ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ • ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING • UNIVERSTITY OF IOANNINA

The Effect of Antagonism on Information Propagation in Social and Technological Networks

Η ΜΕΤΑΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ ΕΞΕΙΔΙΚΕΥΣΗΣ

υποβάλλεται στην ορισθείσα από την
Γενική Συνέλευση Ειδικής Σύνθεσης του
Τμήματος Μηχανικών Η/Υ και Πληροφορικής
Εξεταστική Επιτροπή

από την

Ελευθερία Λιούκα

ως μέρος των Υποχρεώσεών της για τη λήψη του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΔΙΠΛΩΜΑΤΟΣ ΣΤΗΝ ΠΛΗΡΟΦΟΡΙΚΗ
ΜΕ ΕΞΕΙΔΙΚΕΥΣΗ ΣΤΗ
ΘΕΩΡΙΑ ΕΠΙΣΤΗΜΗΣ ΥΠΟΛΟΓΙΣΤΩΝ

Επιβλέπων Καθηγητής : Σπύρος Κοντογιάννης

Ιωάννινα, Ιούλιος 2016

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
Τ.Θ. 1186, ΙΩΑΝΝΙΝΑ 45110
+30 265100 8817
http://www.cse.uoi.gr/

DEPT. OF COMPUTER SCIENCE & ENGINEERING
UNIVERSITY OF IOANNNINA
P.O. BOX 1186, IOANNINA,
GREECE, GR-45110
+30 265100 8817
http://www.cse.uoi.gr/

Περίληψη

Τα κοινωνικά δίκτυα, όσον αφορά τις ανθρώπινες αλληλεπιδράσεις, έχουν αποκτήσει μεγάλη σημασία στις μέρες μας. Για αυτό τον λόγο, διάφορες μορφές πληροφορίας μπορούν να διαδοθούν μέσα από αυτά, όπως για παράδειγμα η διαφήμιση ενός προϊόντος, η εξάπλωση μιας άποψης κτλ. Είναι σημαντικό λοιπόν να μπορούμε να εντοπίζουμε τους 'σημαντικούς' κόμβους μέσα σε ένα τέτοιο δίκτυο, δηλαδή εκείνους τους κόμβους που μπορούν να επηρεάσουν πολλούς άλλους. Για να το επιτύχουμε αυτό, χρειαζόμαστε κάποιες μετρικές που να ξεχωρίζουν τέτοιους κόμβους μέσα στο δίκυο. Κάποιες από αυτές μπορούν να είναι οι μετρικές Betweeness Centrality, Closeness Centrality, ή Degree Centrality. Ωστόσο, υπάρχουν πολλές και διαφορετικές τέτοιες μέθοδοι στη βιβλιογραφία, για να εξετάσει κάποιος.

Στην παρούσα εργασία μελετάται η επίδραση του ανταγωνισμού στη μεγιστοποίηση της επιρροής μέσα στο δίκτυο. Θεωρούμε ότι υπάρχουν δυο ανταγωνιστικές εταιρείες που προσπαθούν να διαφημίσουν το προϊόν τους μέσα στο ίδιο δίκτυο. Για τον σκοπό αυτό, η κάθε μία διαλέγει ανεξάρτητα από την άλλη, μια από τις μετρικές σημαντικότητας, και δημιουργεί μια φθίνουσα σειρά των ατόμων (αντιπροσωπεύονται από κόμβους μέσα στο κοινωνικό δίκτυο) με βάση το επίπεδο επιρροής τους σε άλλα άτομα. Έπειτα, οι δυο φίρμες διαλέγουν τους πρώτους k κόμβους από αυτές τις λίστες και εφαρμόζεται μια μέθοδος που επιλύει τις συγκρούσεις, δηλαδή ορίζει ποια εταιρεία θα κρατήσει έναν κόμβο στην περίπτωση που τον έχουν επιλέξει και οι δυο για να τις αντιπροσωπεύσει. Τέλος, τα δυο τελικά σύνολα αρχικών κόμβων επιρροής για τις δυο εταιρείες ανταγωνίζονται για την διάδοση των προϊόντων που αντιπροσωπεύουν, με βάση δύο μοντέλα προσομοίωσης της διάχυσης πληροφορίας (μοντέλα De Groot και Threshold) σε ένα κοινωνικό ή τεχνολογικό δίκτυο. Στο τέλος της διαδικασίας, μπορούμε να αποφανθούμε για το ποια από τις δυο εταιρείες πέτυχε τη μεγαλύτερη διείσδυση του προϊόντος της στην αγορά, δηλαδή, τη μεγαλύτερη εξάπλωση του προϊόντος της στο δίκτυο, και κατά πόσο και οι δυο εταιρείες επηρεάστηκαν τελικά από το γεγονός της ύπαρξης του ανταγωνισμού μεταξύ τους. Στο πλαίσιο της παρούσας εργασίας διεξήχθησαν εκτενή πειράματα σε πραγματικά power law κοινωνικά και τεχνολογικά δίκτυα, τα οποία κατέγραψαν μια λεπτομερή εικόνα για την επίδραση του ανταγωνισμού στη διάχυση πληροφορίας.

Abstract

The possession of massive digital records of human interactions (e.g., email or file exchanges, friendships or acquaintances, collaborations, phone calls, etc) in recent years provides a new system-wide perspective on social networks. In addition to observing and predicting patterns of collective human behavior, in many cases the dynamics of the network can be engineered. One such example is when attempting to initiate a large cascade by seeding it at certain ''influential'' nodes in the network to promote a novel idea or, a new product through word-of-mouth. The algorithmic challenge of selecting individuals who can serve as early adopters in a manner that will trigger a large cascade in the social network, is known as influence maximization. One might consider several metrics which estimate the ''importance'' of a node in a network. For example, it is well known that high-degree nodes are indeed quite influential in the network. Other popular metrics, such as Betweeness Centrality, Closeness Centrality, or Eigenvector Centrality, might also be considered as good candidates. Moreover, one might consider choosing seeds adaptively, e.g., by choosing the next seed given the subset of already chosen seeds so far.

This thesis studies the effect of competition in influence maximization. We consider a scenario in which two competing firms try to advertise their own product within the same social network. The firms are assumed to assess the importance of the nodes in the network, as ''influencers'' of other nodes towards buying the same product. Each of them selfishly chooses an importance-assessment strategy, i.e., a metric (from a predetermined set of available metrics) to order the nodes in the network. Due to limitations of their advertising budget, each firm is assumed to be able to seed at most k nodes. Therefore, given the importance-orders of the two firms, a seed-selection policy determines the k most important seeds per firm. We then consider two different information-propagation models (De Groot and Threshold) to simulate the dispersion of the new products in the entire network. The measure of success per firm is its own degree of penetration in the market, i.e., the fraction of nodes which will eventually adopt the firm's product. We have implemented algorithms for computing (exactly or in approximation, statically or adaptively) several quite popular importance-assessment metrics. We have also conducted extensive experimental evaluation of this scenario, in order to assess the validity of each importance assessment strategy, under the lens of competition

between the two firms, in several benchmark data sets which are widely used in the literature.

Thanks

At this point, it is essential to express gratitude to those who helped me in several ways, directly or indirectly in the writing of this paper.

I would first like to thank my thesis advisor Kontogianni Spyro of the Department of Computer Engineering and Informatics at University of Ioannina. He was there whenever I ran into a difficulty or had a question about my research or writing. He consistently allowed this paper to be my own work, but steered me in the right direction whenever he thought I needed it.

I would also like to thank phd candidate Athanasiou Naso who has been by my side for the last 2 years, resolving any questions i had patiently, and I am gratefully indebted to his valuable comments on this thesis.

Finally, I must express my very profound gratitude to my parents and to my fiance for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them. Thank you.

Contents

Chapter 1

Introduction

One of the fundamental problems in social network analysis is to determine the significance of the nodes in a network and, based on that importance, to be able to disseminate some information through the network. A "piece of information" could take the form of offering free samples of an advertising product, an adoption of an innovation or a new drug in the market, or embrace an opinion about a subject. The interesting question is whom we should target in order to achieve a diffusion that reaches the largest part of the network. There are several measures that study which are the most influential nodes in a graph. In fact, there is no unique way to define the significance of a person. We can use the number of his friends, or how close he is to other people of the society, or many other techniques. The point is which of them targets those nodes that can influence all the others or most of them. And more importantly, what happens when competition comes into play. Which measure is dominant against the others when two antagonistic entities (e.g. competing firms) both try to advertise a product, or spread a political opinion for example.

Another key point is the fact that we need to achieve maximum diffusion with limited resources. For example, when it comes to advertising a product, a company can afford to distribute its product for free to a limited number of people to publicize it, let's say 5-20 items. This means that we need to choose 5-20 people in a network, whichcan achieve high levels of influence to the rest of the society, given the existence of other competitors which try to do the same for their own product. Thus, the budget that each firm is allowed to use, is an important restriction to our research.

There are a lot of questions to be answered in such networks. In this project we concentrate in answering the following one: how a new technology or opinion that is adopted by a small group of agents, can be diffused to the rest of the society and to what extent, given the existence of antagonistic entities in the network which are assumed to choose their own seed sets according to a strategy from a given collection of alternative seed-selection policies. To be more specific, we are interested to examine the diffusion level that each firm can achieve, when more

than one of them are competing to spread their product in the network.

1.1 How to represent a Social Network

Our work is based and applied to Social Networks and Technological Networks. The term "Social Network" refers to an online community of people with a common interest who use a website or other technologies to communicate with each other and share information, resources, etc. Social Network in our work, refers to a community with $N = \{1, 2, \ldots, n\}$ members which interact with each other. The meaning of this interaction is that a new opinion of a member of a society is able to influence the opinions of his neighbors and they in turn are able to influence the opinion of their neighbors etc. The community as a whole, with the interaction patterns that occurs between the members, may well be represented by a graph $G(V, E)$, where V refers to the set of vertices and E refers to the set of edges. A vertex represents a member of the community in which for the rest of the paper we will refer to as agent or node, and an edge represents a weight that lies in the interval $[0, 1]$ and stands for the proportion of influence that takes place between two agents. The interaction patterns that occur between the agents of the community may well be represented by a matrix to which we will refer as the **interaction matrix**. An interaction matrix has the form

$$A = \begin{pmatrix} w_{1,1} & w_{1,2} & \ldots & w_{1,n} \\ w_{2,1} & w_{2,2} & \ldots & w_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n,1} & w_{n,2} & \ldots & w_{n,n} \end{pmatrix}$$

where $w_{i,j} \in [0, 1]$ are referring to the interaction between agents i,j in the following manner. If the network we examine is undirected, then $w_{i,j} = w_{j,i}$, else if the network is directed, then $w_{i,j} \neq w_{j,i}$ and more specifically, we assign equally weights to all the in-edges of every node in the network.

In some cases, as we will describe later, we need the transpose matrix A^T. In undirected graphs, $A = A^T$ because matrix A is symmetric, however this is not the case for directed ones.

1.2 Power Law Networks and Giant Component

In our analysis the networks we are using are power law graphs. This competitive strategy we examine can be applied to any graph category. We specifically choose to test it in power law graphs because they are sparse networks and can represent very accurately a real society. These networks have the following structure: a few nodes have many neighbors, so they can be considered as **supernodes** while the majority of the nodes have very few neighbors. In such networks, given only their structure, we can assume that most influential nodes will be those that have a lot

of friends in the graph. So, the measure that can incorporate this information to select the best nodes, would probably have good results in spreading its innovation against any other.

Most of the networks consist of maximal connected subsets which are called **connected components**. Some of the importance-assessment strategies that we use, can not provide satisfying results in disconnected graphs, so a restriction we are using is that we extract from the selected power law graphs their **giant component**. The term giant component refers to the largest connected component, regarding to the nodes, that exists in the network. After removing the disconnected parts, we apply the importance-assessment strategies to this giant component and then, using the nodes that the measures provided, we let the competition take place also in the giant component. In the case of a graph with disconnected components, some strategies, can be "trapped" in small societies that consists of very small number of nodes, and as a result the diffusion that they can achieve in the end is insignificant. In order for the competition to be fair, and allow to all measures equal opportunities to win the game, we let it take place only on the giant component of each graph.

1.3 Outline of this thesis

The purpose of this work is to find the technique that selects the most influential nodes in a network in order to spread an opinion or a product, given that another firm is trying to spread its own product as well. We can examine the diffusion that each of the measures we used can achieve by its own, however this is not realistic because we can not assume that a single product is the only one that is advertised, or that a specific opinion is the only one that a person hears from his friends. In fact, someone will hear many different opinions from those around him and based on what he hears, he tend to form his own opinion on the subject. Therefore, it is very important information, if we can find a strategy that dominates on any other, in order to choose the most influential people in a society. Even if we can not adjudicate whether a certain tactic is better than the others generally, we may be able to observe under what conditions each measure has the best results.

This research can be helpful to anyone who want to advertise a product, or try to spread an opinion in a social network against other products or opinions respectively. The results can provide the best tactics that one should use to achieve the highest level of influence, initiated by a certain number of people. In the worst case, we can exclude some techniques that do not achieve large diffusion when they compete with others, to any graph. In the end of this thesis, we recommend the best and worst strategies which arise from competitions on real power law graphs.

Chapter 2

Steps of the procedure

The purpose of this work is to understand how a new product can penetrate in a society when, at the same time, its competitors try to influence the same society with their own products. So, the two (or more) antagonistic entities want to start a cascade process in the network, after each one has selected its advertising strategy. The overall process of the competition is divided into three steps. First, we set all nodes in order based on their importance which is achieved by some predetermined **Importance-Assessment Strategies**. Then, a **Seed Selection Policy** is applied and finally we apply an **Information-Propagation Models** to simulate the dispersion of the new products in the entire network.

2.1 Importance-Assessment Strategies

The first thing that each firm has to do, is to decide a partial/total order of the "most significant" nodes in the network. In order to do that, firms are asked to choose an order technique according to some **importance determination policies**. In this work, we use as importance determination policies some of the known **centrality measures** and some variations of them. There are many models that selects important nodes in a network. Our selection criterion is that we chose the most commonly used centrality measures in the literature. In Chapter 3, all these methods, are examined in detail.

Since the predetermined measures have been decided, each firm is free to use any of them, in order to choose its representative nodes. Firms can select even the same strategy, however, as we will show in the results, if they both select the same importance determination policy, neither of them will manage to diffuse its product. In order to understand which of the strategies performs better, we examined the competitions among all the importance determination policies with each other.

In the end of Step 1, we have a decreasing importance order of all nodes, one for each firm.

2.2 Seed Selection Policy

In Step 2, we need to choose the $k \geq 5$ best nodes for every firm, which will represent the sets of their seed nodes. Although the two firms select independently the order of their seeders, there is a chance some of their choices to be common. In this case, a seed selection policy is needed to solve these conflicts, which operates as follows. Each time a common seeder is detected, each firm gets to keep him with probability $p = 0.5$. The firm that loses, selects the next best choice of the order. This tactic is applied until we extract two sets of k nodes (one set for each firm) where we do not have common agents.

By applying this policy, we manage to solve every conflict between the two competitors. However, dividing nodes in this manner, may give advantage to one of the firms, allowing it to select the best agents and leaving the other with not so good options. In order to be fair among the firms, we repeat the seed selection policy ten times and implement the competitions in these ten different seed sets.

2.3 Information-Propagation Models

After the above steps have been completed, we need a model which will measure the spread in the network. For that reason, two different Information-Propagation Models are applied, in order to measure the level of influence of each firm, given the selected seed sets. The diffusion techniques that we used, are analyzed in Chapter 4 in detail. In the end of diffusion process we get the amount of the influenced nodes for both firms and we are able to decide which one wins.

By applying the above steps among all the importance determination policies and the two Information-Propagation Models, and test them in different networks we can conclude to a dominant strategy (if it exists) or to determine an order from the best to the worst seed selection technique.

Chapter 3

Importance-Assessment Strategies

Centrality measures constitute a significant tool in the study of social networks. With their help we can determine whether an agent is important in a society and, in addition, to study the influence he has to the rest of the agents. These measures can be used in many applications such as finding key agents in a society, so that they can help us in the dissemination of an opinion, to stop the spread of a disease, or to use this information in order to derive useful social and economic results. In order to do that, we use several centrality measures which will be further analyzed in the next sections.

There are many centrality measures that have been studied in this area. However, in our work we chose the most notable and widely used measures in the literature. The measures we are studying can be divided into three categories. **1. Exact Centrality Measures**, **2. Approximate Centrality Measures** and **3. Adaptive (Approximate) Centrality Measures**. On the exact centrality measures we will examine *1.a. Betweeness Centrality* [1], *1.b. Closeness Centrality* [7], *1.c. Diffusion Centrality* [8] and *1.d. Degree Centrality* [8]. Exact centrality measures though, take time to compute and are too expensive for large graphs. The second category we study, which is the approximate centrality measures, includes *2.a. Approximate Betweeness Centrality* [4] and *2.b. Approximate Closeness Centrality*. In the third category we examine the following adaptive approximate measures: *3.a. Adaptive (approximate) Betweeness Centrality* [11], *3.b. Adaptive (approximate) Closeness Centrality* and *3.c. BBCL Centrality* [5]. These measures are used as "importance determination" policies, to be considered by the firms. We will provide more details about each category in the following sections.

3.1 Exact Centrality Measures Algorithms and Analysis

The first group of measures in our analysis are the exact centrality measures. Each of the following measures computes the most central agents using the exact calculation of the formula each measure has. This gives us the exact best agents of each

measure which is the most valuable information we can get. The most frequently used centrality measures are betweeness, closeness, degree and eigenvector. The first three were proposed by Freeman [1] and the eigenvector by Bonachich [23], [24].

3.1.1 Betweeness Centrality

One of the most notable and most commonly used centrality measures is Betweenness Centrality (BC) [1]. It indicates the centrality of an agent by counting the number of times this agent lies on a shortest path between other agents. Generally, an agent with large betweeness centrality is expected to have a strong influence on the transfer of knowledge through the social network. However, this doesn't mean that this agent can spread his opinion to all others agents due to the neighborhoods that form the network itself. The general expression for computing BC is given by

$$BC(v) = \sum_{s \neq t \neq v} \frac{\sigma_{st}(v)}{\sigma_{st}}$$

where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(v)$ is the number of shortest paths from s to t that pass through v. With this computation we need $O(n^2)$ which is expensive for large networks.

In our work, calculation of exact Betweeness Centrality is due to the algorithm of Ulrik Brandes [2] which is the most efficient algorithm so far with $O(nm)$ for unweighted graphs and $O(nm + n^2 logn)$ for weighted graphs. Brandes' algorithm operates in two steps called Single Source Shortest Path (SSSP) step and Accummulation step. In the SSSP step, Brandes performs BFS or Dijksta algorithm, for unweighted and weighted graphs respectively, in order to compute the length and number of shortest paths between all pairs of nodes in a graph. Then, in the second step it defines the **dependency score** for each agent $v \in V$ by the expression $\delta_{st}(v) = \frac{\sigma_{st}(v)}{\sigma_{st}}$ which represents the ratio of shortest paths from s to t passing through v divided by the total number of shortest paths from s to t. After that, it accummulates BC by summing all pair-dependencies on vertex v

$$BC(v) = \sum_{s \neq t \neq v} \delta_{st}(v)$$

So far the procedure is the same as the original algorithm proposed by Freeman. Brandes, however, made a crucial observation that helped him significantly reduce time and space in BC computation algorithm. First he defined the dependency score of a vertex as

$$\delta_{s\bullet}(v) = \sum_{t \in V} \delta_{st}(v)$$

which are the one-siding dependencies. Then, he proved that the dependency on any $v \in V$ obeys to the following recurrent expression

$$\delta_{s\bullet}(v) = \sum_{w \in P_s(v)} \left(\frac{\sigma_{sv}}{\sigma_{sw}} (1 + \delta_{s\bullet}(w) \right)$$

where $P_s(v)$ is a set that contains all the predecessors of v. This means that every $w \in P_s(v)$ is an agent within the shortest path from s to v. Finally, BC is computed by the expression

$$BC(v) = \sum_{s \neq v} \delta_{s\bullet}(v)$$

The Brandes' algorithm we used is given below.

3.1.2 Closeness Centrality

The second measure we examine is the Closeness Centrality (CC) which was originally proposed by Bavelas [12]. CC accesses the importance of an agent s by measuring the relative distance of s to all the other nodes in the graph. Practically, $CC[s]$ is the inverse of the average distance between s and any other agent. In order to estimate that importance, we consider the **farness** of agent $s \in V$ to be the summarized distances from s to all $t \in V$. Then the **Closeness Centrality score** is computed by the expression

$$CC(s) = \frac{1}{\sum_{t \in V} d(s,t)} \tag{3.1}$$

A closeness computation algorithm proposed by K. Kaya et al [3] and proceeds in two steps, like the betweeness algorithm. In the first step a Single Source Shortest Path (SSSP) is solved using once more the BFS algorithm or Dijkstra, depending on whether we have unweighted or weighted graph, respectively. This keeps track of all distances from every single node of the graph to any other. Then these distances are summed and Closeness Centrality scores for every $v \in V$ are resulting using the formula (1).

3.1.3 Diffusion Centrality

Another interesting centrality measure is Diffusion Centrality (Diff. C) [8], which is based on information flow through the network. The idea behind Diffusion Centrality is that every agent is able to hear from the other nodes of the graph some rumors. Furthermore, each agent has the ability to restrain the information that he hears from others, in order to count the number of times he heard about a specific agent. This way, at the end of the process each one has a counter for every other agent in the graph.

Algorithm 1 Brandes Algorithm in weighted graphs

Input: directed weighted Graph $G = (V, E)$
Data: queue Q, stack S, dist[v]: distance from source
Pred[v]: list of predecessors on shortest paths from source
$\sigma[v]$: number of shortest paths from source to $v \in V$
$\delta[v]$: dependency of source on $v \in V$
Output: Betweeness Centrality $BC[v]$ for all $v \in V$

1: **for** $s \in V$ **do**
Single Source Shortest Path problem
2: **for** $w \in V$ **do** $Pred[w] \leftarrow emptylist$
3: **end for**
4: **for** $t \in V$ **do** $dist[t] \leftarrow \infty$, $\sigma[t] \leftarrow 0$
5: **end for**
6: $dist[s] \leftarrow 0$, $\sigma[s] \leftarrow 1$, enqueue s $\rightarrow Q$
7: **while** Q not empty **do**
8: dequeue $v \leftarrow Q$, push $v \rightarrow S$
9: **for all** nodes w such that $(v, w) \in E$ **do**
Path Discovery
10: **if** $dist[w] = \infty$ **then**
11: $dist[w] \leftarrow dist[v] + c[v, w]$
12: enqueue $w \rightarrow Q$
13: **end if**
Path Counting
14: **if** $dist[w] = dist[v] + c[v, w]$ **then**
15: $\sigma[w] = \sigma[w] + \sigma[v]$
16: append $v \rightarrow Pred[w]$
17: **end if**
18: **if** $dist[w] > dist[v] + c[v, w]$ **then**
19: $dist[w] = dist[v] + c[v, w]$
20: **end if**
21: **end for**
22: **end while**
Accumulation
23: **for** $v \in V$ **do** $\delta[v] = 0$
24: **end for**
25: **while** S not empty **do**
26: pop $w \leftarrow S$
27: **for** $v \in Pred[w]$ **do** $\delta[v] \leftarrow \delta[v] + \frac{\sigma[v]}{\sigma[w]}(1 + \delta[w])$
28: **end for**
29: **if** $w \neq s$ **then** $BC[w] \leftarrow BC[w] + \delta[w]$
30: **end if**
31: **end while**
32: **end for**

Algorithm 2 Closeness Algorithm in weighted graphs

Input: directed weighted Graph $G = (V, E)$
Data: queue Q, dist[v]: distance from source
$far[v]$: summarized distances of v to every $u \in V$
Output: Closeness Centrality $CC[v]$ for all $v \in V$

1: **for** $s \in V$ **do**
Single Source Shortest Path problem
2: **for** $t \in V$ **do** $dist[t] \leftarrow \infty$
3: **end for**
4: $dist[s] \leftarrow 0$, $far[s] \leftarrow 0$, enqueue s $\rightarrow Q$
5: **while** Q not empty **do**
6: dequeue $v \leftarrow Q$
7: **for all** nodes w such that $(v, w) \in E$ **do**
8: **if** $dist[w] = \infty$ **then**
9: $dist[w] \leftarrow dist[v] + c[v, w]$
10: $far[s] \leftarrow far[s] + dist[w]$
11: enqueue $w \rightarrow Q$
12: **end if**
13: **if** $dist[w] > dist[v] + c[v, w]$ **then**
14: $far[s] \leftarrow far[s] - dist[w]$
15: $dist[w] = dist[v] + c[v, w]$
16: $far[s] \leftarrow far[s] + dist[w]$
17: **end if**
18: **end for**
19: **end while**
20: $CC[s] = \frac{1}{far[s]}$
21: **end for**

The basic idea is that there is a piece of information initiated at an agent i and passed on his neighbors with probability $p \in (0, 1]$. This process is repeated for T periods, where $T \in Z^+$, meaning that every agent that receives the above information passes it to his neighbors, his neighbors to their neighbors etc. with the same probability $p \in (0, 1]$ independently across neighbors and history. Thus, Diff. C measures the propagation extent of a piece of information as a function of the initial agent.

