for both companies. We are left with nodes v_3, v_4, v_5, v_6, v_8, v_9, v_{10}, v_{11}, v_{13}, v_{14}. The next step is to implement the diffusion model on rest of the social network. We will implement the diffusion model on these nodes as discussed in Section 3.2 and find out its status.

For v_3:

PHASE 1:

- Total Positive Influence is 0.8
- Total Negative Influence is 0.

The total positive influence of v_3 is greater than its positive threshold. So, v_3 will be positive influenced.

PHASE 2:

v_3 will be activated with C_1 with p_3^+=0.5.

For v_4:

Phase 1:

- Total Positive Influence is 0.1
- Total Negative Influence is 0

Here, both the total positive and negative influence is less than the corresponding positive and negative threshold of v_4. Therefore, v_4 will stay uninfluenced for now.

For v_5:

Phase 1:

- Total Positive Influence is 0.4
- Total Negative Influence is 0.7

Here both influences are greater than corresponding thresholds. So,v_5 will be considered as negatively influenced.

Phase 2: Here v_5 will be activated with C_1 with p_5^-= 0.63 as per Equation (5).

For v_6:

Phase 1:

- Total Positive Influence is 0.3
- Total Negative Influence is 0.

Here the total positive influence is equal to v_6ś positive threshold. So v_6 will be positively influenced.

Phase 2:

Here v_5 will be activated with C_1 with p_6^+= 1 as per Equation (5).

For v_8:

Phase 1:

- Total Positive Influence is 0.7.
- Total Negative Influence is 0.

Here the total positive influence is greater than to v_8ś positive threshold. So v_8 will be positively influenced.

PHASE 2:

Here v_8 will be activated with C_2 with $p_8^+ = 1$ as per Equation (5).

For v_9:

PHASE 1:

- Total Positive Influence is 1.4.
- Total Negative Influence is 0.

Here the total positive influence is greater than to v_9ś positive threshold. So v_9 will be positively influenced.

PHASE 2:

Here v_9 will be activated with C_1 with $p_9^+ = 0.79$ as per Equation (5).

For v_{10}:

PHASE 1:

- Total Positive Influence is 0.
- Total Negative Influence is 0.

So, this node will stay uninfluenced or inactivated for now. The same is the case will be for v_{11}, v_{13}.

NOTE: v_{13} doesnt́ have any in-neighbours to influence them. So, it will always stay uninfluenced or unaffected.

For v_{14}:

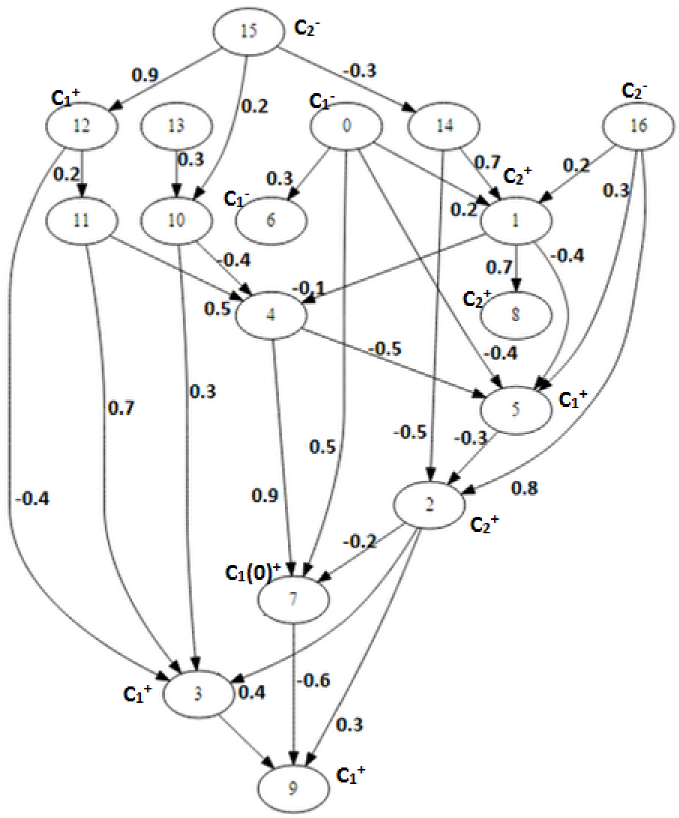
Phase 1:

- Total Positive Influence is 0.
- Total Negative Influence is 0.3

Here, both the total positive and negative influence is less than the corresponding positive and negative threshold of v_4. Therefore, v_4 will stay uninfluenced for now.

We will re-iterate through the rest of the nodes i.e., v_4, v_{10}, v_{11} and v_{14} until all the nodes until no more node can be activated. After processing these nodes we see that these nodes will stay uninfluenced because these doesnt́ have sufficient influential relationships with activated nodes. Additionally the graph is not denser enough. But we see in Figure 23 that the total number of positively activated nodes are always greater than the number of negatively activated nodes and hence the objective is achieved. In figure, C_1^+ denotes the positively activated nodes for Company C_1 and C_1^- denotes the negatively activated nodes with Company C_1. Similarly with C_2.

Now, we run the simulations for Figure 22 (n=17), by varying the input parameters and the results are shown in Figure 24. From there we conclude that no matter what the size of the network is, BNINS-GREEDY always try to achieve the maximum positive spread and at the same time minimizing the negative spread of influence.

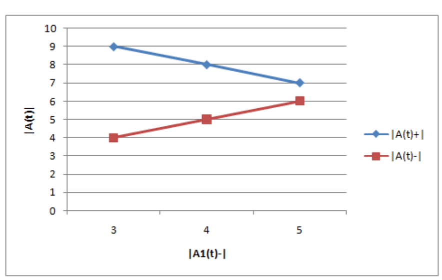


(a)

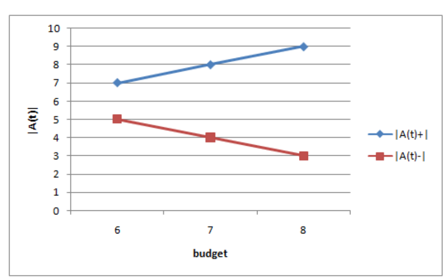
Figure 23: Simulation Validation

n	b	$\|A_1(0)^-\|$	$\|A_1(0)^+\|$	$\|A_1(t)^-\|$	$\|A_1(0)^+\|$
17	7	3	4	4	9
17	7	4	3	5	8
17	7	5	2	6	7
17	6	3	3	5	7
17	7	3	4	4	8
17	8	3	5	3	9

(a)



(b)



(c)

Figure 24: n=17, k=1

CHAPTER VII

CONCLUSION AND FUTURE WORK

In this thesis work, we proposed a new diffusion model considering both positive and negative influences and applied it to solve the Blocking Negative Influential Node Set (BNINS) Selection problem from host perspective, which will be very useful for promoting the products in marketing applications in social networks. We formally define the BNINS Selection problem and proposed a BNINS-GREEDY algorithm to solve it. We validated the proposed algorithm through simulations on random graphs and real online social data. We compared the performance of BNINS-GREEDY with the related work [16] and analysed that BNINS-GREEDY always achieved its objective earlier than CLDAG and that too with great proficiency. Using the random graph, on average, BNINS-GREEDY achieved 17.22% better performance than CLDAG for blocking the negative influence and 5.9% better performance than CLDAG for maximizing the positive influence. Similarly, using the real online social data, on average, BNINS-GREEDY achieved 12.5% better performance than CLDAG for blocking the negative influence and 7.6% better performance than CLDAG for maximizing the positive influence. Apart from that, we tested BNINS-GREEDY on various performance metrics including total number of positive and negative activations with varying number of nodes, varying number of companies and varying time iterations etc. After conducting all these tests, we analyzed the test results and found out that BNINS-GREEDY has achieved its objective with good and favourable results. As a future work, I would like to perform Theoretical Analysis on it and validate the simulation by looking into the order of magnitude for the algorithm.

REFERENCES

[1] D. Kempe, J. M. Kleinberg, and E. Tardos. *Maximizing the spread of influence through a social network.* In Proceedings of the 9th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 137146, 2003.

[2] D. Kempe, J. M. Kleinberg, and E. Tardos. *Influential nodes in a diffusion model for social networks* . In Proceedings of the 32nd International Conference on Automata, Languages, and Programming, pages 11271138, 2005.

[3] M. Kimura and K. Saito. *Tractable models for information diffusion in social networks.* In Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases, pages 259271, 2006.

[4] J. Leskovec, A. Krause, C. Guestrin, C. Faloutsos, J. Van- Briesen, and N. S. Glance. *Cost-effective outbreak detection in networks.* In Proceedings of the 13th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pages 420429, 2007.