As a result, a ''hearing matrix" is defined as

$$H(\mathrm{g}; p, T) := \sum_{t=1}^{T} (p\mathrm{g})^t$$

where g is the adjacency matrix of the graph. The above relation means that the ij-th element of H represents the expected total number of times that agent j hears a rumor coming from agent i, during T periods. Then, the **Diffusion Centrality** is given by the expression

$$Diff.C(\mathrm{g}; p, T) = H(\mathrm{g}; p; T) * \mathbf{1} = \sum_{t=1}^{T} (p\mathrm{g})^t$$

where $\mathbf{1}$ is the right eigenvector with all entries 1.

It is important to mention here that Diff. C allows agents to hear some pieces of information from the same agent multiple times, therefore the total number of some ij-th elements of matrix H may be greater than the number of agents of the entire network. We are actually double-counting in some cases, however, it does not bother us because in the real world hearing multiple times a specific information operates as a better measure of what we learn. Diffusion Centrality

Algorithm 3 Diffusion Algorithm

Input: directed weighted Graph $G = (V, E)$
Data: probability $p \in (0, 1)$, number of iterations k
Output: Diffusion Centrality $Diff.C[v]$ for all $v \in V$
1: Create table $T = p \cdot adjMatrix(G)$
2: **for** $i = 1, \ldots, k$ iterations **do**
3: Calculate T^i
4: **end for**
5: $T_{final} = \sum_{i=1}^{k} T^i$
6: $Diff.C[v] = \sum_{u \in V} T_{v,u}$
7: return $DC[v]$

is directly related with three widely used centrality measures: Degree Centrality (which we will discuss in detail in the next section), Eigenvector Centrality and Katz-Bonachich Centrality [23]. The critical factor of this relation is the extreme

points of period T. On the one hand when we define $T = 1$ we have the Degree Centrality. Degree Centrality is given by the expression

$$d_g(v) = \sum_{s \in V} g_{vs} \tag{3.2}$$

where g_{vs} indicates that s is an in-neighbor of v.

Diffusion centrality assuming $T = 1$ gives us the expression

$$Diff.C(\mathbf{g}; p, 1) = pd(\mathbf{g}) \tag{3.3}$$

so, as we can see, relation (3) which refers to Diff. C with $T = 1$ is proportional to relation (2) of Degree Centrality.

On the other hand, if we let $T \to \infty$ it depends on probability p whether Diffusion Centrality will approach the Eigenvector Centrality, or the Katz-Bonachich Centrality. The Eigenvector Centrality is defined as $v^{(1)}(g)$: the first eigenvector of g. The Katz-Bonacich Centrality is defined as

$$KB(p, \mathbf{g}) := \left(\sum_{t=1}^{\infty} (p\mathbf{g})^t \right) * \mathbf{1}$$

where $\mathbf{g}$ in both cases represents the adjacency matrix, and suppose that both of Bonacich's parameters are set to p. Jackson et al [8] proved that if $p \geq 1/\lambda_1$, then as $T \to \infty$ Diffusion Centrality approximates Eigenvector Centrality and if $p < 1/\lambda_1$, then Diffusion Centrality is Katz-Bonacich Centrality. For a better background and further reading we refer the reader to [9].

In conclusion, the Diffusion Centrality measure is a method that examines a middle case between the two extremes that are Degree Centrality on one side and Katz-Bonachich (or Eigenvector) Centrality on the other side.

3.1.4 Degree Centrality

We already said a few words about Degree Centrality. It is actually the simplest measure and is easily implemented because all we need to compute, is the degrees of the nodes of the network. Every network is given by either an adjacency matrix **g**, where the ij-th element is equal to 1 if there is a link from i to j, or with an adjacency list of pairs (i,j) that is giving us exactly the links from node i to j. Both representations of the graph, give us as many information as we need to keep count of the degree of a node.

In graph theory, the degree of a node is simply the number of edges that are incident to the node. In our analysis, in order to be fair among the centrality measures, when we refer to the degree of a node, we mean only the number of

the in-edges that the node has. According to this differentiation, degree centrality is given by the expression

$$d_g(v) = \sum_{s \in V_{in}} g_{vs}$$

So the agent with the most in-neighbors among all the nodes in the graph is considered as the most important node of the graph.

Algorithm 4 Degree Algorithm

Input: directed weighted Graph $G = (V, E)$
Data: in degree adjacency matrix $inDegree[v]$, $\forall v \in V$
Output: Degree Centrality $Deg[v]$ for all $v \in V$
1: **for** $i = 1, \ldots, n$ **do**
2: $Deg[i] = inDegree[i]$
3: **end for**
4: Sort $Deg[v], v \in V$ in decreasing order
5: return $Deg[v]$

Degree centrality does not use any other topological properties of the network. According to this measure a node is important if and only if that node has a lot of ''friends". This isn't very realistic in the real world because people don't get decisions about their lives depending on what a famous person says about that. On the other hand, in examples like citation networks this measure might be a good method to choose the important ones.

3.2 Approximate Centrality Measures Algorithms and Analysis

So far we examined some of the most known and used measures to find influential nodes in a network. All the metrics above give us a list with the best nodes in a decreasing order, each one of them selecting the nodes in its own way. Although these metrics are the most popular and widely used methods for the selection of central nodes of a network, in real life they are time consuming and expensive algorithmically.

As a solution to these disadvantages, many approximate measures have been proposed for most of the existing metrics, which provide solutions very close to the actual ones in much less time. We will examine one approach for the exact methods of Betweeness Centrality and Closeness Centrality.

3.2.1 Approximate Betweeness Centrality

The first approximate method we will examine is an approach of Betweeness Centrality. BC has been extensively used in many applications during the last years. Therefore there are plenty of approximations algorithms in the bibliography

that give us different techniques to approximate BC e.g. [4], [20], [43]-[47]. In this thesis, we focus on the approximation proposed by M. Mihail et al. [4].

This approximation algorithm is based on an adaptive sampling technique that manages to reduce significantly the shortest path computations for agents with high centrality scores. It is actually an approximation of Brandes' algorithm [2] which we explained above, thus it also operates in two steps, the SSSP and the Accummulation. Brandes' algorithm needs to compute n shortest paths at the SSSP step to calculate the betweeness score of a node. In [4] it is proved that it is possible to estimate the BC score by computing much less shortest paths.

The adaptive sampling technique of source-nodes was first presented by Lipton and Naughton [48] and the basic idea is that the number of samples depends on the information obtained from each sample. Mihail et al. take advantage of this sampling and they incorporate it to the SSSP step in order to reduce the computations. At first, a random source $s \in V$ is selected and an SSSP step is performed using BFS or Dijkstra's algorithm. In addition, it maintains a running sum S of the dependency scores $\delta_{s\bullet}(v)$ for each node $v \in V$, where $\delta_{s\bullet}(v)$ is the dependency of a node $s \in V$ on a single node $v \in V$ that we mentioned in Section 3.1.1. This process continues until sum S is greater than cn for some constant $c \geq 2$; n represents the total number of nodes in the network. At this point, the computations stop and **Betweeness Centrality Score** is approximated by the expression

$$ABC(v) = \frac{nS}{k}$$

where k is the number of samples examined until sum S achieves the limit cn.

The above procedure is repeated for every node $v \in V$, however when a new source is selected we set the number of samples k to zero because each time we need a different number of samples to decide whether a given node is central. This practically means that the super-nodes of a graph collect large centrality score very quickly, so we don't have to examine all shortest paths to get to conclusion.

As we will present later, this method actually reduces the time needed to approximate BC and [4] proves that for $0 < \epsilon < 0.5$ with probability $p \geq 1 - 2\epsilon$, the above algorithm estimates the sum of dependencies of a node to within a factor of $\frac{1}{\epsilon}$.

3.2.2 Approximate Closeness Centrality

As we already saw with BC, exact Closeness centrality is also extremely expensive to compute in a large graph with thousands or millions of agents. Several methods have been proposed so far to approximate closeness centrality scores such as [4], [9]. In this section, we first propose a customization of the Rand algorithm [7], which in fact is an approximation algorithm that we use to compute **Approximate Closeness scores** and analyze its guarantees.

Algorithm 5 Approximate Betweeness Algorithm in weighted graphs

Input: directed weighted Graph $G = (V, E)$
Data: queue Q, stack S, dist[v]: distance from source
Pred[v]: list of predecessors on shortest paths from source
$\sigma[v]$: number of shortest paths from source to $v \in V$
$\delta[v]$: dependency of source on $v \in V$
Output: Approximate Betweeness Centrality $ApprBC[v]$ for all $v \in V$

1: **for** k samples **do**
2: Perform ***Single Source Shortest Path problem*** as Brandes' algorithm and maintain a counter k_{sample} which increases by 1 each time a new source is selected
//Accumulation
3: **for** $v \in V$ **do** $\delta[v] = 0$
4: **end for**
5: **while** S not empty **do**
6: pop $w \leftarrow S$
7: **for** $v \in Pred[w]$ **do**
8: $\delta[v] \leftarrow \delta[v] + \frac{\sigma[v]}{\sigma[w]}(1 + \delta[w])$
9: $sum[v] \leftarrow sum[v] + \delta[v]$
10: **if** $sum[v] > c \cdot n$ and $v \notin SeedSet$ **then**
11: $v \rightarrow SeedSet$
12: $ApprBC[v] \leftarrow \frac{n \cdot sum[v]}{k_{sample}}$
13: **end if**
14: **end for**
15: **end while**
16: **end for**
17: Sort $ApprBC[v], v \in V$ in decreasing order
18: return $ApprBC[v]$

D. Eppstein and J. Wang proposed [7] a randomized algorithm which can approximate closeness centrality with high probability in time $O(\frac{logn}{\epsilon^2}(nlog(n)+m))$ within an error of $\epsilon\Delta$ where ϵ is a constant in $(0,1)$ and Δ is the diameter of the graph. In that algorithm the inverse closeness centrality scores are computed and it is proved that the expected value of the inverse closeness centrality is equal to the exact inverse closeness centrality.

The RAND algorithm as it is called, is given below:

Algorithm 6 RAND Algorithm

1: Assume that k is the number of iterations needed to acquire the desired error bound
2: In every iteration i, select uniformly at random a source s_i from G and run the SSSP problem
3: The inverse closeness centrality score is computed by the expression

$$\hat{c_u} = \frac{1}{\sum_{i=1}^{k} \frac{nd(s_i,u)}{k(n-1)}}$$

In our experiments we used a variation of the above algorithm, which we describe instantly. As we have seen earlier, the algorithm that calculates Closeness Centrality operates in two steps. At step one we sample uniformly at random M nodes from G and create a set $S = \{s_1, s_2, \ldots, s_M\}$ consisting of these nodes. Then, for every node $s_i \in S$ we solve the Single Source Shortest Path (SSSP) problem by running BFS or Dijkstra algorithm. During this process we keep a weight $w_{s_i}(v) = d(s_i, v)$ for every node $v \in V$ we meet. At the end of the SSSP calculations we compute a total sum W for each node $v \in V$ where $W(v) = \sum_{i=1}^{M} d(s_i, v)$. In the second step of the algorithm we use $W(v)$ to approximate Closeness Centrality scores of each node in the graph. The approximate centrality scores are $\bar{C}(v_i) = \frac{M}{W(v_i)}$.

The number of samples that we take must be $M = \frac{\log n}{\epsilon^2}$ where $\epsilon \in (0,1)$ and n is the total number of nodes, in order to guarantee the quality of the results. In order to prove that $\bar{C}(v)$ is a good approximation of exact $C(v)$ we use Hoeffding's inequality:

Proposition 3.2.1 *If $X_1, \ldots, X_k$ are independent random variables, $a_i \leq X_i \leq b_i$,where $i = 1, 2, \ldots, k$ and $\bar{X} = \frac{1}{k}\sum_{i=1}^{k} X_i$ then*

$$Pr\left[|\bar{X} - E[\bar{X}|\right] \geq \xi] \leq 2e^{-2k^2\xi^2/\sum_{i=1}^{k}(b_i-a_i)^2}$$

In our case we define $X_i(v) = w_{s_i}(v) = d(s_i, v)$ to represent the contribution of a sampled source s_i to the total weight $W(v)$ of a vertex v i.e. the expected

value of X_i is given by $E(X_i) = \frac{1}{n}Far(v)$ where $Far(v) = \sum_{s \in V} d(v,s)$. So if $\bar{W}(v)$ is the mean value of total weights regarding the M samples we took then $\bar{W}(v) = \frac{1}{M}\sum_{i=1}^{M} d(s_i,v) = \frac{1}{M}\sum_{i=1}^{M} X_i(v)$ and it is obvious that

$$E(\bar{W}(v)) = \frac{1}{M}\frac{M}{n}Far(v) = \frac{1}{n}Far(v) \tag{3.4}$$

.

As $n \to \infty$ the above can be written as:

$$E(\bar{W}(v)) = \frac{1}{n}Far(v) = \frac{n-1}{n-1}\frac{1}{n}Far(v) = \frac{1}{n-1}Far(v) = \frac{1}{C(v)} \tag{3.5}$$

Algorithm 7 Approximate Closeness Algorithm

```
Input: Graph G = (V, E), error parameter ε ∈ (0, 1), M = log n / ε^2, set S = {}
for i = 1, ..., M do
    sample at random a set of vertices S = (s_1, s_2, ..., s_M)
    set s_i as source and solve SSSP
    for each neighbor v do w_{s_i}(v) = d(s_i, v)
    end for
end for
for each v ∈ V do
    Return the total weight W(v) = Σ_{i=1}^{M} d(s_i, v)
    and calculate C̃(v_i) = M / W(v_i)
end for
Return C̄(v_i) in decreasing order
```

Proposition 3.2.2 *According to the sampled method described above, we can approximate the inverse Closeness of a vertex within a small error of $\epsilon\Delta$ with high probability. Specifically for any vertex v*

$$Pr\left[\frac{1}{\bar{C}(v)} - \frac{1}{C(v)} \geq \epsilon\Delta\right] \leq 2\frac{1}{n^2}$$

where Δ is the diameter of the graph.

Proof From Hoeffding's inequality we have:

$$Pr\left[|\bar{X} - E[\bar{X}|\right] \geq \xi] \leq 2e^{-2k^2\xi^2/\sum_{i=1}^{k}(b_i - a_i)^2} \tag{3.6}$$

where $a_i = 0$, $b_i = \Delta$, $\xi = \epsilon\Delta$ and $M = \frac{log(n)}{\epsilon^2}$. According to these equation (6) is written as:

$$Pr\left[|\bar{W}(v) - E(\bar{W}(v))| \geq \epsilon\Delta\right] \leq 2e^{-2M\left(\frac{\xi}{b_i}\right)^2} \tag{3.7}$$

$$= 2e^{\left[-2\frac{\log n}{\epsilon^2}\left(\frac{\epsilon\Delta}{\Delta}\right)^2\right]} = 2e^{\left[-2\log n\right]} = 2\frac{1}{n^2}$$

Then, from (5), (7) and setting as our estimator for Closeness Centrality of a vertex the quantity

$$\tilde{C}(v) = \frac{M}{W(v)} = \frac{1}{\bar{W}(v)}$$

we get the result of Proposition 3.2.2. □

3.3 Adaptive Approximate Centrality Measures Algorithms and Analysis

Adaptive techniques are very different from approximate ones. So far, we reached the conclusion that with approximate methods we can achieve, with fewer calculations, results equally good as exact measures. The adaptivity on the other hand, accomplishes a different outcome. The basic idea is that we choose every time the best agent, depending on the measure we are using, and then in order to compute the next best node, we remove the first one and recompute the scores in the new network. This helps us because influential nodes in a graph are often close to one another and very central. Thus, it is not efficient to choose more than one **supernodes** as we call them, which are close to each other and have similar neighbors, because the budget of ''advertisers" nodes is limited. In some cases, in order to be able to diffuse information in the entire network we need as seeds distant nodes.

There are plenty of adaptive techniques in the bibliography. We preferred to test out two adaptive methods based on Betweeness and Closeness centralities. In addition, we will examine a method that is not referred to any of the known measures, proposed by Borgs et al. [5]. Eventually we will be able to decide, depending on the problem which we have to solve, which method is more efficient to use.

3.3.1 Adaptive Approximate Betweeness Centrality

So far we examined two algorithms about Betweeness Centrality. In this section, we will discuss an adaptive variant of the approximate Brandes' algorithm. It is interesting to study such a method, because sometimes we want to get the k best nodes which belong to different regions. For this purpose, we used Y. Yoshida's algorithm [11] to compute **Adaptive Betweeness Scores**.

This adaptive algorithm performs in two steps. The first objective is to build a hypergraph as in Borgs et al. [5]. A **hypergraph** H, as we mentioned before, consists of a set of nodes $V = \{v_1, v_2, \ldots, v_n\}$ and a set of hyperedges $E = \{e_1, e_2, \ldots, e_n\}$. Each hyperedge e_i represents a set of nodes S_i, created by computing a directed acyclic graph, started from a source s to a target node t, using BFS or Dijkstra algorithm. Next, due to Brandes' algorithm, weights are assigned to every node of each hyperedge that depicts their importance. The weights are determined by the formula

$$e = \left\{ (v, \frac{\sigma_{st}(v)}{\sigma_{st}}) | v \in P_{st} - \{s,t\} \right\}$$

where σ_{st} is the total number of shortest paths between s and t and $\sigma_{st}(v)$ is the number of shortest paths between s and t that pass through v.

One difference of this algorithm compared with the exact betweeness algorithm, is that shortest paths are computed for pairs of nodes (s,t). This means that the SSSP step of Brandes' algorithm is executed normally until BFS or Dijkstra algorithm reaches the target node t. At that point the SSSP procedure is interrupted and a hyperedge is created which contains all nodes that were discovered thus far. After M hyperedges are created, the importance of each node is estimated by summing the weights that he has in each of the hyperedges that participates. The node which has the highest score is considered as the most influential.

The adaptive step occurs after all the computations above. Once we selected the best node, we replace each hyperedge e_i in which the best node participates, with a new hyperedge e_i'. This new hyperedge is computed from the beginning using the algorithm we described earlier in this section. The selected best node and his edges in the original graph $G = (V,E)$ are removed before new computations.

Algorithm 8 Build Betweeness Hypergraph Algorithm

Input: Graph $G = (V,E)$, an integer $M \geq 1$
1: Initialize $H = (V, \emptyset)$
2: **for** $i = 1, \ldots, M$ **do**
3: Pick a set of vertices (s,t) at random
4: Make a weighted hyperedge $e = \{(v, \frac{\sigma_{st}(v)}{\sigma_{st}}) | v \in P_{st} - \{s,t\}\}$, add it to H
5: return H
6: **end for**

The runtime of the algorithm is almost linear. It needs $O(hM)$ to build the hypergraph. By using a priority queue, finding the maximum weights at most nM times it gives us $O(hM\log(n))$. So, finally the total run time is $O((n+m)M + hM + hM\log(n))$.

Algorithm 9 Top-k ABC Algorithm

Input: Graph $G = (V, E)$, an integer $k \geq 1$
1: $M \leftarrow O(\log(n)/\epsilon^2)$
2: $H \leftarrow BuildBetweenessHypergraph(G, M)$
3: **for** $i = 1, \ldots, k$ **do**
4: $v_i \leftarrow argmax_v\{w_H(v)\}$
5: **for** each hyperedge e incident to v_i **do**
6: Replace it with a new weighted hyperedge
7: $\{(v, \frac{\sigma_{st}(v|v_1,\ldots,v_i)}{\sigma_{st}})|v \in P_{st} - \{s,t\}\}$
8: **end for**
9: **end for**
10: return the set $\{v_1, \ldots, v_k\}$

3.3.2 Adaptive Approximate Closeness Centrality

In this section, we define an adaptive method for computing Closeness Centrality scores in a network. The technique we are using is based to Yoshida's top-k algorithm we described in the previous section. Moreover, we used Borgs' et al. [5] technique to create a hypergraph on which we apply the measure.

The first part of the algorithm that we describe, is similar to algorithm 7 of Section 3.2.2. From lines 1-7, where the SSSP problems are solved, the procedure remains the same except from an addition of a line where the hyperedges will be generated. Each hyperedge consists of a set of nodes S_i that is created by running BFS or Dijkstra algorithm, starting with source node s_i. Applying this to algorithm 7 will give us one hyperedge for each SSSP we solve, so in the end of the process M hyperedges will have been created. Now, we need to assign weights to the hyperedges in order to be able to decide the importance of the nodes. Thus, for each $v \in P_{e_{s_i}}$, where $P_{e_{s_i}}$ is the set of nodes that constitute the shortest paths from source s_i to all other nodes, we set a weight given by $w_{s_i} = d(s_i, v)$. Therefore, until this point we have constructed a hypergraph $H = (V, E_H)$ where $V = \{1, \ldots, n\}$ is the set of nodes of the initial graph G, and $E_H = \{e_{s_1}, \ldots, e_{s_M}\}$ is the set of hyperedges. Note that hyperedges are actually pairs of (v, w_v) for each $v \in P_{e_{s_i}}$.

The second part consists of the adaptive step. Once we have the hypergraph H, we get as the most influential node the one that achieves the highest score by the expression

$$W(v) = \sum_{i=1}^{M} w_{s_i} = \sum_{i=1}^{M} d(s_i, v)$$

Afterwards, we create a new set A which will provide the final seed set we need. Every time a best node is selected we add it to that set A. To continue with the next seed node, we remove the one we took from graph G, and we recompute all the hyperedges of H that it participates. So in every repetition, if that node is a

part of k hyperedges, we need to recalculate the distances for these k hyperedges starting from source nodes $s_i, i = 1, \ldots, k$, excluding the selected seed node and his edges in real graph G.

So, for a set $A \subset V$ the **Adaptive Closeness Centrality (ACC) score** of a node v can be defined as

$$ACC(v) = ACC(v|A) = \frac{|V - A|}{F(v|A)} = \frac{|V - A|}{\sum_{s \in V \setminus A} d(v, s|A)}, v \in V \setminus A \tag{3.8}$$

At this point we can define the total Adaptive Closeness Centrality of a set of vertices $A \subset V$, combining relations (1),(8) to be

$$ACC(A) = \frac{|V - A|}{F(A)} = \frac{|V - A|}{\sum_{i=1, s \in V \setminus A}^{n} d(v_i, s|A)}, v \in A, i = \{1, \ldots, k\} \tag{3.9}$$

An observation here is that relation (9) gives us a completely different outcome from **Group Closeness Centrality (GCC)** [14], [15]. GCC calculates the score of a set of vertices too, however it doesn't take into consideration the already chosen seeds. Below we present the algorithm we described.

Algorithm 10 Adaptive Closeness Centrality

Input Graph $G = (V, E)$, error parameter $\epsilon \in (0, 1)$,$M = \frac{\log n}{\epsilon^2}$, set $S = \{\}$, set $A = \{\}$
for $i = 1, \ldots, M$ **do**
    sample at random a set of vertices $S = (s_1, s_2, \ldots, s_M)$
    set s_i as source and solve SSSP
    make a new weighted hyperedge $e_{s_i} = \{v, w_{s_i}\}$ where
    for each neighbor v **do** $w_{s_i}(v) = d(s_i, v)$
    end for
end for
Return the total weight $W(v) = \sum_{i=1}^{M} d(s_i, v)$ from all hyperedges
and calculate $\tilde{C}(v) = \frac{M}{W(v)}$
for $i = 1, \ldots, k$ **do** pick v_i with $max\tilde{C}(v)$ and put it in $A = \{\}$
    for each e_{s_i} incident to v_i **do** run SSSP from s_i and
        replace e_{s_i} with a new $e_{s_i} = \{v, w_{s_i}(v) = d(s_i, v|A)\}$
    end for
    for each $v \in (V - A)$ **do** calculate $W(v|A) = \sum_{i=1}^{M} d(s_i, v|A)$
        return $\tilde{C}(v) = \frac{M}{W(v|A)} = \frac{M}{\sum_{i=1}^{M} d(s_i, v|A)}$ in decresing order
    end for
end for
Return set $A = \{v_1, v_2, \ldots, v_k\}$

In a few words, in each iteration algorithm 10 extracts node v_i with the highest

ACC score. Then, v_i is excluded from the graph, along with his edges. The ACC scores are recomputed in the new graph $G - \{v_i\}$ and a new best node appears. This process continues for k steps, so in the end we have the set $A = \{v_1, \ldots, v_k\}$, which gives us the k best agents in an adaptive manner.

As we proved in Section 3.2.2, Proposition 3.2.2 applies to the adaptive measure too.

Proposition 3.3.1 *If $\tilde{A}$ is the result of our algorithm, then algorithm 2 is with high probability a good approximation of the inverse Closeness Centrality of the exact greedy algorithm within a small error of $\epsilon\Delta$ and in fact*

$$Pr\left[\frac{1}{\tilde{C}(\tilde{A})} - \frac{1}{C(\tilde{A})} \geq \epsilon\Delta\right] \leq 2\frac{1}{n}$$

3.3.3 BBCL Centrality

The BBCL centrality measure is based on finding those nodes in a graph that can begin a diffusion process, in order to achieve the maximum resulting cascade. The most important outcome though, is that BBCL runs in nearly optimal time which makes this measure highly competitive. To be fair with the other measures we examine, we will compare the results of BBCL metric with the results of the approximate and adaptive methods. Exact measures are, by their very nature, slow and apparently will lose the speed issue. Even though exact metrics can provide accurate results, networks these days can be volatile so we need metrics that can compute quick solutions rather than slow and precise ones.

The algorithm proposed in [5] relies on the **Independent Cascade model (IC)** formalized by Kempe et al [21]. In this model, a network is depicted as a weighted directed graph G, in which diffusion occurs via a random process starting with a set of nodes S. Then, every node at the moment he is influenced has a chance to spread the influence himself to his own neighbors. The weighted edges $e = (u, v)$ represent the probability that the influence passes from u to v. IC model shows that influence can spread stochastically through a network.

On the basis of the IC model, BBCL assumes the following "polling" process, as it is called. Select a uniformly random node v from graph G. Then, estimate a set of nodes A that could have influenced node v. Repeat that estimation k times, and if a specific node u appears continuously as influencer then he is probably a good node to be selected as seed node. If we define as **I(S)** the number of nodes that are eventually affected by the seed set S, then [5] proves that the probability of a node u to appear in a set of influencers is proportional to E[I(u)]. In order to find the seed set S the polling process is implementing to the reverse graph G', which is the original graph G with reverse edge directions.

Algorithm 11 BBCL Algorithm

Require: Precision parameter $\epsilon \in (0, 1)$, directed edge weighted graph G

$R = 144(m+n)\epsilon^{-3}log(n)$
$H = BuildHypergraph(R)$
return BuildSeedSet(H,k)

BuildHypergraph(R):

Initialize $H = (V, \emptyset)$
repeat
    Choose node u from G uniformly at random
    Simulate influence spread, starting from u, in G^T.
    Let Z be the set of nodes discovered
    Add Z to the edge set of H
until R steps have been taken in total by the simulation process
return H

BuildSeedSet(H,k):

for $i = 1, \ldots, k$ **do**
    $v_i = argmax_v\{deg_H(v)\}$
    Remove v_i and all incident edges from H
end for
return $\{v_1, \ldots, v_k\}$

The basic algorithm works in two steps. First, using the above procedure of random sampling, a hypergraph H is created as follows. First it selects a random node $s \in V$ as source and then it uses Depth First Search algorithm (DFS) to create a spanning tree started at s. Next, a coin is flipped on every edge of the spanning tree and each edge is kept with probability 0.5. At this point, we have a disconnected spanning forest. In order to build the hypergraph mentioned above, the DFS algorithm runs again on the spanning tree this time and a new smaller tree is returned which is connected. So, we get a set $A = \{v_1, v_2, \ldots, v_k\}$ where $v_1, v_2, \ldots, v_k$ are the nodes of the spanning tree that resulted. This set A is defined as a hyperedge e of the hypergraph H. If we repeat these steps for k times, the result will be a hypergraph $H = (V, E')$ where V is the set of nodes of the original graph G and E' is the set that consists of the hyperedges above.

Afterwards, we need to decide which node is the most important one. To do that, for every node $v \in V$, is computed the number of the hyperedges that he participates. At this point, we have a list with n counters for n nodes that represent the level of influence of each node in the graph. Thus, if we arrange them in decreasing order based on that counter, the node with the largest counter value will be the most important one. These counters are defined as **degrees on the hypergraph** structure. In order to take the k most important nodes, an adaptive technique is used. Every time that a node is selected as the best one,

he is removed from every hyperedge that he participates. Then, the degrees of hypergraph are re-computed for every $v \in V$ and again the node with the highest degree is selected.

The BBCL algorithm is limited by parameter $R = 144(m+n)\epsilon^{-3}log(n)$ and this bound is maintained so that the algorithm runs in almost linear time. The main result is that for any $\epsilon > 0$ the algorithm returns a set of nodes S which approximates by a factor of $(1 - \frac{1}{e} - \epsilon)$ the exact set S' that a greedy method would produce, with probability $\frac{3}{5}$ and in $O((m+n)\epsilon^{-3}log(n))$ time.

Chapter 4

Information-Propagation Models

Up to this point, we studied different ways to decide which nodes of a graph can be considered as important. Given the seed sets for the two firms, we need to apply a method that will simulate the amount of other agents in the network that the important ones can influence. To do that, we will use two different methods of diffusion: **1) De Groot Model** and **2) General Threshold Model**.

In particular, we are interested not just about the diffusion each measure from Section 3 can achieve in a network by its own, but for the extent to which the diffusion is achieved for each firm, in a competitive scenario where agents have to decide which of the two opinions to adopt. Each firm, is represented by a set of nodes that spread an opinion, or advertise its product, or even the expansion of a disease. In each case, an opinion or a product, if it is the only one that spreads in the network, may have the chance to reach every other agent starting from the selected seed nodes. In reality though, there will not be a single opinion in a society or a unique product for advertising. So, here we enter the concept of competition. We assume that we have two different opinions to spread in the network. In the rest of the thesis, we will use those terms to examine the diffusion level between two competitors.

In every competition between two firms, each one has to choose one centrality measure in order to settle on its seed nodes. In order for the selection to be fair, we let the firms choose a centrality measure from the same category. This means that firms can compete using both *1) Exact Centrality measures*, *2) Approximate Centrality measures* or *3) Adaptive Approximate Centrality measures*. Furthermore, we make the assumption that they are not allowed to choose the same measure. According to these restrictions, we examine all combinations of exact, approximate and adaptive approximate competitions.

At first, we let each firm to select 10 seed agents depending on the centrality measure she chose. In many cases though, centrality measures we studied, can export same nodes. So, at this point, we have to make a choice about whether one of the firms will keep this common seed agent or neither of the firms will. Initially we thought that it would be fair that if they happen to choose a same

seed node, they both lose him, so as neither of them can take advantage of him. Practically, the result was that both companies lost a significant amount of seed nodes, and more important, they lost their best ones. So, in the end, the diffusion was extremely limited which is understandable considering that selected seeds were less influential nodes.

As a middle ground, we considered the following procedure. Each firm has chosen 10 agents as seeds. For every node that both select, we flip a coin so that if "heads" comes up firm 1 keeps the seed else if "tails" comes up firm 2 keeps him. Therefore, in case that we have conflict, each firm keeps the seed agent with probability 0.5. With this process, the influential nodes are not being wasted. However, this creates a new problem to deal with. If we apply the above once, then we give advantage to the firm that ultimately kept the common good seed agent. To avoid this outcome, we apply the probability procedure ten times for each competition and hence, there are ten different pairs of seed sets created. As a result, the most influential nodes are split, some to firm 1 and others to firm 2. Eventually, the total diffusion of each firm is computed by the average value of these ten competitions.

In the next two sections, the two diffusion methods we used are described in detail.

4.1 De Groot Model

The first diffusion model we study is a continuous metric. The basic idea of De Groot model [9] is that the opinion of an agent in a subject depends on the opinions of his neighbors on that subject.

Consider a network $G = (V, E)$ where $V = \{1, 2, \ldots, n\}$ is the set of nodes and $E = \{e_1, e_2, \ldots, e_m\}$ is the set of edges. Each agent $v \in V$ has an initial opinion on a subject. These opinions can be represented as a n-dimensional vector of probabilities $p(0) = (p_1(0), p_2(0), \ldots, p_n(0))$, where $p_i \in (0, 1)$. Then, a possibly weighted, non-negative $n \times n$ matrix T is introduced, which represents the interaction patterns. More specifically, T is a **row stochastic matrix**, that is, if we add all the values in each row, this sum is equal to 1. Every element T_{ij} of that matrix belongs to the interval $(0, 1)$ and shows the weight that agent i gives on the opinion of agent j, in order to form his own opinion for the next time period. So, if we start with an initial vector of opinions $p(0)$, in the next time period the corresponding vector will be $p(1) = T \cdot p(0)$. Applying this rule, $p(2)$ is given by $p(2) = T \cdot p(1) = T \cdot T \cdot p(0)$ and continuing in this matter for t times, in the end we will have the expression

$$p(t) = T \cdot p(t-1) = T^t \cdot p(0) \tag{4.1}$$

This expression shows that knowing only the initial vector of opinions and the matrix T, we can compute all the other vectors of probabilities until we reach convergence

of the model. To understand how De Groot model operates, we describe it with the following example.

Suppose we have a small society composed of three individuals whose interaction patterns are represented by the following matrix

$$T = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 1/2 & 0 \\ 0 & 1/4 & 3/4 \end{pmatrix}$$

and the initial opinions are represented by vector

$$p(0) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$$

According to what we mentioned above, the vector of beliefs at time 1 will be

$$p(1) = Tp(0) = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/2 & 1/2 & 0 \\ 0 & 1/4 & 3/4 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1/3 \\ 1/2 \\ 0 \end{pmatrix}$$

If we apply this t times, where $t \to \infty$ we have

$$p(t) = Tp(t-1) = T^t p(0) \to \begin{pmatrix} 3/11 \\ 3/11 \\ 3/11 \end{pmatrix}$$

In order to arrive at the final vector of opinions, we expect $T^t \cdot p(0)$ to converge in a single vector after t times. This means that, from this point on, the final opinions have been formed and they won't change anymore by the neighbor's opinions. However, this isn't always the case. There are some examples that $T^t \cdot p(0)$ never achieves convergent. For instance if we have the following updating matrix

$$T = \begin{pmatrix} 0 & 1/2 & 1/2 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

then there will be no convergence of beliefs to a single value since the matrix will oscillate

$$T^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix}, T^3 = \begin{pmatrix} 1/2 & 1/2 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix}, T^4 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/2 & 1/2 \\ 0 & 1/2 & 1/2 \end{pmatrix}$$

Golub and Jackson [25], [9] proved that a stochastic matrix T is convergent if and only if the underlying graph is strongly connected and aperiodic. In large graphs where T is a $n \times n$ matrix not all agents converge to the same value but

convergence still exists. In such graphs usually there is one or several closed and strongly connected groups of agents and each of those groups will reach its own consensus. In other words there is a point where each of those groups will have the same specific value of opinions. DeMarzo et al. [26] shows that the remaining agents who are somehow connected with directed paths to the strongly connected groups will end up with some weighted average of the limit beliefs that have been achieved by the strongly connected groups.

4.1.1 De Groot Model Implementation

In this section we describe how we used De Groot model in order to achieve maximum diffusion in a network. First of all, the networks we study, are referring in directed (or undirected) and weighted graphs. In any directed graph, a node v has a number of out-edges and a number of in-edges. In simple words, node v follows some of the other nodes of the graph and indicated by his out-edges, and also nodes which follow v are indicated by v's in-edges. So, we can separate the original graph in the out-degree and the in-degree representations. These two, represent the same original graph; the only difference is that each of them consider different neighborhoods for every $v \in V$.

Suppose we are given a graph G. This means that we have an adjacency list

$$u_1 \to v_1$$

$$u_2 \to v_2$$

$$\dots$$

$$u_m \to v_m$$

which shows us the interactions between all nodes. Suppose we have the following adjacency list of a graph

$$1 \to 4$$

$$2 \to 1$$

$$2 \to 3$$

$$3 \to 4$$

$$4 \to 1$$

$$4 \to 2$$

Then, there are two ways to demonstrate this graph which are

$$I_{out} = \begin{pmatrix} 0 & 0 & 0 & w_{1,4} \\ w_{2,1} & 0 & w_{2,3} & 0 \\ 0 & 0 & 0 & w_{3,4} \\ w_{4,1} & w_{4,2} & 0 & 0 \end{pmatrix}$$

and I_{in} is practically the transpose I_{out}^T

$$I_{in} = \begin{pmatrix} 0 & w_{2,1} & 0 & w_{4,1} \\ 0 & 0 & 0 & w_{4,2} \\ 0 & w_{2,3} & 0 & 0 \\ w_{1,4} & 0 & w_{3,4} & 0 \end{pmatrix}$$

where $w_{i,j}$ are the weights of the edges in G.

After examining both cases in several graphs, we noticed that De Groot model operates way better with I_{in} as an input. To understand that, consider that each node wants to influence his neighbors so that his opinion can spread. As we described before, the in-degree of each node v represents the followers of v. So, if v tries to propagate his opinion to others, those that will hear him are his followees and not the followers. In his turn, v as well, will hear the nodes which he follows and that procedure continues for all the graph.

This perspective works for De Groot model, however, some centrality measures from Section 3 cannot follow. For example, if Diffusion Centrality plays against other measures, it fails to spread the initial opinions using the seed set which provides. So, to be fair both to the centrality measures and the diffusion models, we assume that all methods will apply to the in-degree graph. For some measures though, like Betweeness or Closeness centralities, there is little difference to the final seed sets in whatever graph (in or out degree) we use. This happens due to their own technique on choosing influential agents.

As for the weights of the graphs, we assign to all in-edges of each node $v \in V$, a weight equal to $1/inDegree(v)$. This occurs for two reasons. Firstly, in the De Groot model we need a matrix which is row stochastic in order to operate properly. And secondly, since our aim is to compare the two diffusion models, De Groot and Threshold, in order to decide which one achieves a larger spread, we must use the same matrix. As we will describe in the next section, the Threshold model does not take into account the weights of the edges. This means, that every node considers all of his neighbors equals, so each one can influence him with $1/inDegree$. Thus, if we assign these weights instead of random ones, we examine the spread in both diffusion models in identical graphs.

Note that, these weights refer to the percentage of influence that neighbors of v have on v. So, consider a small neighborhood with three agents and $(1/2, 1/4, 1/4)$ to be the first line of the adjacency matrix. This means that node 1 takes into consideration 50% of his own opinion, 25% of node's 2 and 25% of node's 3, to form his new opinion.

At this point, we need to make some additional customizations so as to create a game between the firms. At first, we consider two additional nodes in our graph, where node $n+1$ represents the first firm and node $n+2$ the second one. These nodes are considered as "stubborn" agents in the way that they can influence some of the other agents in the graph, but all the other agents cannot influence them.

Thus, their initial opinion has value 1 and remains constant throughout the process. Practically, we add two rows and two columns in our matrix, so the new adjacency matrix T has dimensions $(n+2) \times (n+2)$.

$$T = \begin{pmatrix} w_{1,1} & w_{1,2} & \dots & w_{1,n} & 0 & 0 \\ w_{2,1} & w_{2,2} & \dots & w_{2,n} & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 1 & 0 \\ 0 & 0 & \dots & 0 & 0 & 1 \end{pmatrix}$$

As we can see above, at the beginning nodes $n+1$ and $n+2$ are isolated and they only care about their own opinion. The purpose of these additional nodes is simple. Suppose that the firms have selected their seed sets, $\{u_1, u_2, \dots, u_{10}\}$ and $\{v_1, v_2, \dots, v_{10}\}$ for firm 1 and firm 2 respectively. Then, we connect nodes $\{u_1, u_2, \dots, u_{10}\}$ to stubborn agent $n+1$, and nodes $\{v_1, v_2, \dots, v_{10}\}$ to stubborn agent $n+2$. Since both sets have been selected as seed sets, we want their opinion to stay unchangeable from future interactions. To achieve that, we define a constant $c = 1000$ which depicts the level of influence that stubborn agents have upon the seed sets. After this observation, the last two columns of matrix T becomes

$$\begin{array}{c} \begin{matrix} \text{n+1} & \text{n+2} \end{matrix} \\ \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 1000 & \vdots \\ \vdots & \vdots \\ 1000 & 0 \\ 0 & 0 \\ \vdots & \vdots \\ 0 & 1000 \\ \vdots & \vdots \\ \vdots & 1000 \\ \vdots & \vdots \\ 1 & 0 \\ 0 & 1 \end{pmatrix} \end{array} \begin{matrix} 1 \\ 2 \\ \vdots \\ u_1 \\ \vdots \\ u_{10} \\ \vdots \\ \vdots \\ v_1 \\ \vdots \\ v_{10} \\ \text{n} \\ \text{n+1} \\ \text{n+2} \end{matrix}$$

In the next step, we normalize the rows where we added constant c in order to have a row stochastic matrix to apply De Groot model.

So far, we have created a new interaction matrix W, which incorporates two firms as "stubborn" agents and the interaction patterns that occurs between all

nodes of the society. Now, we need to define an initial vector of opinions $p(0) = [p_1, p_2, \ldots, p_{n+2}]$ where $p_i \in [0, 1]$. Opinion vector $p(0)$ represents the initial opinions of all agents. Before the process begins, we assume that all agents, except for the firms and the seed sets nodes, are indifferent. Thus, if we set firm $n+1$ to be depicted by value 0, and firm $n+2$ by value 1, then all the indifferent nodes at the beginning have the value 0.5.

Theoretically, a node v remains neutral until the end of the process, if the $v-th$ co-ordinate is equal to 0.5. In computer science, it is almost impossible to remain immutable, regardless of what is happening to others around you. So, in our experiments, we suggest that there is an interval $[0.4, 0.6]$ that indicates the indifference of each agent. After all, as we move away from either of the original interval limits, which are 0 and 1, we tend to become neutral.

The key point here, is the fact that both interaction matrix W and initial opinion vector $p(0)$, includes two different firms as stubborn agents, and each one choose a centrality measure which represents the agents that will advertise their products. It is known from the literature, that stubborn agents tend to make other agents to compromise with their opinion or even adopt it [9], [10]. This doesn't mean that they can achieve to spread the opinion through all the network. In order to transmit a piece of information to everyone, the two stubborn agents are insufficient and one of the most important factors is the graph structure.

It is obvious that each firm uses the same seed set in both W and $p(0)$. Setting the set of centrality measures as the set of strategies, and adopting a game theoretical approach, it is easy to see that there is a **zero-sum game** between the firms. A zero-sum game means that one firm's gain is equivalent to another's loss. In our analysis, if the first firm manages to influence most of the graph, then the other one will take only the left over nodes. The purpose of this game is to define which strategy (i.e. which centrality measure) is the dominant one in order to prevail in the society's market.

Combining everything that was mentioned so far, and by using formula (10), we are able to decide which strategy is better than another. Due to the De Groot model, matrix W should reach convergence to stop the procedure. In our experiments we used a large enough number of iterations, to approach as much as possible that convergence. The actual number we used is $t = 2^{10}$. Due to the number of iterations that are needed, and considering the size of the networks we examine, the time required for this algorithm is quite large. The basic time consuming operations are the powers of the interaction matrix W. Generally, each multiplication needs $O(n^3)$, so in order to compute W^t for a large t, we need $t \cdot n^3$ time at worst case. We succeeded to restrain this limit by reducing the multiplications using the following technique.

Starting with matrix W, we need to compute first matrices $W^2, W^3, \ldots, W^t$ and then take the result of the multiplication $W^t \cdot p(0)$. That means that we have to compute $t-1$ multiplications. Instead of doing it this way, we exploit the fact that matrix W^t results by continuously multiply W to itself. Thus, at first

iteration we compute W^2, then continue with $W^2 \cdot W^2 = W^4$, $W^4 \cdot W^4 = W^8$, ..., $W^{t/2} \cdot W^{t/2} = W^t$. Using that trick, we can calculate higher powers t, in less time. Note that, the higher power we get, the closer we are to the convergence of the matrix, according to De Groot model.

4.1.2 Rules and restrictions

Each measure has its own special features and may depend on a completely different concept than the others. For example Diffusion and Closeness Centralities in a power law graph will probably give high scores in vertices that are located in small components rather in large ones. That is an unrealistic and misleading result and when those strategies compete with betweenness or BBCL method it is absolutely certain that they will lose. Thus our major restriction in order to have a fair competition between firms and the strategies they use, is to limit our research in the giant component of such graphs, so that all measures are implemented and then compete in a (strongly) connected network where there exists a (directed) path from every node to every other node.

The second rule we applied, was mentioned in Chapter 4 and involves the management of seed nodes that both firms selected. Naturally, some important agents will appear in two or maybe more seed sets from different strategies. In that case, we let only one firm keep the common seed node and the second firm replaces him with her next best choice. In order to decide which of them will keep him, we conduct a random experiment and with probability $p = 1/2$ each, they keep or extract the node from the seed set respectively. We iterate this process until we get two different seed sets composed of ten nodes, one for each strategy. This **excluding rule** is applied to avoid any kind of conflict between agents on whose side to take.

Since the outcome from this rule is a probabilistic one, it could turn to be unfair if we use it just once and the results from the De Groot process would be unrealistic. For example, we may have the case where all the common agents of the competitors seed sets happens to go to one strategy and the other strategy is obliged to compete with far less important agents, and this will result in losing the game. So, for each competing pair of strategies we apply the excluding rule ten times, so as to get ten different competing pairs of seed sets. Finally, we run De Groot model for each pair of seed sets separately. Thus, for each competing pair of strategies we get ten different results of the De Groot process. Generally, the strategy that provokes the larger diffusion throughout the society on the average is the winner. We use this method for all competitions where we have observed common agents between seed sets and in Chapter 4, we demonstrate those results using box plots.

4.2 Threshold Model

The second diffusion model we examine is a discrete diffusion model called General Threshold model. There are various types of threshold models like **linear threshold model** [18] and **general threshold model** [21]. In our experiments we used general threshold model which is a generalization of Granovetter's linear threshold model [27]. The basic idea is that every agent can be influenced by his neighbors equally, and he adopts an opinion if a fraction p of his neighbors have adopted it. Each individual can be either active or inactive. Every agent's tendency to become active increases monotonically as more of his neighbors become active over time. This is a very simple idea and easy to implement.

In the General Threshold model, each node v has a **monotone activation function** f_v and a **threshold value** θ_v which is chosen uniformly and independently at random from the interval $(0, 1]$. Threshold value θ_v represents the fraction of v's neighbors that must become active in order for v to become active as well. Every node at the beginning of the process is considered as inactive (or as we named them in the De Groot model, neutral) except for the nodes that are selected as seeds. A node v which is inactive at time t, becomes active at time $t+1$, if $f_v(S) \geq \theta_v$, where S is the set of v's neighbors that are active at time t. This means that at some point, if a large subset of v's neighbors have the same opinion about a subject, then v will adopt their opinion too. Once a node becomes active at some time $t+1$, he stays active for the rest of the procedure and he tries to influence his neighbors at the next time periods. So, from the moment he becomes active, he cannot switch to inactive again.

The nodes that have become active are considered as contagious nodes due to the fact that they can affect other agents in the graph. Each time a node tries to affect one of his neighbors, he influences them to adopt the current opinion that he has. The value of threshold θ_v can be different for every node $v \in V$, or it can be the same number for all agents. The process terminates when all agents reach a stable phase, which means that for two consecutive iterations the inactive ones stay inactive.

There are various works on the subject of spreading an innovation or an opinion and the effects of ''word of mouth'' strategies. The first investigations on the subject were concerned about the adoption of medical and agricultural innovations [28, 29, 30]. Later, there were studies interested in diffusion processes of ''word of mouth'' and ''viral marketing'' effects in the success of new products [31] - [37] and suddenly penetrate to game theoretic setting [38] - [42]. So, as we can see, there are many applications for these models and we refer the reader to the above studies for more information.

4.2.1 General Threshold Model Implementation

The whole concept of our research is to investigate a game theoretic approach between two firms which compete to maximize their influence in a society. In order

to do that we use the same logic as Section 4.1 for the De Groot model, but this time we combine the Threshold Model with Game Theory concepts.

Once again, the networks we are testing are weighted and directed (or undirected) graphs. The Threshold model is applied to the in-edges graph in order to be fair among the diffusion models. One difference between the Threshold and the De Groot model is that the first one is not using the edge's weights to accomplish the spread through the network. This means that for every node $v \in V$, each of its in-edges neighbors is considered as equal, i.e. each one of them can influence node v with $1/InDegree(v)$. In Section 4.1, we defined weights of every graph G to be given exactly from that expression.

To start the process we need two inputs. The first one, as we mentioned above, is a directed graph G and the in-edges weights for each agent v given by $1/InDegree(v)$. The second input, are the seed sets selected from the firms (derived from the centrality measures in Chapter 3). Each firm has chosen 10 agents to spread her product against the other one. So, we have two different centrality measures competing for a bigger level of influence in a society. In other words, we have a 2-player zero-sum game between two firms, who introduce their innovation into two different seed sets of agents and try to achieve the best possible propagation level in order to win the game.