[5] R. Narayanam and Y. Narahari. *A shapley value based approach to discover influential nodes in social networks.* IEEE Transactions on Automation Science and Engineering, 2010. to appear. Online version available at http://clweb.csa.iisc.ernet.in/nrsuri/social-networks-nrsuri-2010.pdf.

[6] W. Chen, Y. Wang, and S. Yang. *Efficient influence maximization in social networks.* In Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2009.

[7] W. Chen, C. Wang, and Y. Wang. *Scalable influence maximization for prevalent viral marketing in large scale social networks*. In Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2010.

[8] Y. Wang, G. Cong, G. Song, and K. Xie. *Community-based greedy algorithm for mining top-k influential nodes in mobile social networks*. In Proceedings of the 16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2010.

[9] W. Chen, Y. Yuan, and L. Zhang. *Scalable influence maximization in social networks under the linear threshold model*. In Proceedings of the 10th IEEE International Conference on Data Mining, 2010.

[10] M. Granovetter. *Threshold Models of Collective Behaviours*. American Journal of Sociology, 83(6):1420, 1978.

[11] T.C. Schelling. *Micromotives and Microbehaviour*. W.W. Norton and Company, 1978.

[12] J. Goldenberg, B. Libai, and E. Muller. *Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata*. Academy of Marketing Science Review, 2001(9):1-18, 2001.

[13] S. Bharathi, D. Kempe, and M. Salek. *Competitive influence maximization in social networks*. In WINE07.

[14] C. Budak, D. Agrawal, and A. El Abbadi. *Limiting the spread of misinformation in social networks*. In WWW11.

[15] T. Carnes, C. Nagarajan, S. M. Wild, and A. van Zuylen.*Maximizing influence in a competitive social network: a followers perspective*. In ICEC07.

[16] X. He, G. Song, W. Chen, and Q. Jiang. *Influence blocking maximization in social networks under the competitive linear threshold model*. In SDM12.

[17] Wei Lu, Fransesco Bonchi, Amit Goyal and Laks V.S. Lakshmanan. *The Bang for the Buck: Fair Competitive Viral Marketing from the Host Perspective*. In proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 928-936, 2013.

[18] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. *An analysis of approximations for maximizing submodular set functions - i. Mathematical Programming*, 14(1): 265294, 1978.

[19] M. Richardson, P. Domingos. *Mining Knowledge-Sharing Sites for Viral Marketing*. In ACM 2002.

[20] P. Domingos, M. Richardson. *Mining the Network Value of Customers*. Seventh International Conference on Knowledge Discovery and Data Mining, 2001.

[21] P. Rozin and E. B. Royzman. *Negativity bias, negativity dominance,and contagion*. Personality and Social Psychology Review, 5(4):296320, 2001.

[22] F. F. Reichheld. *The one number you need to grow*. Harvard

[23] G. Peeters and J. Czapinski. *Positive-negative asymmetry in evaluations: The distinction between affective and informational negativity effects*. European Review of Social Psychology, 1:3360, 1990.

[24] S. E. Taylor. *Asymmetrical effects of positive and negative events: The mobilization-minimization hypothesis*. Psychological Bulletin, 110(1):6785, 1991.

[25] R. F. Baumeister, E. Bratslavsky, and C. Finkenauer. *Bad is stronger than good.* Review of General Psychology, 5(4):323370, 2001. Business Review, Dec. 2003.

[26] M. Eftekhar, Y. Ganjali, N. Koudas. *Information Cascade at Group Scale.* In proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining. Pages 401-409. 2013.

[27] Y. Li, W. Chen, Y. Wang, Z. Zhang. *Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships.* WSDM '13 Proceedings of the sixth ACM international conference on Web search and data mining. Pages 657-666. ACM New York, NY, USA 2013

[28] Z. wang, Z. Qian, S. Lu.*A probability based algorithm for influence maximization in social networks.*Internetware '13 Proceedings of the 5th Asia-Pacific Symposium on Internetware. Article No. 12. ACM New York, NY, USA,2013

[29] Jing (Selena) He, Shouling Ji, Xiaojing Liao, Hisham Haddad, and Raheem Beyah. *Minimum-sized Positive Influential Node Set Selection for Social Networks: Considering Both Positive and Negative Influences.* In IPCCC 2013. San Diego, California, USA. December 6-8, 2013.

[30] Epinions Dataset: http://snap.stanford.edu/data/soc-Epinions1.html