At this point, we set as inactive all agents in the graph, except for the seed nodes. The initial opinions are represented by a vector $p(0) = [p_1, p_2, \ldots, p_n]$, where $p_i \in [0, 1]$. The values 0, 1 represent the two firms respectively. All the other agents are neutral, so their value is set to 0.5. Therefore, initial vector $p(0)$ will have the form

$$p(0) = [0.5, \ldots, 0.5, \mathbf{0, \ldots, 0}, 0.5, \ldots, 0.5, \mathbf{1, \ldots, 1}, 0.5, \ldots, 0.5]$$

where the values 0 and 1 are not necessarily assigned to consecutive nodes. The Threshold model is a discrete model in contrast with the De Groot model. This means that, whenever a node decides to adopt an opinion, his value on the opinion vector will be either 0 or 1. If he decides to stay neutral his value will remain 0.5. Vector p can take only these three values, one for each state.

In the De Groot model we saw that there are various levels of influence, and more specifically we defined an entire interval of indifferent nodes. Moreover, it was important how close to 0 and 1 were the values of vector $p(0)$ because that showed the level of adoption of the innovation. Here, the values of p are absolute, nodes don't tend to adopt an opinion but they actually adopt it.

The procedure continues as follows. In each iteration, every node keeps two counters, one for firm 1 and one for firm 2. Let's assume that a node $v \in V$ counts his neighbors at iteration i and the result is that he has k neighbors which have adopted firm 1, and l which have adopted firm 2. Then, node v has to decide if he will adopt one of the innovations with respect to the counters that he kept in the previous step. In order to follow either of them, he examines if the

neighbors which are not neutral, exceeds his threshold value θ_v. If

$$\frac{k+l}{neighbors(v)} \geq \theta_v$$

then he must choose which one of them he will adopt. To do that, we produce a random number $r \in [0, k+l]$ and if $r \in [0, k]$ then v follows firm 1 else if $r \in [k+1, l]$ he follows firm 2. This experiment practically says that v follows:

- firm 1, with probability $p_1 = \frac{k}{k+l}$
- firm 2, with probability $p_2 = \frac{l}{k+l}$

Note that this is not the only way to use threshold value θ_v. For example we could define that neighbors from one firm only should exceed θ_v in order to influence node v. We use the total number of decided nodes in our experiments, because the networks we examine are sparse and we would have limited diffusion otherwise.

With the propagation rule we entered, we created the following problem. Due to probabilities p_1 and p_2 we can get very different results for a competition between the two firms, using the same pair of seed sets. So, to eliminate the extreme cases, we repeat the Threshold model 1000 times for each competition separately and we consider as final outcome the average value of these 1000 independent experiments. Later, when we will present the experiments, we will consider the amount of dispersion found in every case, to see how often we encounter extreme situations.

The process of the Threshold model terminates if for some consecutive iterations k, $k+1$ all agents are stable. This means that the inactive nodes stay inactive, and the active ones maintain the opinion which they adopted since iteration k, in iteration $k+1$.

4.2.2 Rules and restrictions

As in the DeGroot model we have two major restrictions in the implementation. Firstly, we have to limit our research in the giant component of the graph so as we examine both diffusion models with the same inputs. As we explained in Section 4.1.2, this limitation ensures that there will be a fair competition between firms and helps to come up with realistic conclusions from the results. Secondly, in directed graphs we let our centrality measures to be computed in the in-degree graph. We applied Threshold model to both in-degree and out-degree graphs, and the results weren't so different. However, we have this restriction in order to be able to compare results from both diffusion models with the same criteria.

Moreover, the excluding rule that we use in De Groot model remains here as well, to ensure that we will have a fair procedure. For each competing pair of strategies, we apply this rule ten times, so as to get ten different competing pairs of seed sets. Combining that rule with the 1000 independent experiments for each

pair of seed set that we described before, we get a total of 10.000 iterations of Threshold model for each competition. The fact that the Threshold model runs far more quickly than the De Groot, enables us to do experiments like that, and furthermore, examine what happens in much larger graphs.

Chapter 5

Experiments

All experiments were conducted at a personal computer that has an AMD 8350 processor with 8 cores running at 4.3 GHz. It also has 8 GB RAM. All algorithms have been implemented in java version jdk1.8.25 and for compilation we have been using eclipse luna 4.4.1.

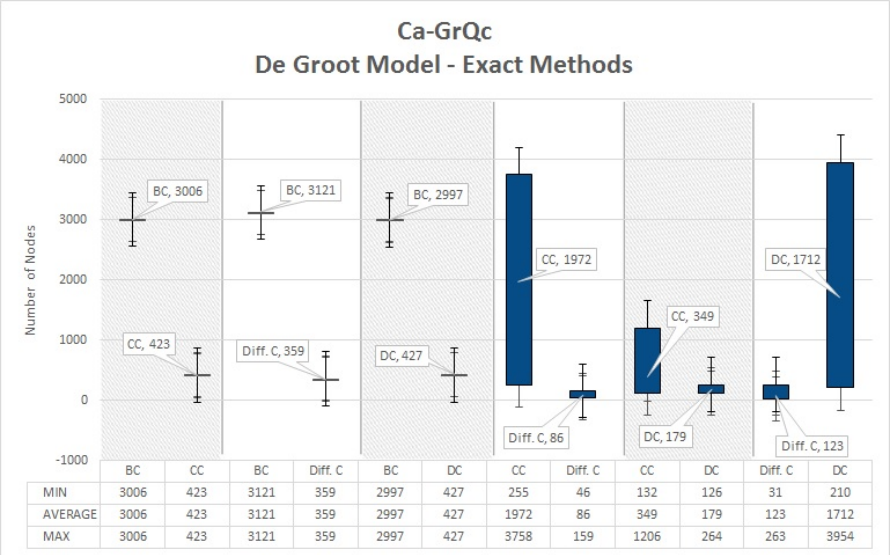
5.1 General Relativity and Quantum Cosmology

General Relativity and Quantum Cosmology (GR-QC) graph stands for a collaboration network that is taken from the e-print arXiv and covers scientific collaborations between authors of papers submitted to General Relativity and Quantum Cosmology category. The data covers papers in the period from January 1993 to April 2003 (124 months). It has 5242 nodes and 28980 edges in total, but because of our restriction we investigate only the largest component which has 4158 nodes and 26850 edges. This network is undirected and its diameter is 17.

5.1.1 De Groot experiments

Below, we examine the competitions among the exact centrality measures, then the approximate measures and adaptive ones, using as diffusion procedure the De Groot model. The charts show us if we can draw a conclusion about which of the measures achieves greater diffusion against the others.

In Figure 5.1 we can see the competitions between exact measures. It is obvious that Betweeness Centrality is the dominant strategy here. BC wins all the other competitors and achieves to influence most of the graph while its opponents manage to spread their product only to a small number of agents. In this experiment BC does not have common agents in its seed set with the other ones, so neither of them loses good nodes. Closeness centrality comes second winning both Diffusion and Degree measures. Note here that CC has common agents with Diffusion and Degree measures, so we examined ten different sets of initial seed sets and as a result we define the average diffusion of these ten experiments. Regarding the CC



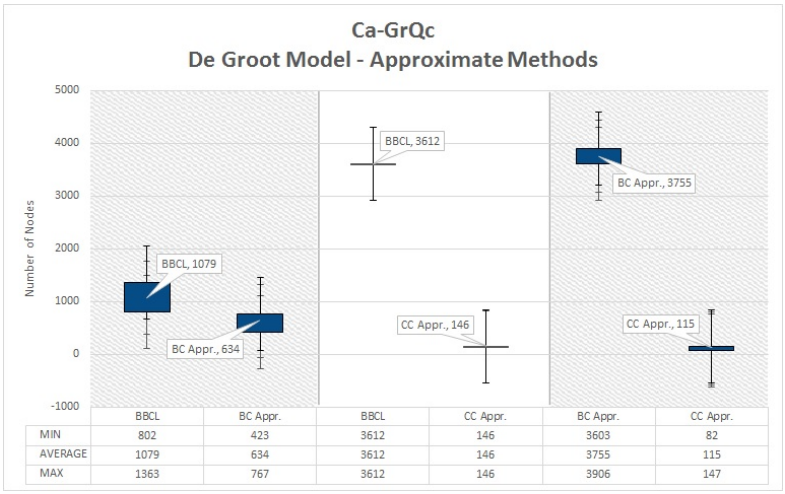
	BC	CC	BC	Diff. C	BC	DC	CC	Diff. C	CC	DC	Diff. C	DC
MIN	3006	423	3121	359	2997	427	255	46	132	126	31	210
AVERAGE	3006	423	3121	359	2997	427	1972	86	349	179	123	1712
MAX	3006	423	3121	359	2997	427	3758	159	1206	264	263	3954

Figure 5.1: De Groot Model - Competitions of the Exact Methods

VS Diffusion results, we can see that CC can influence at minimum 255 and at maximum 3758 agents. This difference on the amount of spread results from the different seed sets that we are using. In the best case CC manages to influence almost all the network but this happens only using a specific seed set where it keeps its best nodes. CC wins also DC, however in that case the diffusion for both measures is very low. CC can achieve to influence the 1/3 of the network at best but on average it does not have so good results. Finally, when Diffusion and Degree compete, DC stands out and wins on average. Nevertheless, in some experiments, Diffusion keeps some of the good nodes and manages to block DC in spreading. Diffusion centrality is the worst strategy here, since it loses from all the others.

Next, Figure 5.2, presents the results from the approximate measures competitions. In approximating methods BBCL seems to be the dominant one since it wins the other two strategies. BBCL has many common seeds with BC Approximate measure, so the results are derived from ten different seed sets. Despite that, the diffusion has a small range from 802 to 1363 agents. This shows us that whichever seed set we use the influence can only reach at best 1/3 of the network. Regarding to BBCL VS CC Approximate competition, BBCL also wins and in fact it wins easily and with great difference from his opponent. These two measures do not have common agents, so the experiment takes place using the initial nodes that they selected. The next best choice is the approximation of BC centrality since it manages to influence most of the graph when it competes with approximate CC, as in the exact case.

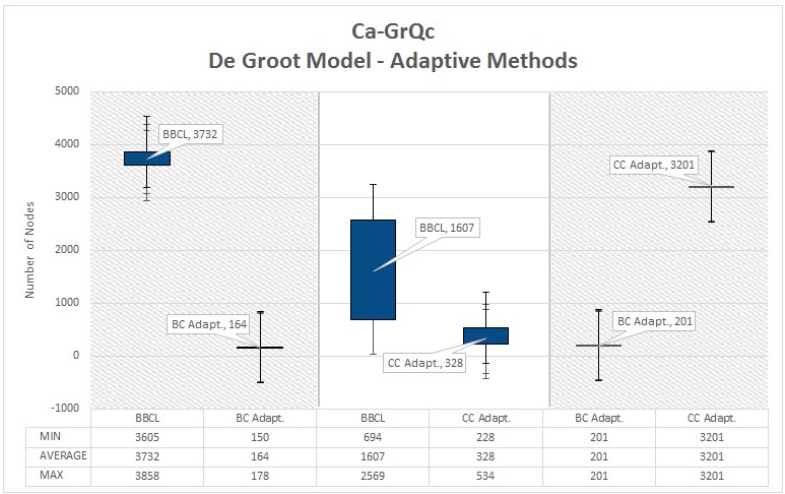
Finally, the competitions among adaptive measures follows in Figure 5.3. BBCL is the best strategy again, winning both adaptive BC and CC by far. When it



	BBCL	BC Appr.	BBCL	CC Appr.	BC Appr.	CC Appr.
MIN	802	423	3612	146	3603	82
AVERAGE	1079	634	3612	146	3755	115
MAX	1363	767	3612	146	3906	147

Figure 5.2: De Groot Model - Competitions of the Approximate Methods

competes with adaptive BC it achieves the higher level of diffusion, taking 4/5 of the agents. BBCL has common agents with adaptive BC and with adaptive CC, so there are more than one seed sets that are tested. Nevertheless, it manages to win regardless the agents that it loses. The next best strategy is adaptive CC. Adaptive CC and adaptive BC have no common agents, so the results are due to the original seed set they created. Adaptive CC wins and influences many agents, showcasing the crucial effect of adaptivity in the computation of the two measures.

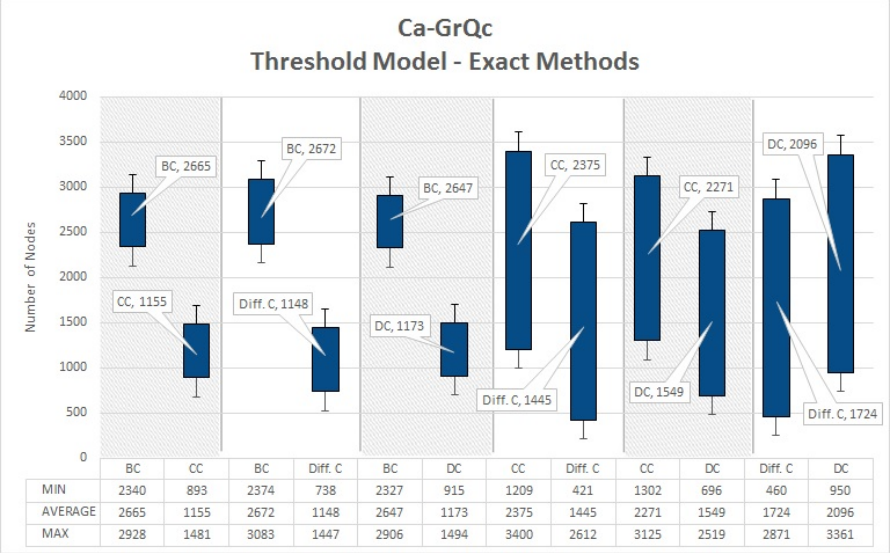


	BBCL	BC Adapt.	BBCL	CC Adapt.	BC Adapt.	CC Adapt.
MIN	3605	150	694	228	201	3201
AVERAGE	3732	164	1607	328	201	3201
MAX	3858	178	2569	534	201	3201

Figure 5.3: De Groot Model - Competitions of the Adaptive Methods

5.1.2 Threshold experiments

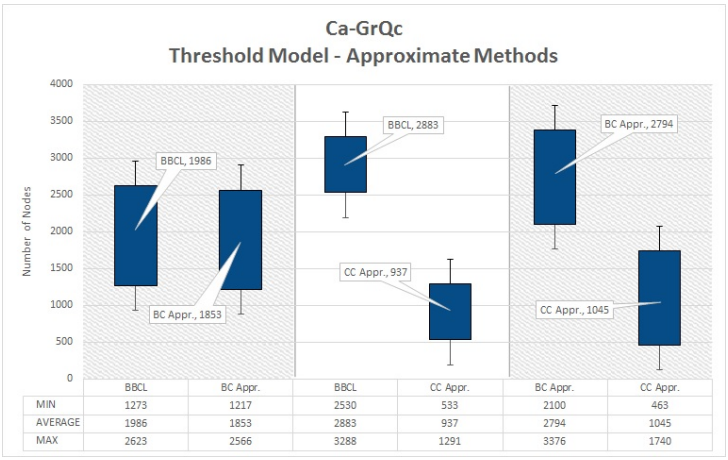
Using the same pairs of seed sets with the above competitions, we have the following results from Threshold Model. In Figure 4 we represent the chart with the competitions of the exact measures. Betweeness Centrality is the best strategy also here, using Threshold model. BC wins all the other measures and the diffusion level is high. The second one is CC which wins the other two measures in average. We notice that the box plots of CC and Diffusion overlap in relation with the number of nodes that they affect. The same observation applies also on the CC VS Degree and Diffusion VS Degree competitions. On these competitions, as we mentioned above, there are ten different pairs of seed sets that are examined, so it is expected the fact that we will have a larger range of results. Nevertheless, on average CC achieves to dominate against both Diffusion and Degree measures and can be considered as the second best technique. Continuing, DC comes third and Diffusion loses from all the others, so it is the worst strategy.



	BC	CC	BC	Diff. C	BC	DC	CC	Diff. C	CC	DC	Diff. C	DC
MIN	2340	893	2374	738	2327	915	1209	421	1302	696	460	950
AVERAGE	2665	1155	2672	1148	2647	1173	2375	1445	2271	1549	1724	2096
MAX	2928	1481	3083	1447	2906	1494	3400	2612	3125	2519	2871	3361

Figure 5.4: Threshold Model - Competitions of the Exact Methods

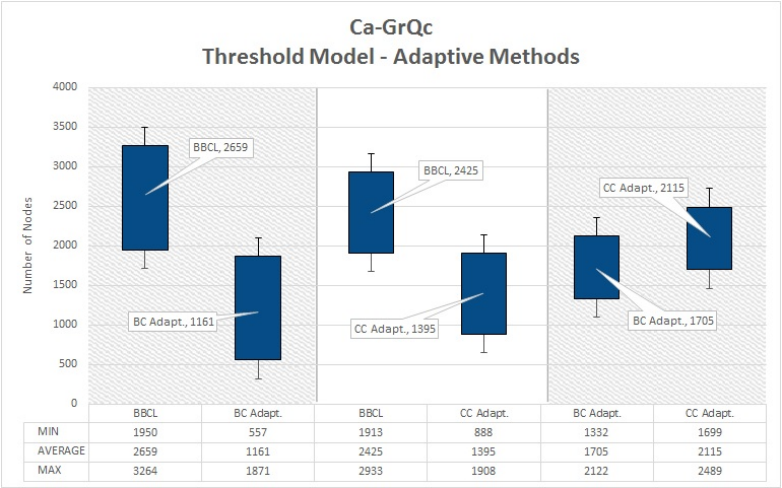
Observe that, whichever of the diffusion techniques we are using, the decreasing order of the best measures is as follows. BC manages to win in both cases all the other metrics, CC is next, followed by DC and last is the Diffusion centrality. From that observation we can understand that the order of centrality measures is not influenced by the diffusion methods, but only on the level of the diffusion that is achieved. This is significant because, if we have to choose among the two diffusion techniques, the Threshold model is way more faster and can be applied to very large networks. This is not the case for the De Groot model though, which has extensive calculations in every step and this makes it an inefficient diffusion simulation model for large networks.



	BBCL	BC Appr.	BBCL	CC Appr.	BC Appr.	CC Appr.
MIN	1273	1217	2530	533	2100	463
AVERAGE	1986	1853	2883	937	2794	1045
MAX	2623	2566	3288	1291	3376	1740

Figure 5.5: Threshold Approximate Methods

For the approximate measures we have the following results. BBCL is the best choice since it wins the other two approximate measures and approximate BC comes second. BBCL and approximate BC are very close and at most of the experiments they influence half of the network each, but on average we can see that BBCL dominates with some small difference. On the other two competitions the results are more clear about who is the winner. This is almost the same picture as we have in the De Groot model with the only difference that in the Threshold Model we get much larger diffusion levels from all strategies, whether they win or not.



	BBCL	BC Adapt.	BBCL	CC Adapt.	BC Adapt.	CC Adapt.
MIN	1950	557	1913	888	1332	1699
AVERAGE	2659	1161	2425	1395	1705	2115
MAX	3264	1871	2933	1908	2122	2489

Figure 5.6: Threshold Adaptive Methods

In adaptive measures we get exactly the same picture regarding the BBCL technique against the other two adaptive measures. BBCL is the dominant one again, even when we use adaptive algorithms for betweeness and closeness metrics.

This is a very powerful result. BBCL can dominate among all other measures and is not affected by the diffusion technique. On the other hand, we observe that CC manages to win adaptive BC, despite the fact that on the exact and the approximate competitions between them, BC always won so far. The adaptivity helps CC to choose better seed nodes this time and he gets to win in the end.

5.1.3 Quantiles

In this section we provide a chart for every competition among the exact, approximate and adaptive techniques, applied in the Threshold Model, in order to examine the distribution of the results of the 10.000 experiments (in each competition). We need to consider if the average values of the firms are resulting from some extreme cases in order to understand how the different seed sets affect the outcome.

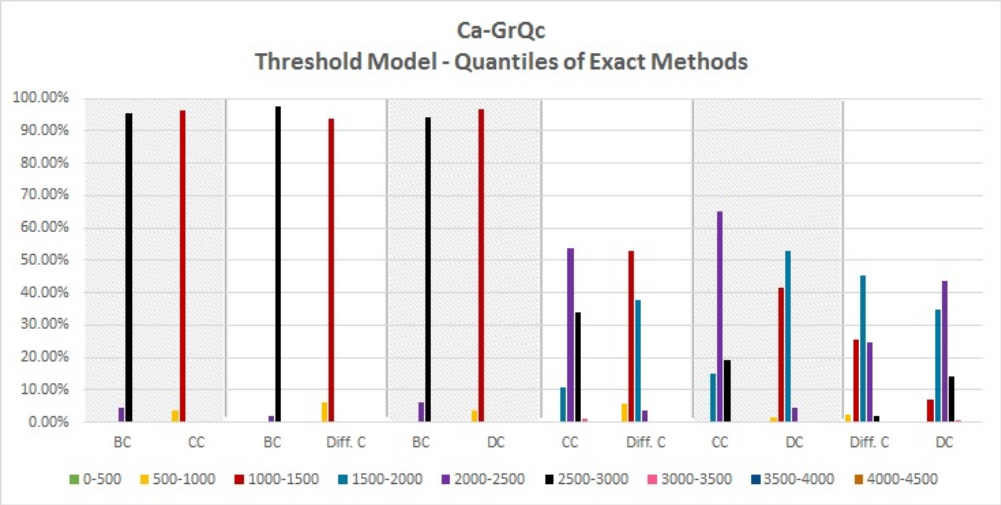


Figure 5.7: Quantiles of the Exact Methods

For the exact methods we can see in Figure 5.7 that in the first three competitions of the BC with the other measures, BC influences 2500-3000 nodes among 95% of the 10.000 experiments. So, as we can see, there aren't extreme events, regardless the seed agents. The same result is observed also for the other competitions among CC, Diffusion and DC Centralities.

Next we provide the quantile graphs for the approximate and adaptive measures.

In both cases, we can see that in every case, $95-99\%$ of the experiments influences a number of nodes that approach the average values of the two firms.

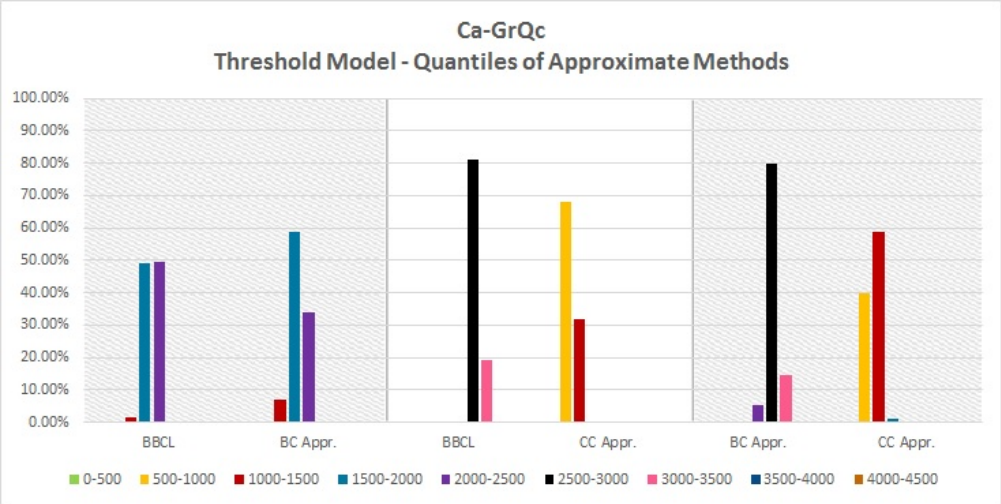


Figure 5.8: Quantiles of the Approximate Methods

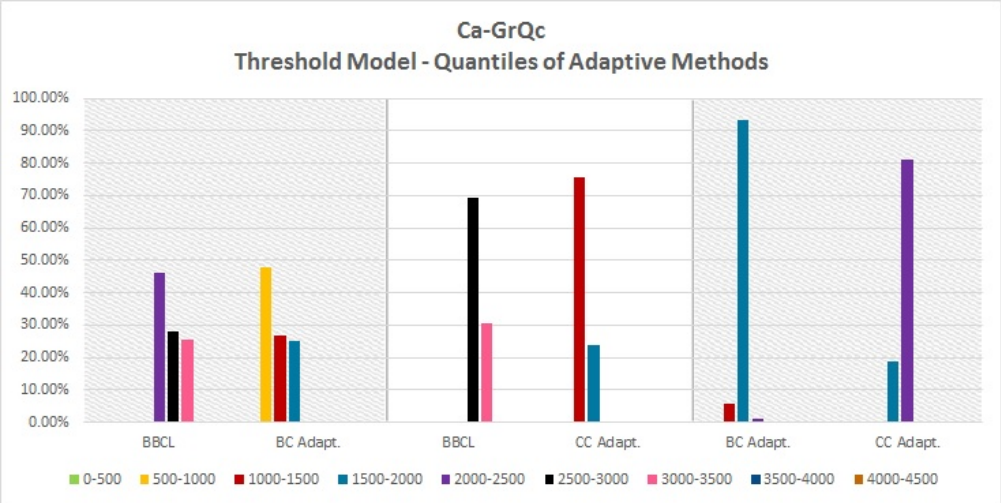


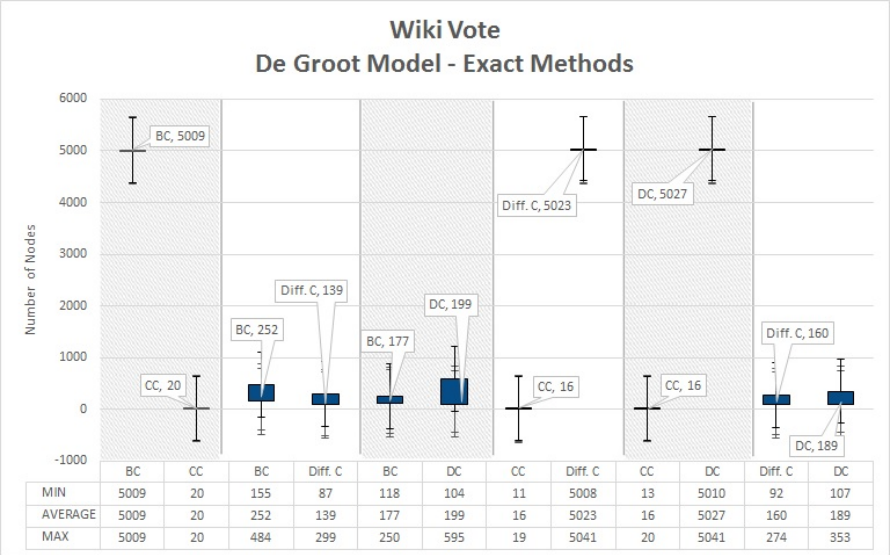
Figure 5.9: Quantiles of the Adaptive Methods

5.2 Wiki Vote

Wikipedia is a free encyclopedia written collaboratively by volunteers around the world. A small part of Wikipedia contributors are administrators, who are users with access to additional technical features that aid in maintenance. In order for a user to become an administrator a Request for adminship (RfA) is issued and the Wikipedia community via a public discussion or a vote decides who to promote to adminship. So Wiki-Vote graph represents all the Wikipedia voting data from the inception of Wikipedia till January 2008. Nodes in the network represent Wikipedia users and a directed edge from node i to node j represents that user

i voted on user j. There are in total 7115 nodes and 103689 edges, but the largest component that we are interested in has 6243 nodes and 68430 edges. The diameter of Wiki Vote graph is 7.

5.2.1 De Groot experiments

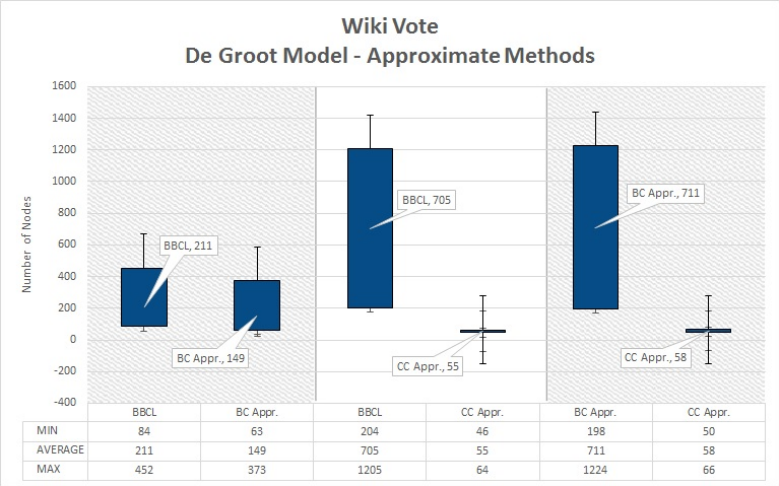


	BC	CC	BC	Diff. C	BC	DC	CC	Diff. C	CC	DC	Diff. C	DC
MIN	5009	20	155	87	118	104	11	5008	13	5010	92	107
AVERAGE	5009	20	252	139	177	199	16	5023	16	5027	160	189
MAX	5009	20	484	299	250	595	19	5041	20	5041	274	353

Figure 5.10: De Groot Model - Competitions of the Exact Methods

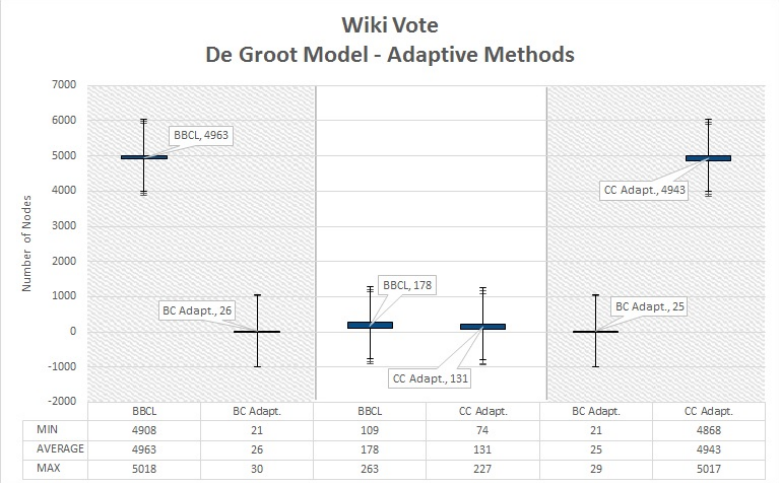
In the exact competitions category Degree centrality is the best strategy as it dominates all other strategies. Betweenness centrality comes second and Diffusion third. Closeness centrality is doing really badly in this graph as it provokes the worst possible spread. In fact the initial seed set influences only few more agents from the whole giant component. As a result strategies that compete with Closeness are able to influence almost the rest of the graph. There is a high level of indifferent nodes between the other three strategies, which means either that they select almost equally important nodes and their influence basically is blocking each other, or that the De Groot diffusion model can not spread the influence very far. We will examine the results when the Threshold model is applied and we will get to a conclusion about these odd cases. Another explanation for these high levels of indifference could be the topology of the specific network. Wiki Vote is a denser graph and since we have edge weights proportional to the degree of a node it is difficult for an agent to influence a neighbor with high degree.

Closeness centrality is the worst strategy in approximate measures too. Exactly as in the previous network, BBCL comes first and wins the other two approximate metrics. However, the diffusion levels in all cases are very low as they were on the exact competitions too.



	BBCL	BC Appr.	BBCL	CC Appr.	BC Appr.	CC Appr.
MIN	84	63	204	46	198	50
AVERAGE	211	149	705	55	711	58
MAX	452	373	1205	64	1224	66

Figure 5.11: De Groot Model - Competitions of the Approximate Methods

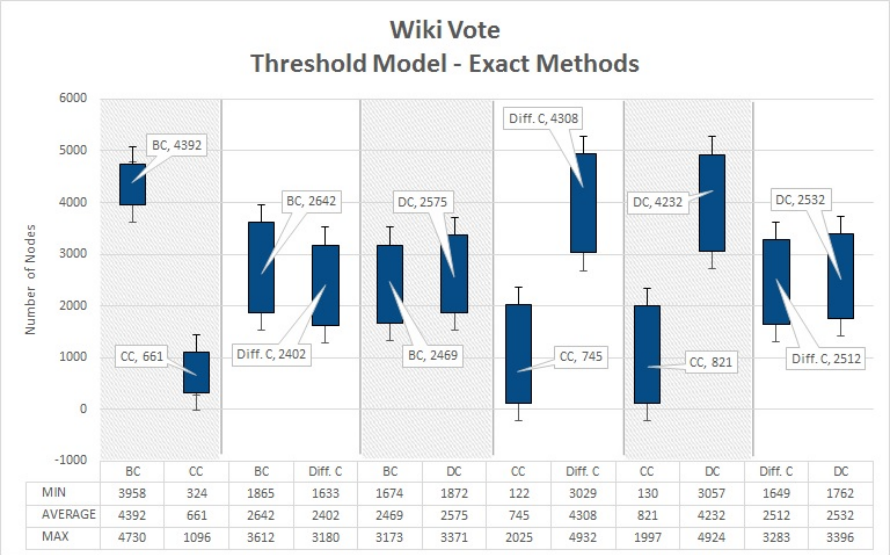


	BBCL	BC Adapt.	BBCL	CC Adapt.	BC Adapt.	CC Adapt.
MIN	4908	21	109	74	21	4868
AVERAGE	4963	26	178	131	25	4943
MAX	5018	30	263	227	29	5017

Figure 5.12: De Groot Model - Competitions of the Adaptive Methods

For the adaptive measures we have the same dominant strategy which is BBCL. Observe that adaptive betweeness can not achieve much diffusion, so the measures that are its opponents manage to take almost all the other graph. This shows us that the selected seed nodes from adaptive betweeness are most probably remote agents and definitely not the supernodes who have a lot of neighbors and are central. Although BBCL and adaptive CC win adaptive betweeness, when these two play as opponents they block each other and neither of them achieves to influence many other nodes.

5.2.2 Threshold experiments



	BC	CC	BC	Diff. C	BC	DC	CC	Diff. C	CC	DC	Diff. C	DC
MIN	3958	324	1865	1633	1674	1872	122	3029	130	3057	1649	1762
AVERAGE	4392	661	2642	2402	2469	2575	745	4308	821	4232	2512	2532
MAX	4730	1096	3612	3180	3173	3371	2025	4932	1997	4924	3283	3396

Figure 5.13: Threshold Model - Competitions of the Exact Methods

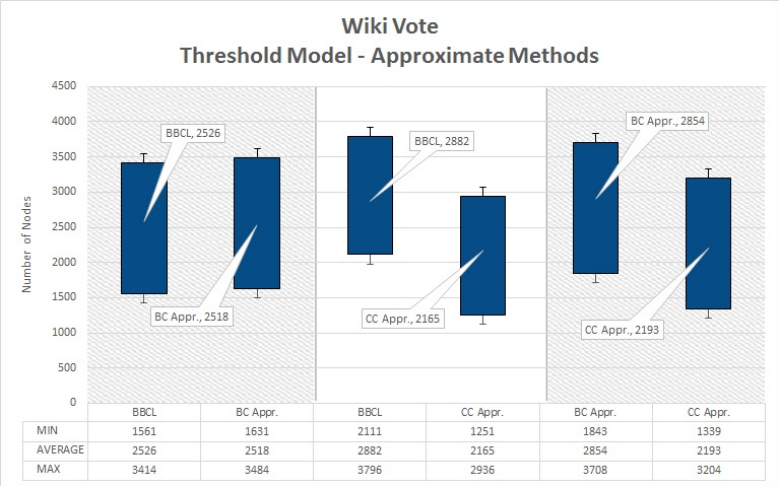
In the Threshold model the diffusion levels for both exact betweeness and degree centrality are equally good. They both win when their opponent is either closeness or diffusion. When they play as opponents the graph is separated almost equal between them. Diffusion comes next, which defeats closeness but this is not so important in this network because closeness centrality does not pick influential nodes at all as we can see. CC loses from any other measure in both De Groot and Threshold models. The highest score is reached by DC, so if we want to distinguish one of the measures, that would be DC.

Regarding the competitions on the approximate measures, BBCL is the dominant strategy followed by approximate BC with very minor difference. Furthermore, diffusion levels are higher here than in the exact methods.

Finally, in adaptive metrics the results are not good. Whenever adaptive betweeness is playing with the other measures as opponents, the level of diffusion stays very low. As we mentioned in the De Groot experiments, this network is quite dense. In the Threshold model we use as threshold value $\theta_v = 0.2$ which means that if 1/5 of v's neighbors decide to buy a product for example, then v is called to decide based on his neighbors' selection, which of the products he will by himself. That limit is high when a node has many neighbors and considering that the procedure starts with 20 nodes (10 for each firm).

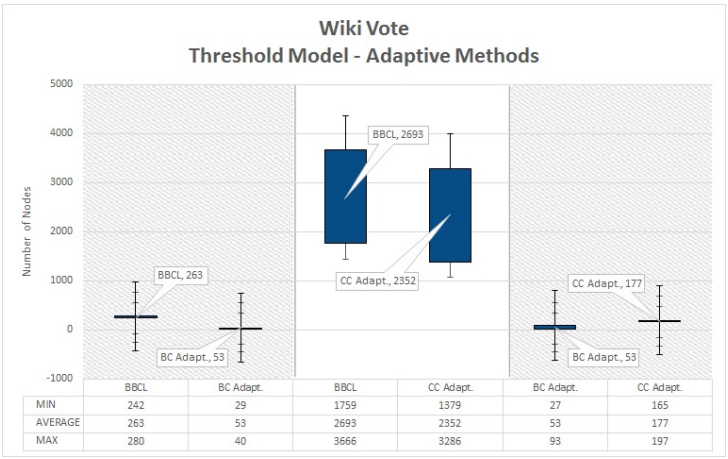
If we set $\theta_v = 0.1$ then we have the results in Figure 5.16.

So, as we can see, threshold value θ_v has an important role on the level of the diffusion we want to achieve. When we have larger and denser networks, we must either get a smaller threshold value or get more nodes as seeders in order to



	BBCL	BC Appr.	BBCL	CC Appr.	BC Appr.	CC Appr.
MIN	1561	1631	2111	1251	1843	1339
AVERAGE	2526	2518	2882	2165	2854	2193
MAX	3414	3484	3796	2936	3708	3204

Figure 5.14: Threshold Model - Competitions of the Approximate Methods



	BBCL	BC Adapt.	BBCL	CC Adapt.	BC Adapt.	CC Adapt.
MIN	242	29	1759	1379	27	165
AVERAGE	263	53	2693	2352	53	177
MAX	280	40	3666	3286	93	197

Figure 5.15: Threshold Model - Competitions of the Adaptive Methods

achieve wide spread.

5.2.3 Quantiles

In Wiki Vote graph, quantiles are more clear than before. Both strategies, regardless which wins, achieve in $>90\%$ of the experiments to influence the same number of agents. In very few cases we have results away from the average values.

The same observation applies also in the approximate measures, although here we have a maximum of 90% of the experiments to have the same rank.

The competitions of the adaptive measures achieved very small diffusion level, so the quantile graph shows that only in the experiments between BBCL and approximate Closeness we have a range of values. In that case, most of the

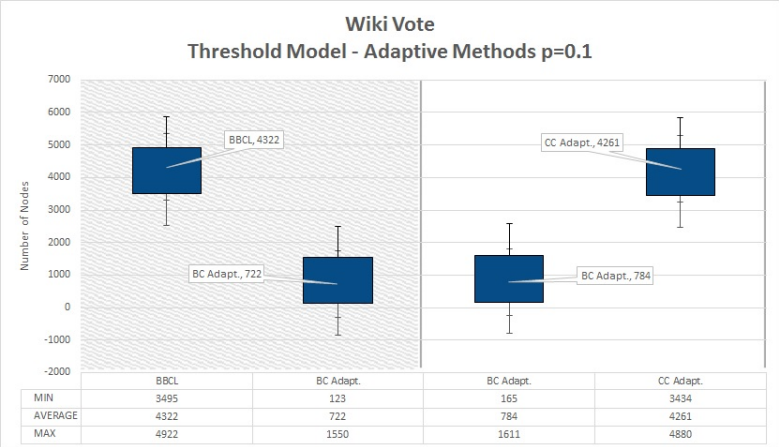


Figure 5.16: Threshold Model - Competitions of the Adaptive Methods with p=0.1

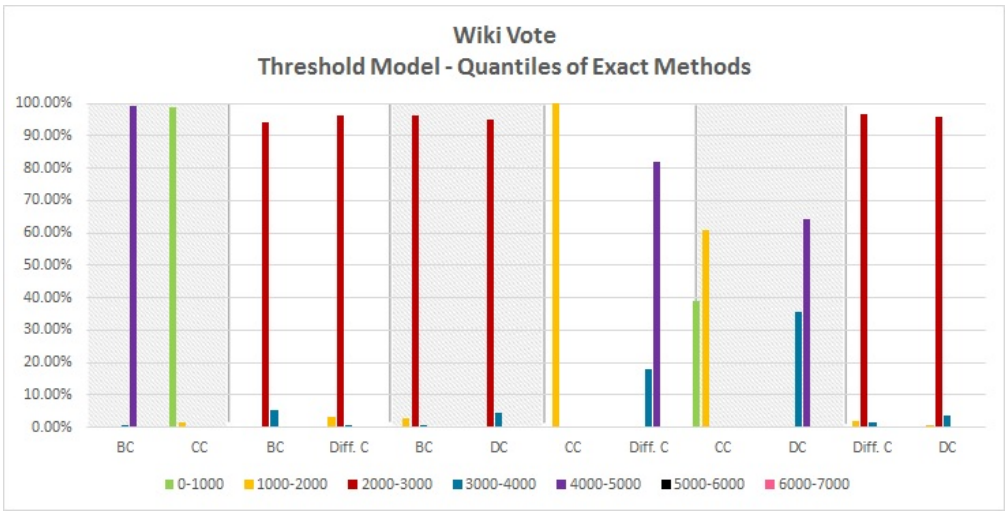


Figure 5.17: Quantiles of the Exact Methods

experiments gives a result of $2000 - 3000$ nodes for each firm.

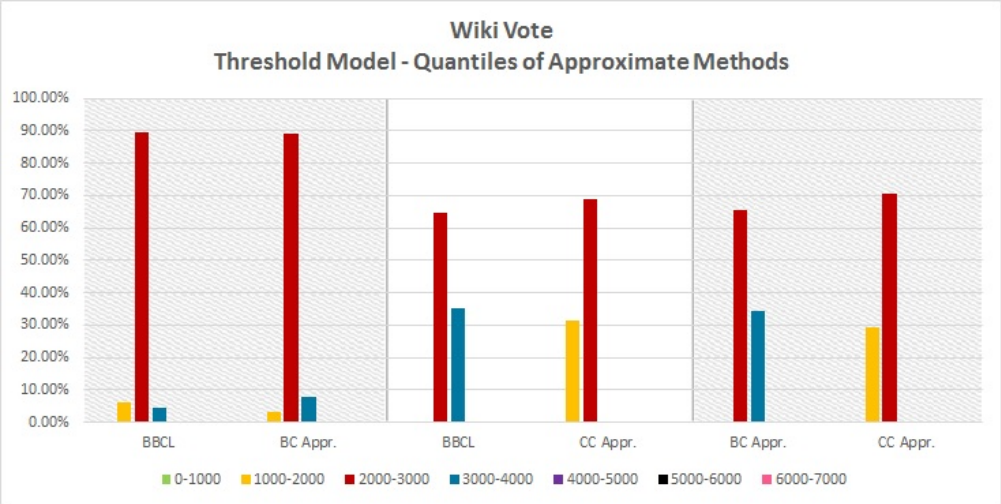


Figure 5.18: Quantiles of the Approximate Methods

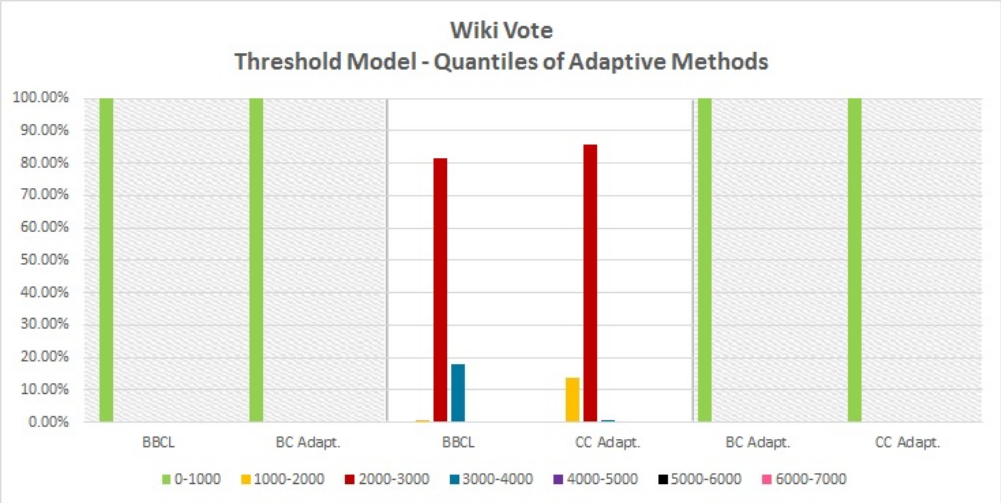


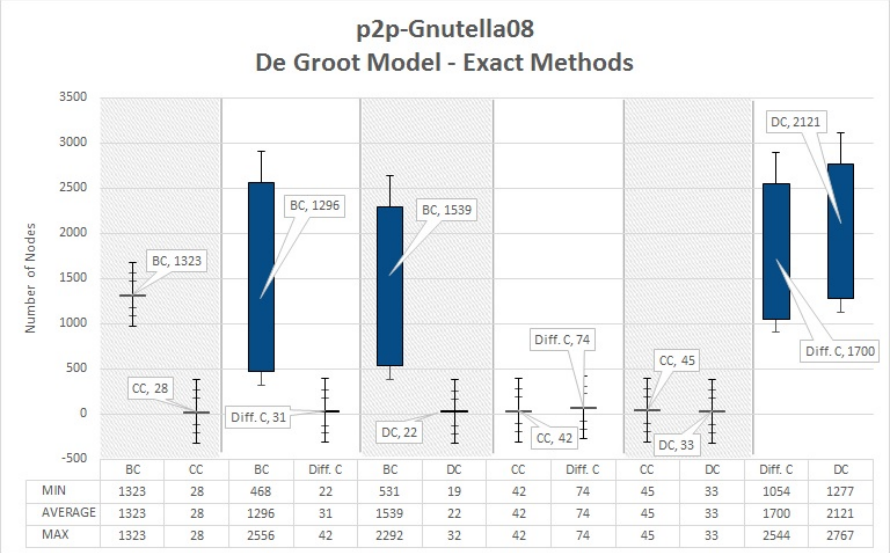
Figure 5.19: Quantiles of the Adaptive Methods

5.3 Peer-to-peer Gnutella

Gnutella is a large peer-to-peer network. It was the first decentralized peer-to-peer network of its kind, leading to other, later networks adopting the model. This graph is derived by a sequence of snapshots of the Gnutella peer-to-peer file sharing network from August 2002. There are total of 9 snapshots of Gnutella network collected in August 2002. It has 6301 nodes and 20777 edges in total, but the giant component that we use in our experiments has 6299 nodes and 20776 edges. This network is directed and its diameter is 9.

5.3.1 De Groot experiments

Regarding the exact methods, BC is the dominant strategy since it wins all his opponents. We observe that CC blocks the diffusion level in every competition that it has. Note that BC has a large diffusion when competes with Diffusion and Degree centralities, which is achieved due to the different pairs of seed sets. CC has not common seed nodes with any of its opponents so the diffusion levels are low. The other strategies seem to be equal.

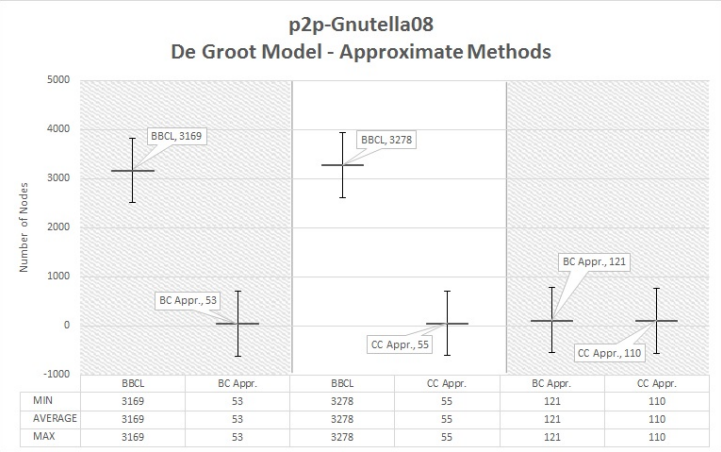


	BC	CC	BC	Diff. C	BC	DC	CC	Diff. C	CC	DC	Diff. C	DC
MIN	1323	28	468	22	531	19	42	74	45	33	1054	1277
AVERAGE	1323	28	1296	31	1539	22	42	74	45	33	1700	2121
MAX	1323	28	2556	42	2292	32	42	74	45	33	2544	2767

Figure 5.20: De Groot Model - Competitions of the Exact Methods

In approximate methods, BBCL is the dominant measure and its seeders achieve to influence a respectable amount of agents. There are no common agents in any of the approximate competitions so the results are the outcome of a single pair of seed sets. In the competition between approximate betweeness and closeness the number of influenced agents is insignificant.

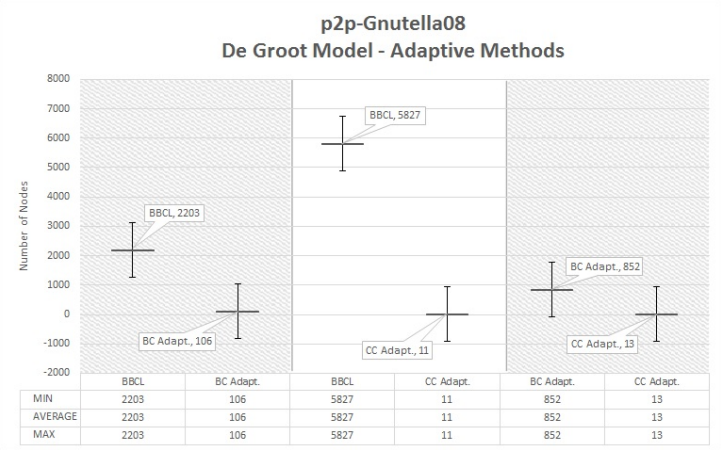
In this category we have almost the same picture as above. BBCL is far better than the others which keep very low levels of influence. The major difference here compared with the previous case, is the fact that we have one competition where BBCL, while competing with Adaptive Closeness centrality, manages to influence a very large amount of agents. It seems that Adaptive Closeness centrality chooses agents far less important than BBCL. On the other hand, Adaptive Betweeness is doing better than Approximate Betweeness and even if it can not win, it limits the spread of its competitors. BBCL manages to influence 2203 agents against Adaptive Betweeness while in the previous chart he got 3169 agents having Approximate BC as opponent. So, we can see that BBCL lost about 1000 agents due to betweeness adaptivity. Furthermore, adaptive BC wins Adaptive CC and influences more than 1/6 of the graph which is way better than the approximate



	BBCL	BC Appr.	BBCL	CC Appr.	BC Appr.	CC Appr.
MIN	3169	53	3278	55	121	110
AVERAGE	3169	53	3278	55	121	110
MAX	3169	53	3278	55	121	110

Figure 5.21: De Groot Model - Competitions of the Approximate Methods

competition of these two measures.



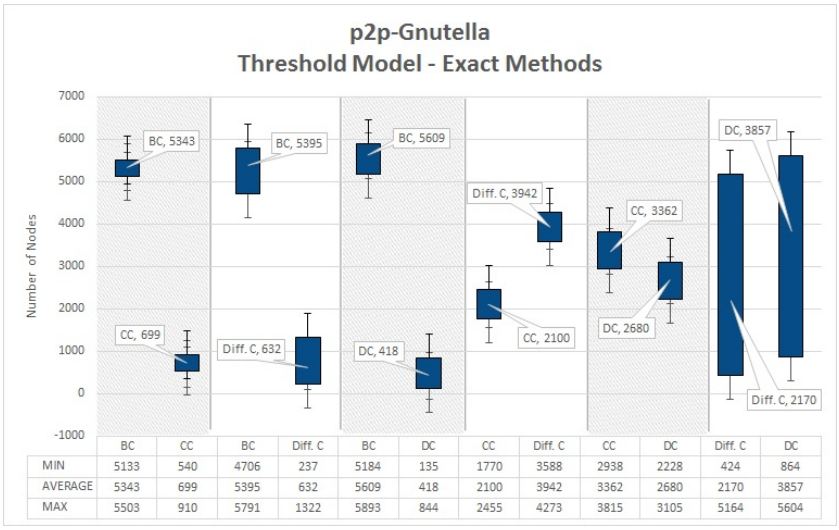
	BBCL	BC Adapt.	BBCL	CC Adapt.	BC Adapt.	CC Adapt.
MIN	2203	106	5827	11	852	13
AVERAGE	2203	106	5827	11	852	13
MAX	2203	106	5827	11	852	13

Figure 5.22: De Groot Model - Competitions of the Adaptive Methods

5.3.2 Threshold experiments

In this graph there are extremely high levels of influence since almost all agents of the graph are influenced from both strategies initial seed sets in every competition. Betweenness centrality is the best choice for a firm in this graph and when it competes there exist a big difference between its influenced agents and the ones of other strategies. Closeness, Diffusion and Degree centrality are considered as equals since they win one another.

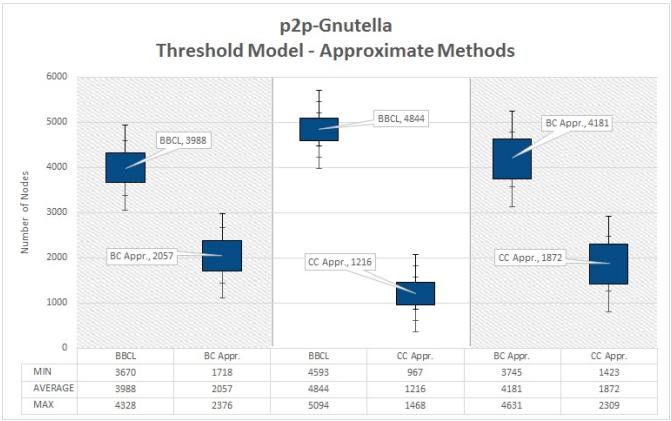
BBLC dominates in approximate competitions and approximate Betweenness centrality comes second. BBCL wins by far the other two competitors so this is



	BC	CC	BC	Diff. C	BC	DC	CC	Diff. C	CC	DC	Diff. C	DC
MIN	5133	540	4706	237	5184	135	1770	3588	2938	2228	424	864
AVERAGE	5343	699	5395	632	5609	418	2100	3942	3362	2680	2170	3857
MAX	5503	910	5791	1322	5893	844	2455	4273	3815	3105	5164	5604

Figure 5.23: Threshold Model - Competitions of the Exact Methods

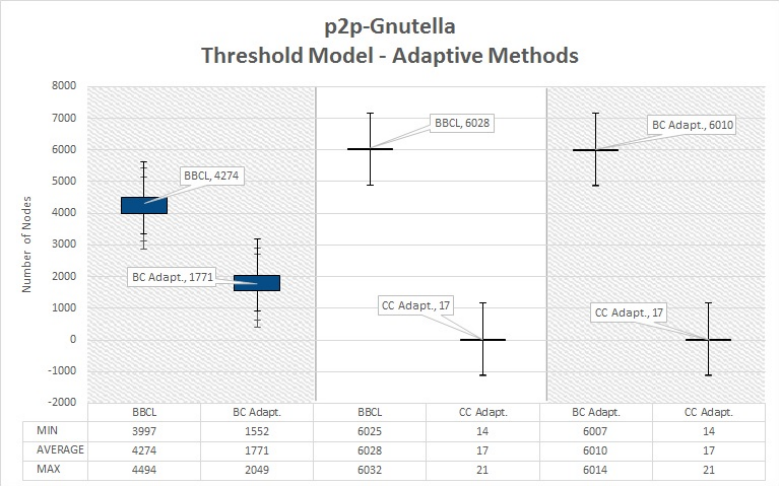
the best choice. The high levels of influence remain but seem a little bit lower than before.



	BBCL	BC Appr.	BBCL	CC Appr.	BC Appr.	CC Appr.
MIN	3670	1718	4593	967	3745	1423
AVERAGE	3988	2057	4844	1216	4181	1872
MAX	4328	2376	5094	1468	4631	2309

Figure 5.24: Threshold Model - Competitions of the Approximate Methods

Once more in adaptive measures we get exactly the same picture with the approximate ones. The high diffusion levels still remain but an interesting aspect here is the fact that apart from the competition between BBLC and Adaptive Betweenness centrality, we have bigger differences between the winner strategies and the loser ones than before. Adaptive Closeness makes no difference to its opponents. It seems like both BBCL and Adaptive BC play alone when they compete with Adaptive CC, and actually influence most of the network.



	BBCL	BC Adapt.	BBCL	CC Adapt.	BC Adapt.	CC Adapt.
MIN	3997	1552	6025	14	6007	14
AVERAGE	4274	1771	6028	17	6010	17
MAX	4494	2049	6032	21	6014	21

Figure 5.25: Threshold Model - Competitions of the Adaptive Methods

5.3.3 Quantiles

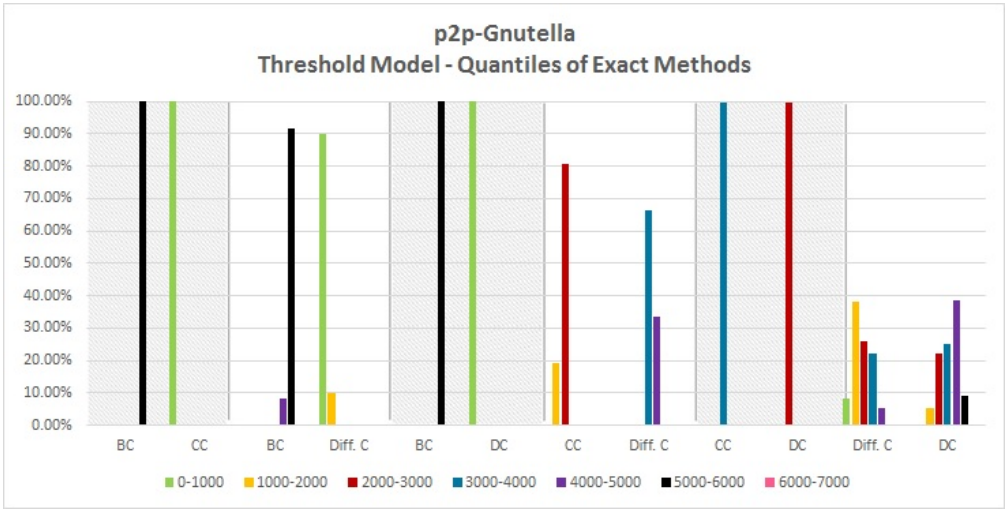


Figure 5.26: Quantiles of the Exact Methods

In the first three competitions, it is obvious that BC is the best strategy and the results of almost the 100% of the experiments shows that manages to influence the same number of nodes regardless the seed set. The same observation stands also for the competitions among CC and the other strategies. On the other hand, in the Diffusion VS Degree competition, values can not gather in an interval but spread from 0-5000 in every case. This shows that different results can be observed, depending on the final seed set that each firm have selected. Furthermore, between these two measures, neither of them can dominate the other in this network.

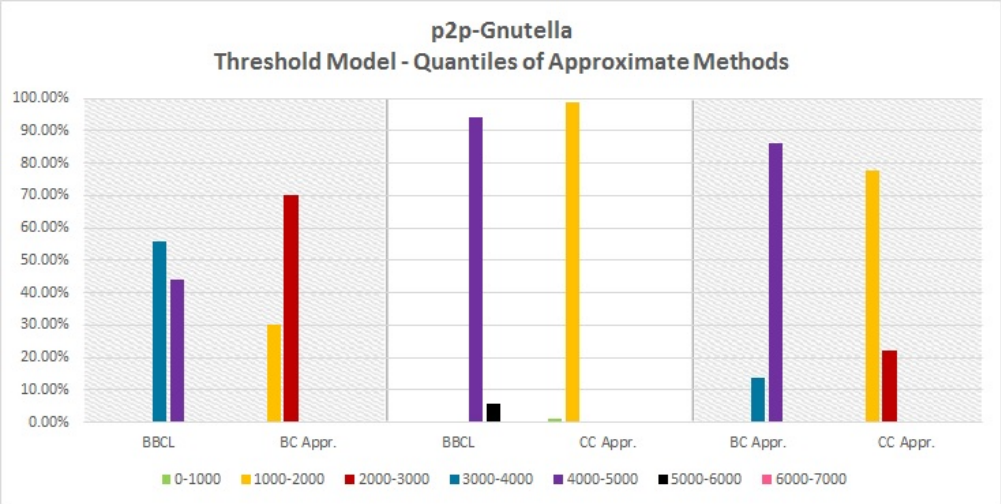


Figure 5.27: Quantiles of the Approximate Methods

The approximate measures are acting better, and they manage to keep the resulting values at most cases, in an interval. We also observe a level of expansion but it can not compare to the previous case.

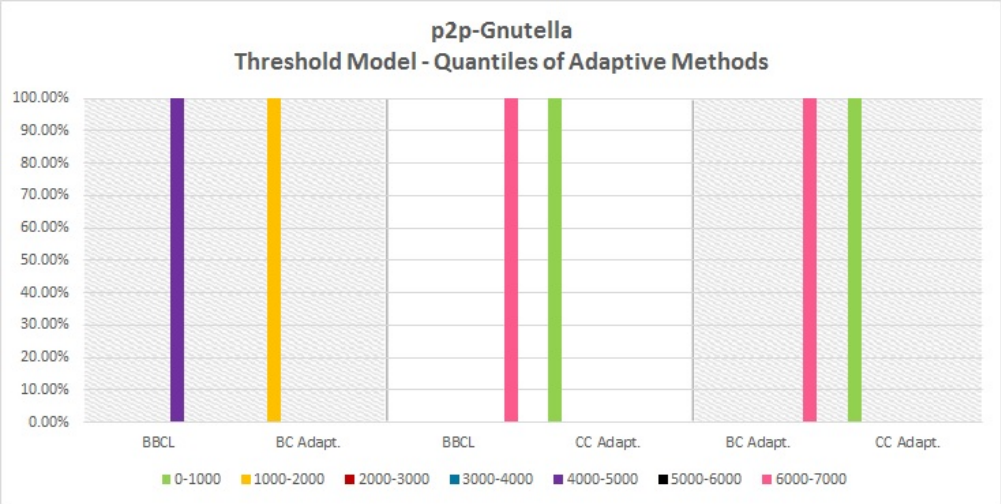


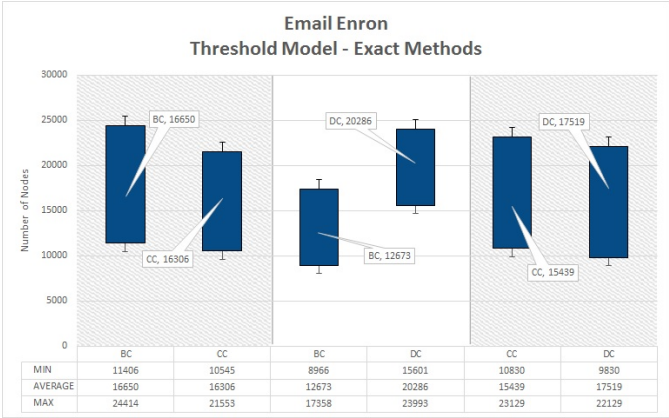
Figure 5.28: Quantiles of the Adaptive Methods

In Section 5.3.2 we have shown that Adaptive Closeness is by far the worst of the adaptive techniques in this network, since it achieves to influence at most 7 more agents except for the seeders. This outcome can be also proven from the quantile graph which shows that the opponent of Adaptive CC gets all the network. In the competition among BBCL and Adaptive BC they both stay in a limited interval on the 100% of the experiments and extreme cases do not exist.

5.4 Email Enron

Enron email communication network covers all the email communication within a dataset of around half million emails. This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation. Nodes of the network are email addresses and if an address i sent at least one email to address j, the graph contains an undirected edge from i to j. Note that non-Enron email addresses act as sinks and sources in the network as we only observe their communication with the Enron email addresses. The total number of nodes is 36692 and the total number of edges is 367622. Since we use only the giant component we are restricting our experiments to 33696 nodes and to 361622 edges in total.

5.4.1 Threshold experiments



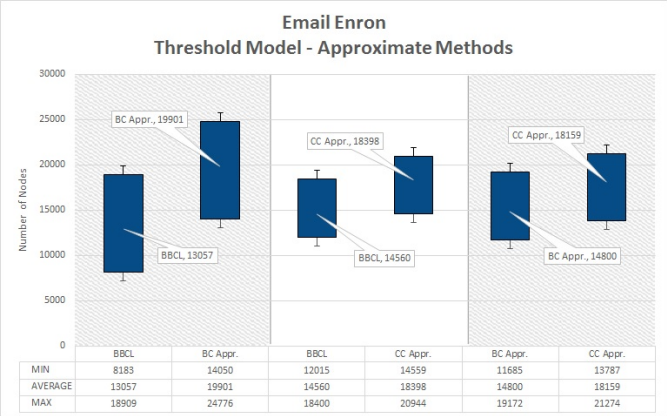
	BC	CC	BC	DC	CC	DC
MIN	11406	10545	8966	15601	10830	9830
AVERAGE	16650	16306	12673	20286	15439	17519
MAX	24414	21553	17358	23993	23129	22129

Figure 5.29: Threshold Model - Competitions of the Exact Methods

In Figure 5.29 the exact competitions of Email Enron are represented. Degree centrality is the dominant strategy here and Betweenness centrality comes second. In every competition, measures have many common agents. We had to apply the excluding rule and the experiments were performed in ten different pairs of seed sets. Nevertheless, both competitors manage to influence almost the entire graph in every competition.

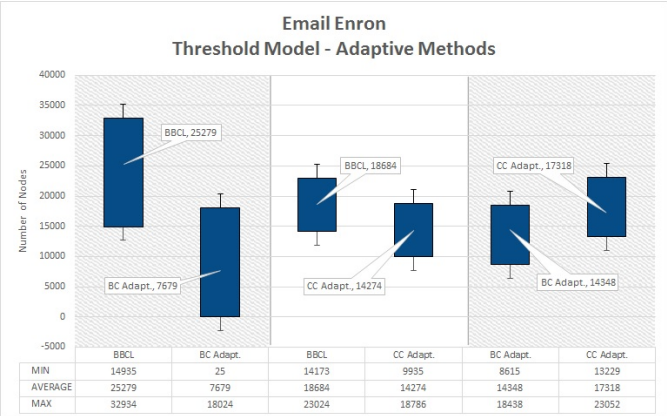
In the games between the approximate methods Betweenness centrality is the best strategy and Closeness comes second. Here there are a little bit bigger differences between the winner and the loser strategy than before, but the high levels of diffusion remain, which seems logical since, in contrast to the exact methods, in approximate methods there was no application of the excluding rule.

It is obvious from Figure 5.31 that in the adaptive methods we get a different picture. The differences between the winner and the loser strategy are even higher than before but the influential levels remain high. BBLC is the dominant strategy



	BBCL	BC Appr.	BBCL	CC Appr.	BC Appr.	CC Appr.
MIN	8183	14050	12015	14559	11685	13787
AVERAGE	13057	19901	14560	18398	14800	18159
MAX	18909	24776	18400	20944	19172	21274

Figure 5.30: Threshold Model - Competitions of the Approximate Methods



	BBCL	BC Adapt.	BBCL	CC Adapt.	BC Adapt.	CC Adapt.
MIN	14935	25	14173	9935	8615	13229
AVERAGE	25279	7679	18684	14274	14348	17318
MAX	32934	18024	23024	18786	18438	23052

Figure 5.31: Threshold Model - Competitions of the Adaptive Methods

and Adaptive Closeness comes second. Adaptive Betweeness does not achieve much diffusion, although the Approximate Betweeness gives better results. An explanation for this outcome is that adaptive methods select agents in different neighborhoods in order to achieve wider diffusion. This is not always efficient though. In some cases, the agents which are selected belong to small neighborhoods and their influence is limited.

5.4.2 Quantiles

In the first two cases, those of the exact and the approximate competitions, the outcome of the 10.000 experiments is as we expected. The influenced agents fluctuate around the average value which we can see in 5.29, 5.30 respectively. Extreme cases do not exist here.

In adaptive measures we have a slightly different picture. Especially on the first competition of BBCL with approximate Betweeness, we have a rather large

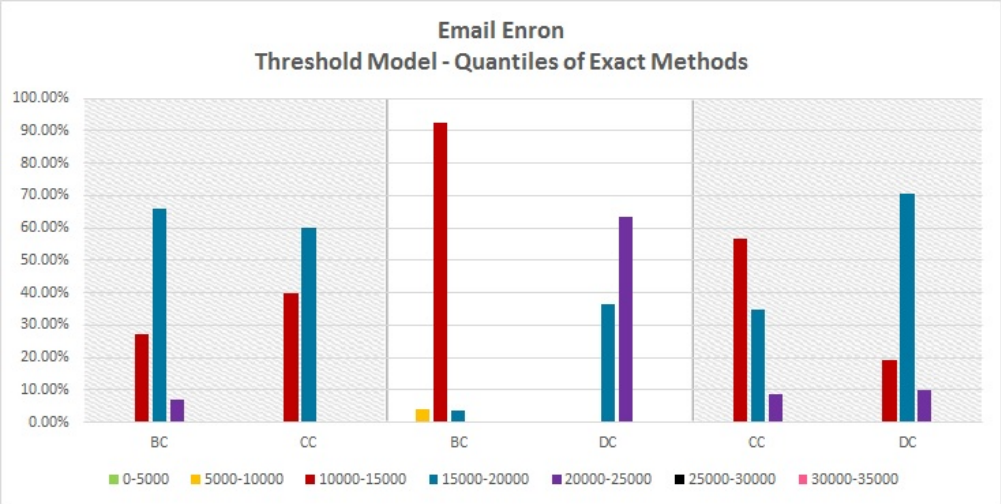


Figure 5.32: Quantiles of the Exact Methods

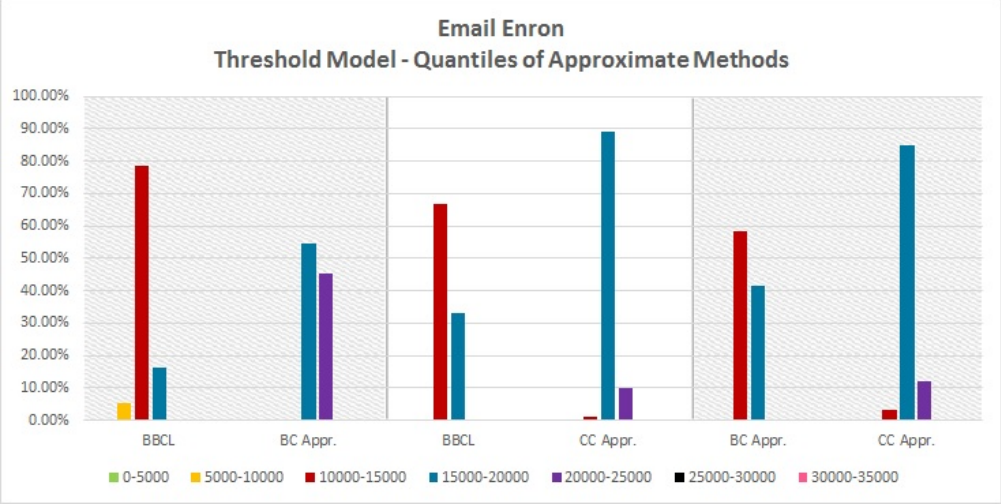


Figure 5.33: Quantiles of the Approximate Methods

diffusion of the resulted values. This happens due to the different pairs of seed sets that are selected and we can not avoid it. However, on average it is obvious that BBCL wins, even when it chooses the worst seed set.

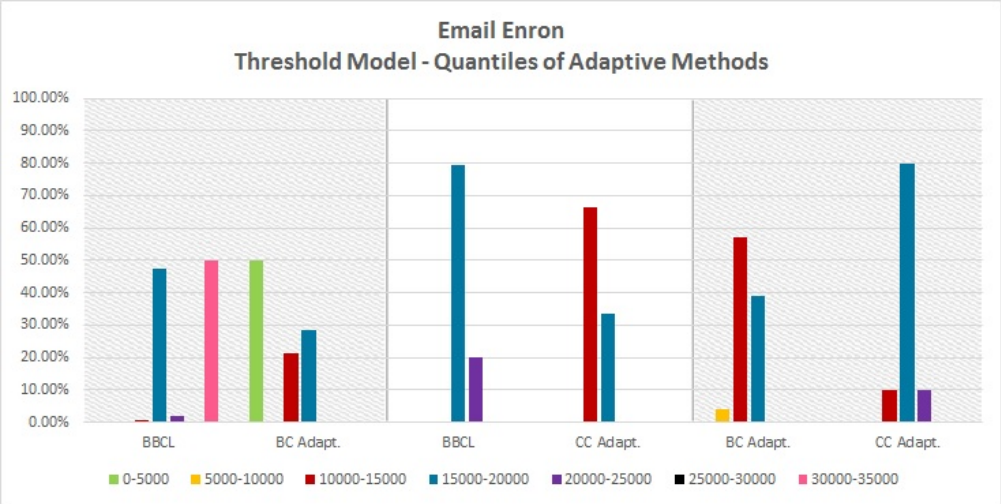


Figure 5.34: Quantiles of the Adaptive Methods

5.5 Slashdot

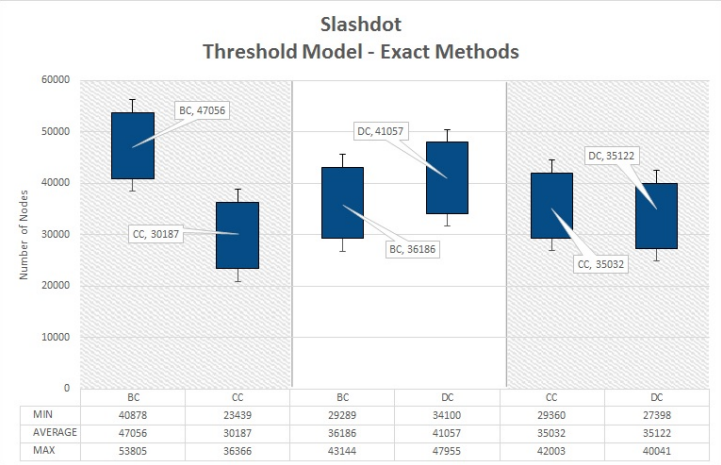
Slashdot is a technology-related news website know for its specific user community. The website features user-submitted and editor-evaluated current primarily technology oriented news. In 2002 Slashdot introduced the Slashdot Zoo feature which allows users to tag each other as friends or foes. The network contains friend/foe links between the users of Slashdot. The network was obtained in November 2008 and has 77360 nodes and 905468 edges. Fortunately we use the whole graph in this case since there isn't separated components in the graph.

5.5.1 Threshold experiments

As it is shown in Figure 5.35, there is an interesting case here. Degree centrality is doing better than the others if we count the times that it wins. However, BC manages to influence the largest part of the network in its own competitions. The diffusion levels are extremely high, since in every game both strategies influence almost the whole graph.

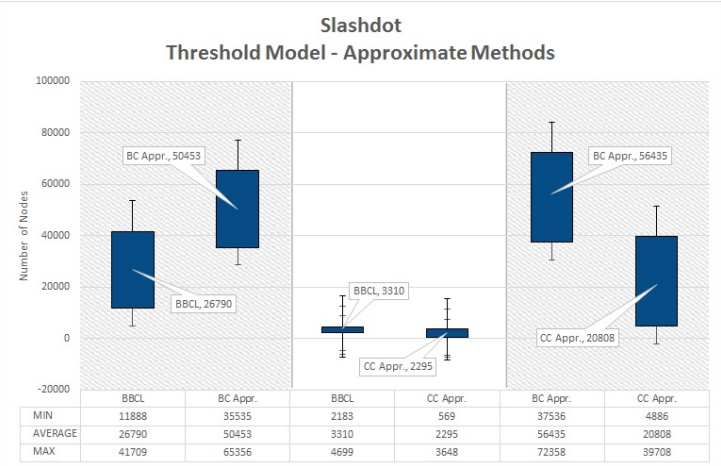
In approximation methods, Betweeness can be considered as the best choice by far. When approximate BC competes with the other methods, manages to influence 2/3 of the network by average. BBCL comes second, however when it competes with closeness neither of them achieves much diffusion. In this case the reason of the small diffusion level is the threshold value θ_v. The seed nodes which are selected by the two firms are far from each other in the graph and this means that the value $\theta_v = 0.2$ that we use to all experiments, is not reached.

To understand that, we tested the competition of these two competitors, on the same seed sets and with the only difference to be that $\theta_v = 0.1$ and the results



	BC	CC	BC	DC	CC	DC
MIN	40878	23439	29289	34100	29360	27398
AVERAGE	47056	30187	36186	41057	35032	35122
MAX	53805	36366	43144	47955	42003	40041

Figure 5.35: Threshold Model - Competitions of the Exact Methods



	BBCL	BC Appr.	BBCL	CC Appr.	BC Appr.	CC Appr.
MIN	11888	35535	2183	569	37536	4886
AVERAGE	26790	50453	3310	2295	56435	20808
MAX	41709	65356	4699	3648	72358	39708

Figure 5.36: Threshold Model - Competitions of the Approximate Methods

are represented in Figure 5.37.

Next we examine the adaptive measures. BBLC is the dominant strategy here and Adaptive Betweenness centrality is the next best choice. Unfortunately in all games we get much lower diffusion levels, with the lower one between Adaptive Betweenness and Adaptive Closeness, where the initial seed sets from both strategies manage to influence only 641 agents in total out of 77360 agents.

Again, the threshold value $\theta_v = 0.2$ is not enough to achieve a significant level of diffusion. In these experiments the firms do not have common seeders, so there is only one pair of seed set that is examined. The only two opponents that have one seeder in common are BBCL and approximate betweeness. Below we show the outcome of the competitions with $\theta_v = 0.1$.

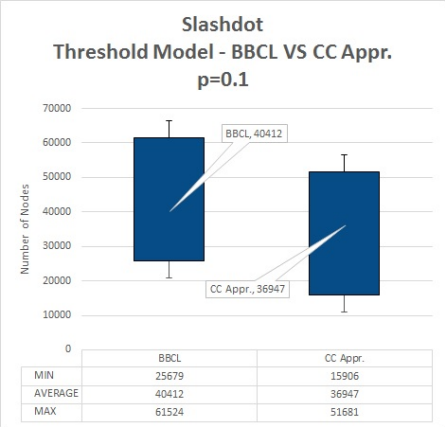


Figure 5.37: Threshold Model - BBCL VS Approximate CC

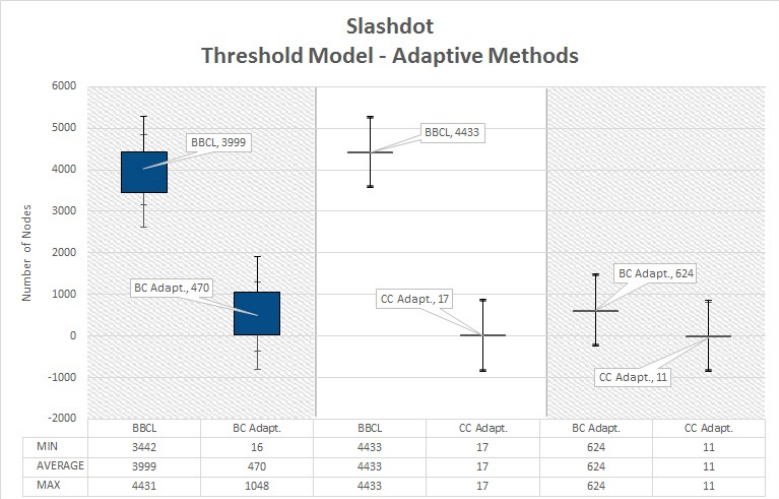
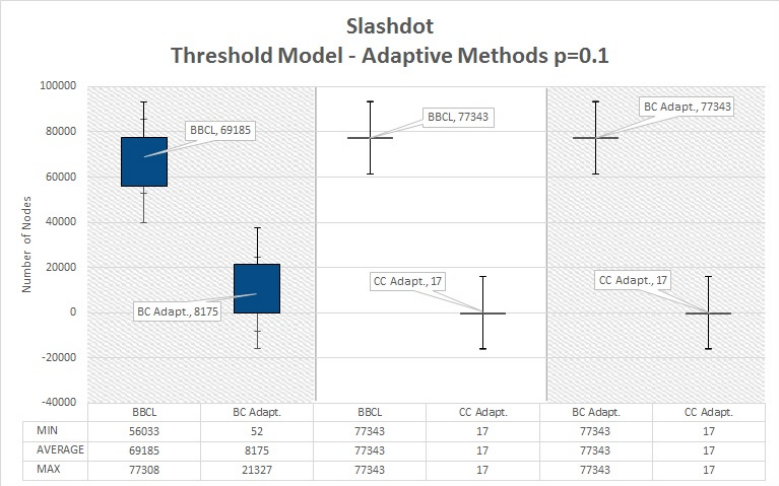


Figure 5.38: Threshold Model - Competitions of the Adaptive Methods

5.5.2 Quantiles

The first two competitions of Betweeness with the other two measures are balanced and values fluctuate in reasonable range. In the competition between Closeness and Degree Centralities, we notice a larger range on the results. In this case, there is one pair of seed set that achieves minimum diffusion for both metrics. Apparently when it happens to choose this specific seed set, they can not reach the threshold value $\theta_v = 0.2$ that is required to spread their influence.

In Figure 5.41, the only odd case is this of the competition of BBCL with approximate Closeness. As we proved in Section 5.5.1 this is also due to the threshold value $\theta_v = 0.2$, and as it is indicated in 5.37, the diffusion level is growing when θ_v is decreasing.



	BBCL	BC Adapt.	BBCL	CC Adapt.	BC Adapt.	CC Adapt.
MIN	56033	52	77343	17	77343	17
AVERAGE	69185	8175	77343	17	77343	17
MAX	77308	21327	77343	17	77343	17

Figure 5.39: Threshold Adaptive Methods ($p = \theta_v = 0.1$)

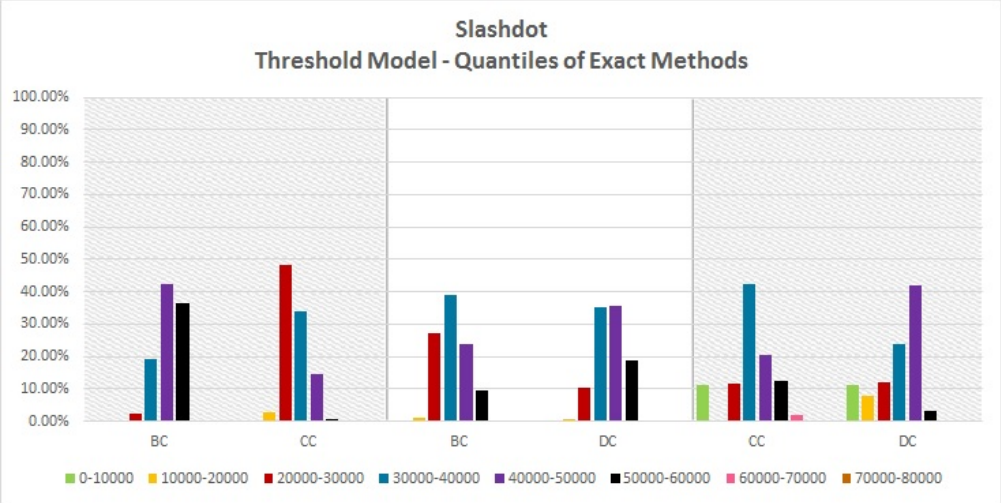


Figure 5.40: Quantiles of the Exact Methods

The same observation stands also, in the adaptive competitions. The diffusion levels are highly increased when threshold value is decreased. So, the quantile graph here can not show the spread of the results.

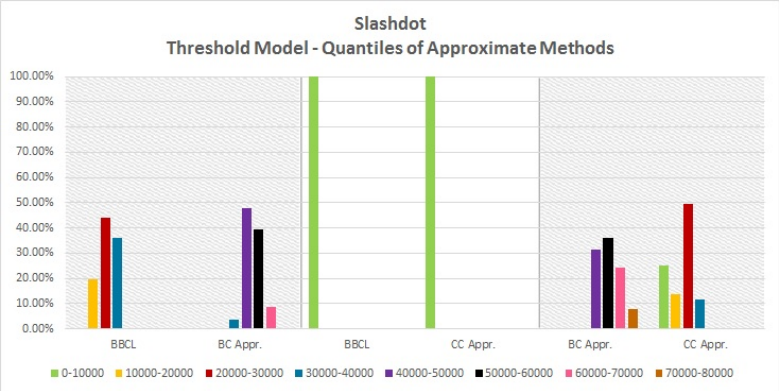


Figure 5.41: Quantiles of the Approximate Methods

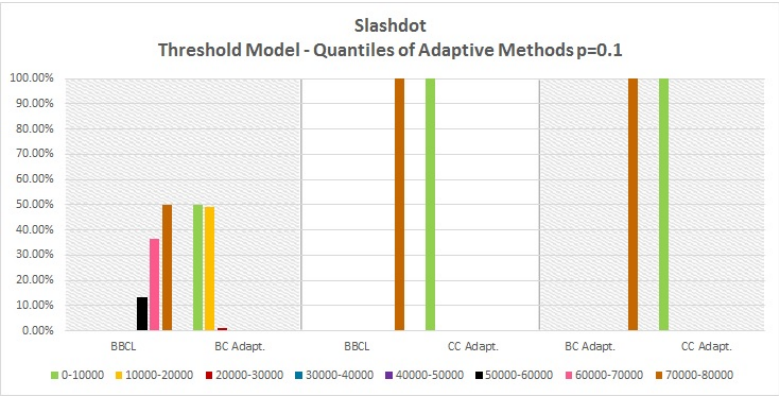


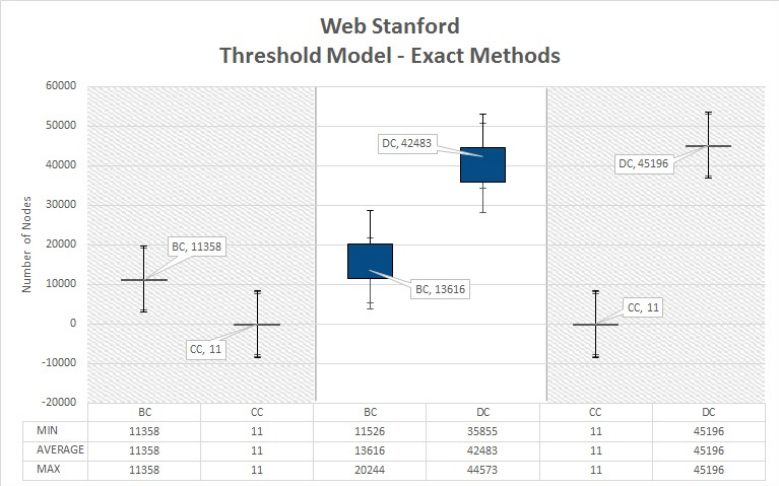
Figure 5.42: Quantiles of the Adaptive Methods

5.6 Web Stanford

Nodes represent pages from Stanford University (stanford.edu) and directed edges represent hyperlinks between them. The data was collected in 2002 and include 281903 nodes and 2312497 edges. We let the Threshold model operate in the giant component of 255265 nodes and 2234572 edges. The diameter of Web Stanford is 674.

5.6.1 Threshold experiments

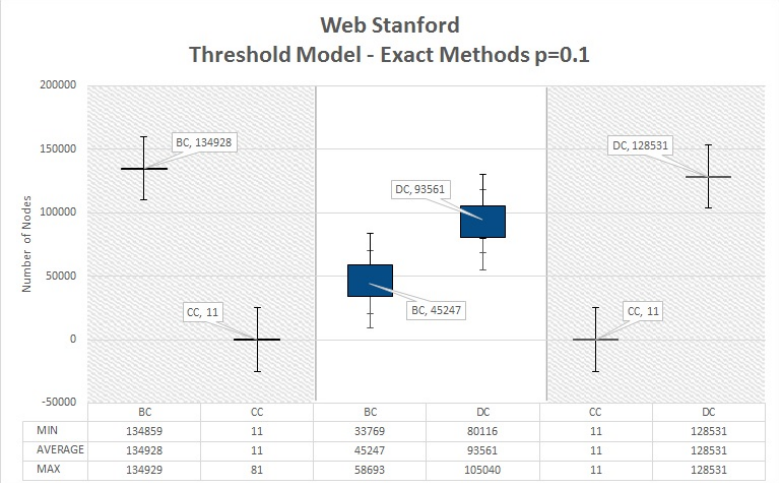
From Figure 5.43 it is obvious that Degree Centrality is by far the best strategy in this graph. Betweenness Centrality is slightly better than Closeness Centrality but the strange thing here is the fact that all strategies are reaching very low influence levels proportionally to the size of the graph. Web Stanford is a graph that exhibits very large number of triangles and very high average clustering coefficient. At the same time this graph has the bigger diameter (674) than any other graph we have



	BC	CC	BC	DC	CC	DC
MIN	11358	11	11526	35855	11	45196
AVERAGE	11358	11	13616	42483	11	45196
MAX	11358	11	20244	44573	11	45196

Figure 5.43: Threshold Model - Competitions of the Exact Methods

examined. So combining these factors we can say that graph's agents have the tendency to form relatively small clusters with few edges to link one cluster to another. Thus in such a large graph with many clusters it is difficult to have a widespread diffusion throughout the network from 10 seeders and a threshold value $\theta_v = 0.2$.

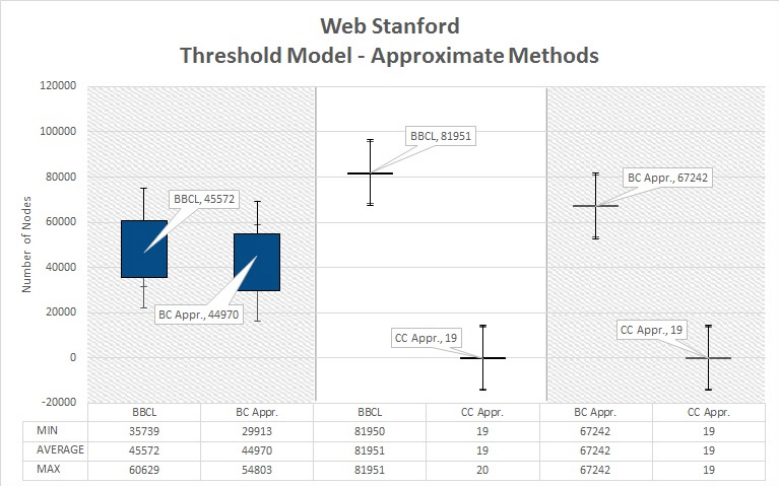


	BC	CC	BC	DC	CC	DC
MIN	134859	11	33769	80116	11	128531
AVERAGE	134928	11	45247	93561	11	128531
MAX	134929	81	58693	105040	11	128531

Figure 5.44: Threshold Model - Competitions of the Exact Methods ($p = \theta_v = 0.1$)

We have run the same experiments for this graph with smaller threshold ($\theta_v = 0.1$) and in every game the seed sets from both strategies achieve much higher influential levels than before. In fact, they achieve influence twice as large, compared to the degree of influence achieved before.

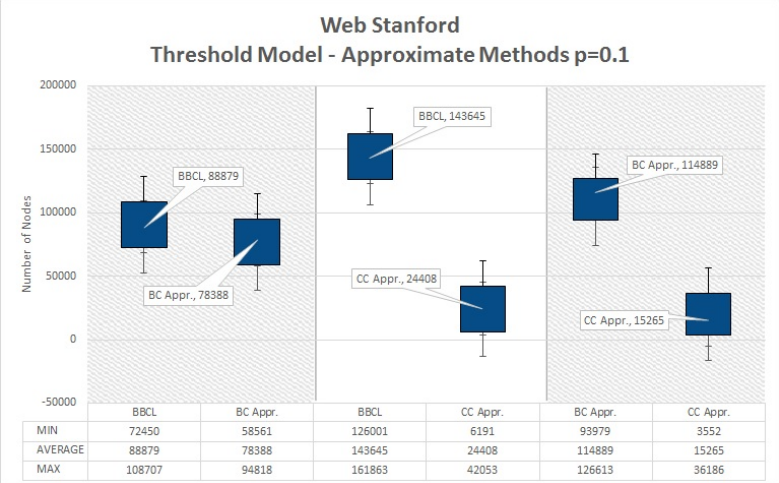
In the games between the approximate methods, BBLC dominates the other two



	BBCL	BC Appr.	BBCL	CC Appr.	BC Appr.	CC Appr.
MIN	35739	29913	81950	19	67242	19
AVERAGE	45572	44970	81951	19	67242	19
MAX	60629	54803	81951	20	67242	19

Figure 5.45: Threshold Model - Competitions of the Approximate Methods

strategies. There are higher levels of influence here compared to the exact methods, but still the higher one barely exceeds the 30% of agents of the giant component. Additionally, there was no application of the excluding rule in any of our strategy pairs. In general, the larger a graph is the better is doing in approximate measures. Nevertheless in any case Approximate Closeness and Betweenness cannot dominate over BBLC.

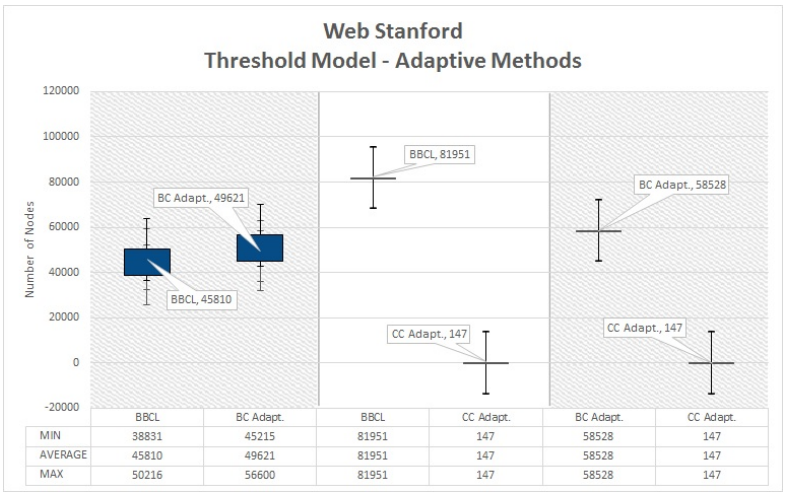


	BBCL	BC Appr.	BBCL	CC Appr.	BC Appr.	CC Appr.
MIN	72450	58561	126001	6191	93979	3552
AVERAGE	88879	78388	143645	24408	114889	15265
MAX	108707	94818	161863	42053	126613	36186

Figure 5.46: Threshold Model - Competitions of the Approximate Methods ($p = \theta_v = 0.1$)

Finally, in adaptive methods we have exactly the same picture that we had before. The only difference is the fact that adaptive Closeness centrality is doing slightly better than Approximate Closeness centrality, but the very low diffusion

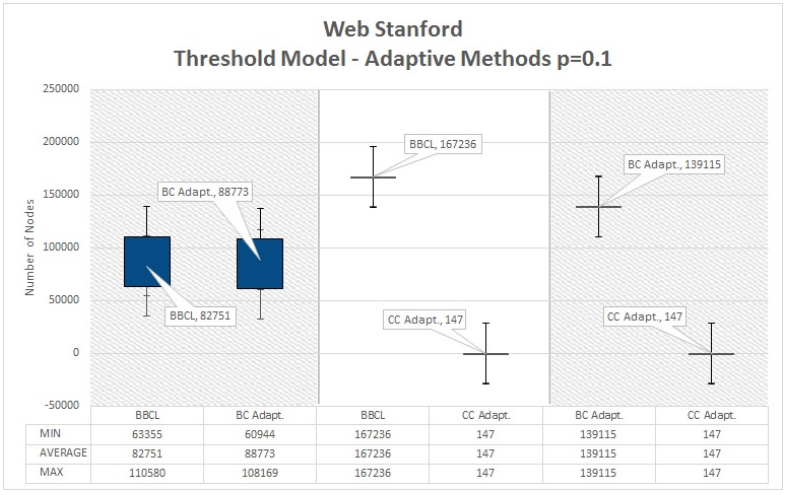
levels remain here as well.



	BBCL	BC Adapt.	BBCL	CC Adapt.	BC Adapt.	CC Adapt.
MIN	38831	45215	81951	147	58528	147
AVERAGE	45810	49621	81951	147	58528	147
MAX	50216	56600	81951	147	58528	147

Figure 5.47: Threshold Model - Competitions of the Adaptive Methods

Below, are represented the results for the adaptive methods with threshold value $p = \theta_v = 0.1$. As we can see, the diffusion levels are much better as θ_v increases.



	BBCL	BC Adapt.	BBCL	CC Adapt.	BC Adapt.	CC Adapt.
MIN	63355	60944	167236	147	139115	147
AVERAGE	82751	88773	167236	147	139115	147
MAX	110580	108169	167236	147	139115	147

Figure 5.48: Threshold Model - Competitions of the Adaptive Methods ($p = \theta_v = 0.1$)

5.6.2 Quantiles

In this section, we provide quantile graphs for the experiments with $\theta_v = 0.1$ because in the case that $\theta_v = 0.2$ the diffusion level is too low and a quantile graph won't show us the spread of the results.

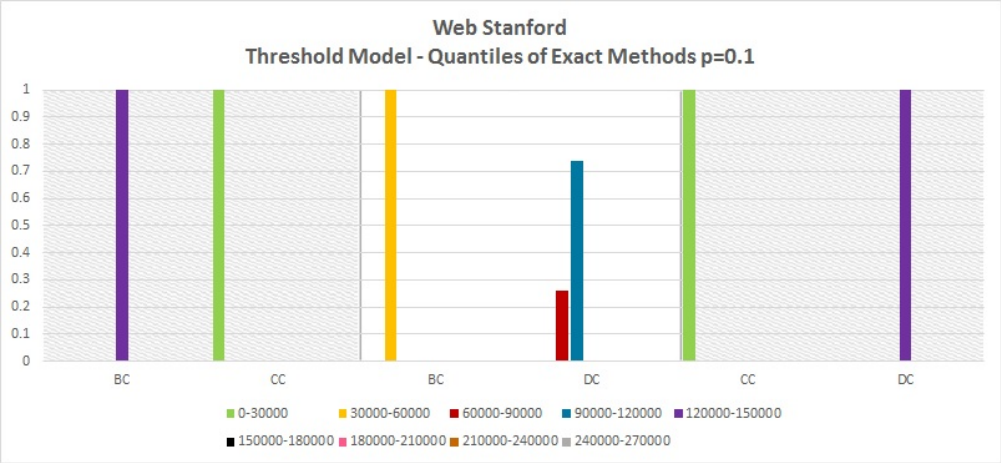


Figure 5.49: Quantiles of the Exact Methods ($p = \theta_v = 0.1$)

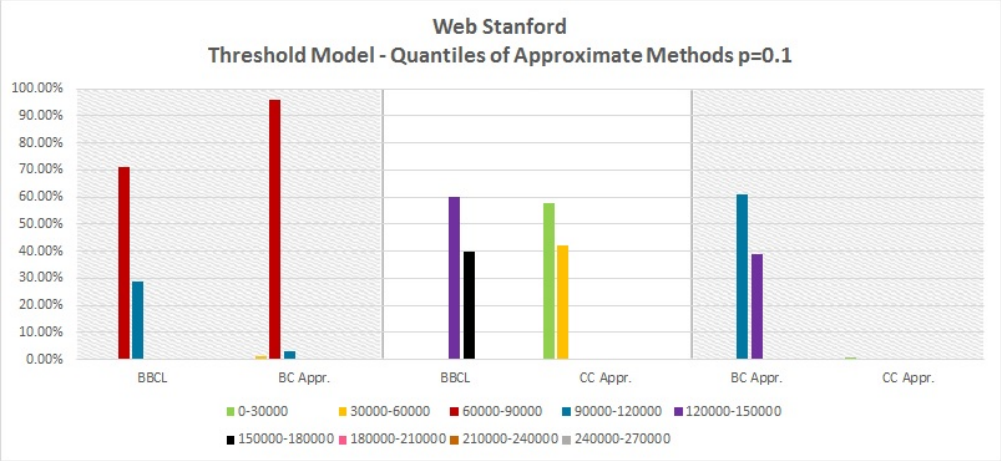


Figure 5.50: Quantiles of the Approximate Methods ($p = \theta_v = 0.1$)

In Figures 5.49, 5.50 and 5.51, we can see that most of the experiments produce similar results, all within a range of values. There are not extreme situations in any of the competitions and the winner, at most cases, stands out.

5.7 Results & Observations

The experiments were performed on six different networks, each with its own characteristics and different size. We examined directed and undirected, sparse and dense graphs, and each one represented a different community. The competitions performed over two diffusion techniques, the De Groot model and the Threshold model. So, at this point, we have to answer two questions. 1) Which of the

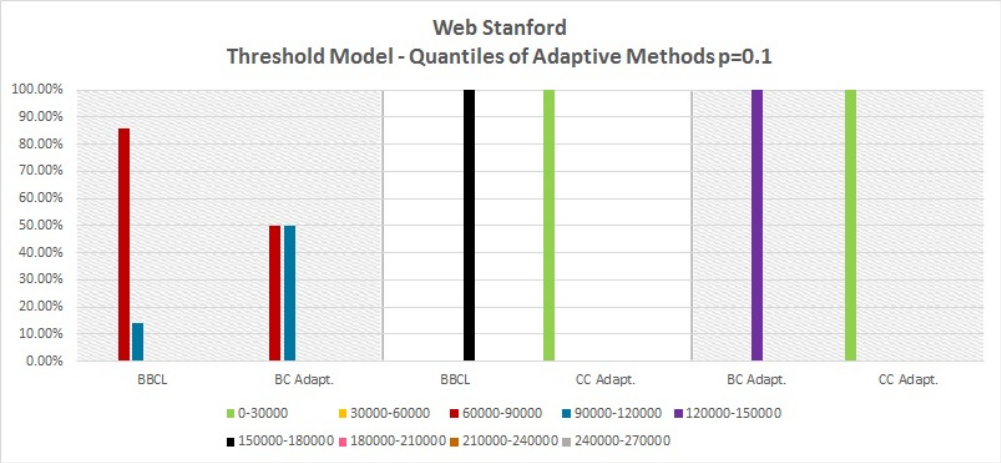


Figure 5.51: Quantiles of the Adaptive Methods ($p = \theta_v = 0.1$)

diffusion models achieves better results? and 2) Do we have a dominant strategy which wins in any case all the others?

It is much easier to answer the first question. In every experiment, the De Groot model needs significantly more time to run than the Threshold model. Furthermore, in many cases they both achieve the same level of diffusion. However, the Threshold model manages to spread the innovations to much larger parts of the network, in most of the competitions. Thus, the Threshold model not only runs way faster that the De Groot model, but achieves to maximize the influence, which is exactly what we were looking for.

To answer the second question we should define what we mean when we say that a strategy "wins". There are two options that we consider with the term "victory". On the one hand, one may define as victory the number of times that a strategy manages to win another strategy within the same network. This way of thinking is very simple and does not count the level of the diffusion that each strategy can achieve among the competitions. This is the second option, to consider as winner the strategy that manages the largest level of influence among all the competitions in the same network. Below, we represent the decreasing order of the strategies, considering **a) the number of victories** and **b) the level of influence**.

In Exact Measures, at most of the competitions Degree Centrality is the dominant measure, and Betweeness Centrality comes second. We observe that DC wins in larger graphs which is more useful information. Power law graphs are large and sparse networks so we need a metric that can influence a large part of the network at most cases.

In approximate measures it seems that BBCL and Approximate Betweeness are competing to win the first place. They both manage to influence many agents and in large graphs they both doing very well. Thus, although BBCL wins every time

	EXACT METHODS			
	1st	*2nd*	*3rd*	*4th*
Ca-GrQc	BC	CC	DC	DIFF. C
Wiki Vote	DC	BC	DIFF. C	CC
p2p-Gnutella	BC	CC≃DIFF. C≃DC		
Email Enron	DC	BC	CC	
Slashdot	DC	BC	CC	
Web Stanford P=0.2	DC	BC	CC	
Web Stanford P=0.1	DC	BC	CC	

Figure 5.52: Decreasing order of the exact strategies depending on the number of victories

	APPROXIMATE METHODS		
	1st	*2nd*	*3rd*
Ca-GrQc	BBCL	BC Appr.	CC Appr.
Wiki Vote	BBCL	BC Appr.	CC Appr.
p2p-Gnutella	BBCL	BC Appr.	CC Appr.
Email Enron	CC Appr.	BC Appr.	BBCL
Slashdot	BC Appr.	BBCL	CC Appr.
Web Stanford P=0.2	BBCL	BC Appr.	CC Appr.
Web Stanford P=0.1	BBCL	BC Appr.	CC Appr.

Figure 5.53: Decreasing order of the approximate strategies depending on the number of victories

	ADAPTIVE METHODS		
	1st	*2nd*	*3rd*
Ca-GrQc	BBCL	CC Adapt.	BC Adapt.
Wiki Vote	BBCL	CC Adapt.	BC Adapt.
p2p-Gnutella	BBCL	BC Adapt.	CC Adapt.
Email Enron	BBCL	CC Adapt.	BC Adapt.
Slashdot	BBCL	BC Adapt.	CC Adapt.
Web Stanford P=0.2	BBCL	BC Adapt.	CC Adapt.
Web Stanford P=0.1	BC Adapt.	BBCL	CC Adapt.

Figure 5.54: Decreasing order of the adaptive strategies depending on the number of victories

in smaller graphs, we can consider that these measures are almost equal concerning the influence that they achieve.

Finally, in the competitions among the adaptive measures, results are pretty clear. BBCL wins the other two adaptive metrics in every network we examined.

This is certainly the dominant strategy in this category since it manages to win regardless the topology and the characteristics of the network. the other two measures can be considered equal with BC being slighly better than CC.

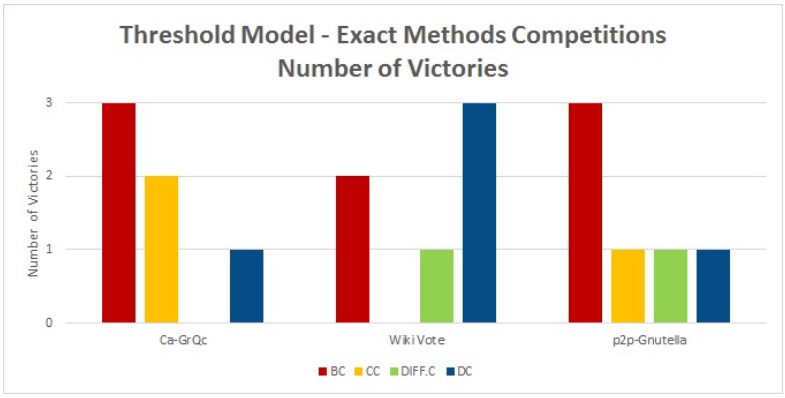


Figure 5.55: Number of victories of the exact strategies in small networks

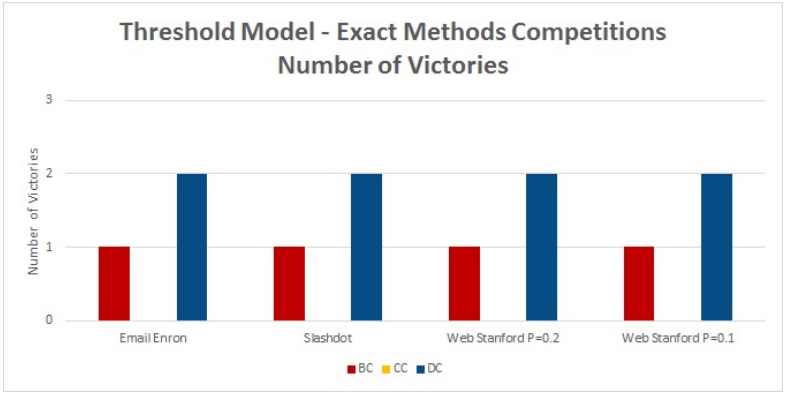


Figure 5.56: Number of victories of the exact strategies in large networks

This arrangement is due to the number that each strategy wins the others. In the second case we have the dominant strategies according to the level of diffusion that they achieve among all their competitions.

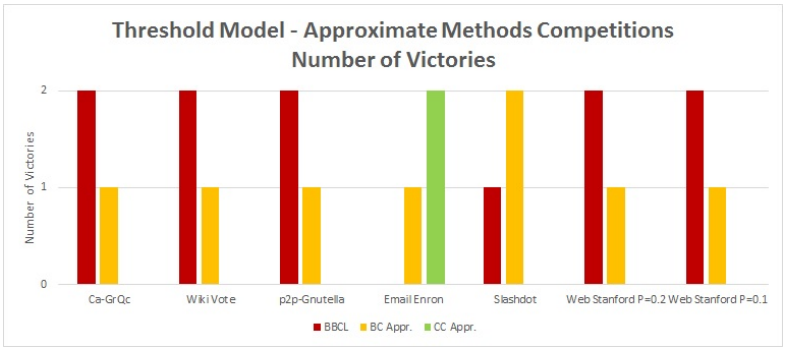


Figure 5.57: Number of victories of the approximate strategies

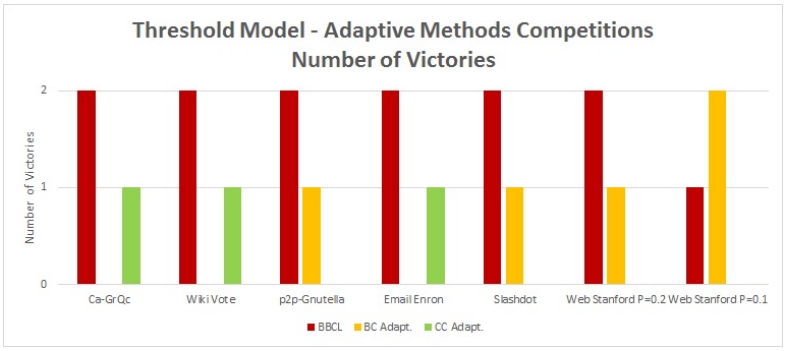


Figure 5.58: Number of victories of the adaptive strategies

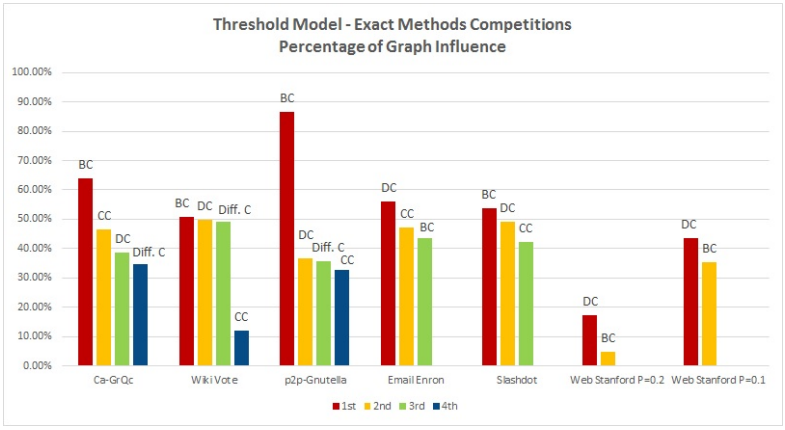


Figure 5.59: Percentage of diffusion level of the exact strategies

In Figures 5.59, 5.60 and 5.61 we see the results according to the diffusion level in each network. In general, the dominant strategies are those which have the most wins against the others. However, in some cases, a strategy that achieves 2/3 wins, manages to spread in a largest part of the network. Since we care

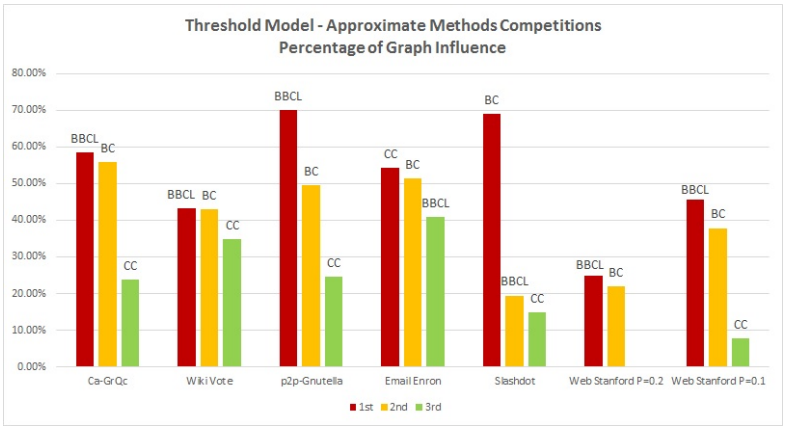


Figure 5.60: Percentage of diffusion level of the approximate strategies

about the level of the spread that can be achieved in every competition, this is a more suitable methodology to define the winners.

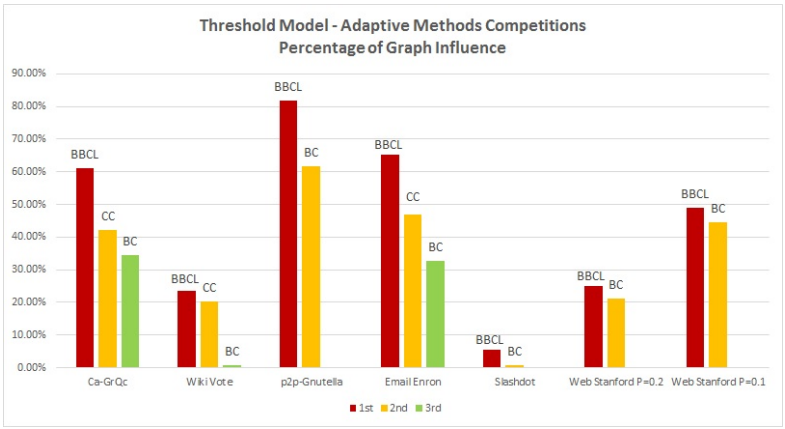


Figure 5.61: Percentage of diffusion level of the adaptive strategies

5.8 Dominant Strategies Competitions

An interesting experiment is to examine the competition between the dominant strategies of two different categories of Centrality measures. In the exact strategies we have either BC or DC to be the dominant one, depending on the network. On the other hand, in adaptive techniques, BBCL is always the best choice regardless the graph. So, we demonstrate the results of these competitions in Figure 5.62.

As we can see, exact measures wins the adaptive ones almost in every graph. There are two exceptions, the peer-to-peer and the Web Stanford network, where we have the the opposite result. An adaptive technique dominates the best exact

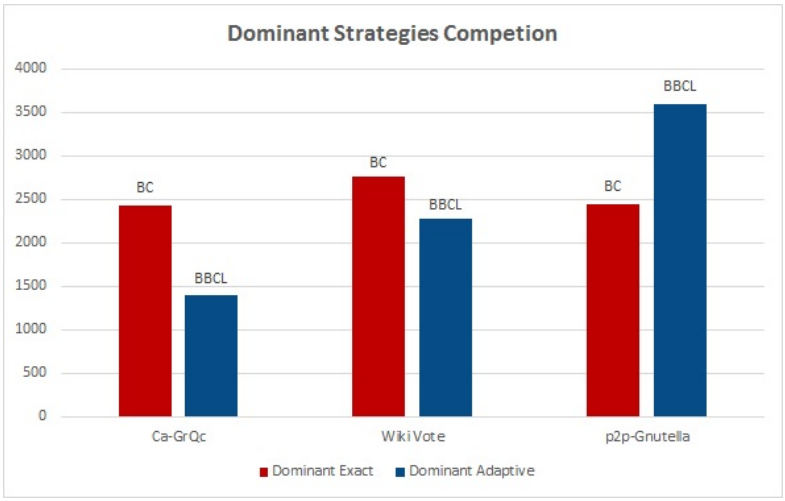


Figure 5.62: Exact Dominant VS Adaptive Dominant Competition in small networks

in each case. This is very interesting, since the exact measures are supposed to be the best strategies in extracting good nodes in a network. Thus, there should be some other factors that influence the outcome of the competitions.

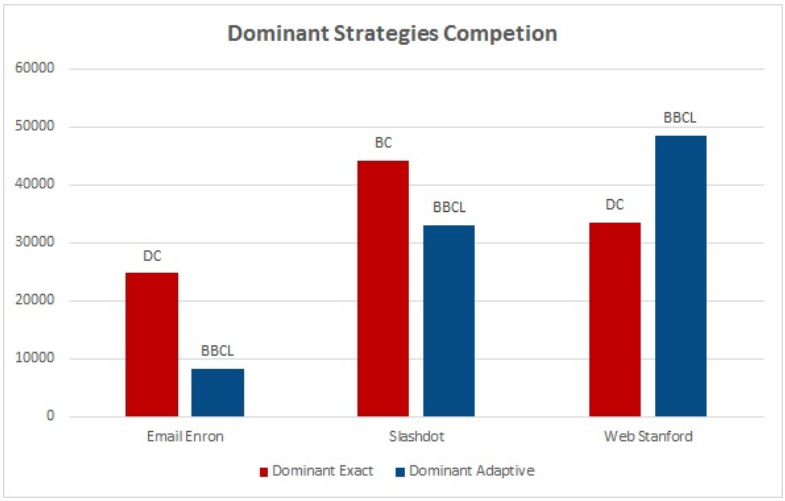


Figure 5.63: Exact Dominant VS Adaptive Dominant Competition in large networks

Chapter 6

Conclusions

The purpose of this thesis was to examine if there is a way for a firm to choose some important nodes in a network, and based on them, to start a cascade process that can reach the entire network. Furthermore, the strategy that is chosen should prevent other firms which have the same goal: to win any other competitor and spread its own product.

Among the strategies that were used in this research, we managed to find a decreasing order for them, according to the diffusion levels that they achieve in different networks. The outcome of the experiments is that two of the importance-assessment strategies can be considered equal and these are Betweeness and Degree Centrality. Both of them can influence a large part of the network even when they compete with each other.

Degree Centrality has good results, especially when we use the Threshold model as an Information-Propagation Model. This is very reasonable since these two tactics help each other. Degree Centrality picks nodes that have many neighbors and the Threshold model operates counting the active neighbors of the nodes. Furthermore, it is known that nodes with high degree in a network are quite influential to others.

Regarding the approximate and adaptive methods that we examined, one measure manages to be the dominant one in almost every competition. BBCL centrality wins even the most difficult opponents. It extracts very quickly the influential nodes of a network, regardless of its topology, and wins, in some of them, even the exact measures. This information is very important if we can prove why is this happening. If we can find the factors that BBCL take advantage and achieves the victories, we could use this information to choose the best strategy depending on the network.

An interesting observation is that BBCL dominates over the exact measures when they compete in a graph with low number of triangles. The number of triangles affects the strategies which count shortest paths and as a result they select less important agents. This observation could be expand by analyzing and proving why the triangles have that influence in this category of measures.

Finally, we recommend as future work, to examine the competition of more importance-assessment strategies, using the Threshold model. Furthermore, one can experiment in much larger graphs and see the diffusion levels in these cases.

Bibliography

[1] Freeman, L. C. A set of measures of centrality based on betweenness. Sociometry, 40:35-41, 1977.

[2] Brandes, U.: A faster algorithm for betweenness centrality. J. Mathematical Sociology, 25(2):163-177, 2001.

[3] A. E. Sariyüce, K. Kaya, E. Saule, and Ümit V. Catalyürek. Incremental algorithms for network management and analysis based on closeness centrality. CoRR, abs/1303.0422, 2013.

[4] D. A. Bader, S. Kintali, K. Madduri, and M. Mihail. Approximating betweenness centrality. In WAW, pages 124-137, Springer-Verlag, 2007.

[5] C. Borgs, M. Brautbar, J. Chayes, and B. Lucier. Maximizing social influence in nearly optimal time. In SODA, 2014.

[6] Morris H DeGroot. Reaching a consensus. Journal of the American Statistical Association, 69(345):118b 121, 1974.

[7] Eppstein, D., Wang, J.: Fast approximation of centrality. In: Proc. 12th Ann. Symp. Discrete Algorithms (SODA-01), Washington DC, 228-229, 2001.

[8] Banerrjee A., Chandrasekhar A., Duflo E., Jackson M. Gossip:identifying central individuals in a social network, 2014

[9] Jackson, M. Social and Economic Networks, Princeton: Princeton University Press, 2008.

[10] Ghaderi J, Srikant R. Opinon Dynamics in Social Networks: A Local Interaction Game with Stubborn Agents, American Control Conference (ACC), Washington DC, 2013.

[11] Yoshida Y. Almost Linear-Time Algorithms for Adaptive Betweenness Centrality using Hypergraph Sketches, KDD, New York, 2014

[12] Bavelas A., Communication patterns in task oriented groups, Journal of the Acoustical Society of America, 22:271-282, 1950.

[13] M. A. Beauchamp. An improved index of centrality, Behavioral Science, 10:161-163, 1965.

[14] Everett, M. G., Borgatti, S. P., Extending centrality, P. J. Carrington, J. Scott, S. Wasserman (Eds.), Models and methods in social network analysis (pp. 57-76), New York, Cambridge University Press, 2005.

[15] Zhao J., Lui J.C.S., Towsley D., Guan X., Measuring and Maximizing Group Closeness Centrality over Disk-Resident Graphs, WWWb 14 Companion, Seul, Korea, 2014.

[16] Hoeffding, W., Probability inequalities for the sum of bounded random variables, Journal of the American Statisitcal Association, 58:13-30, 1963.

[17] R. Albert, H. Jeong and A.-L. Barabasi: Diameter of the world-wide web, Nature 401 (1999), 130b 131.

[18] Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence in a social network. In: Proc. 9th Intl. Conf. on Knowledge Discovery and Data Mining, 2003.

[19] Nemhauser, G., Wolsey, L., Fisher, M.: An analysis of the approximations for maximizing submodular set functions. Mathematical Programming 14 (1978) 265b 294, 137b 146

[20] U. Brandes, C. Pich, Centrality estimation in large networks. Intl. Journal of Bifurcation and Chaos, Special Issue on Complex Networks' Structure and Dynamics, 2007.

[21] Kempe, D. and Kleinberg, J. and Tardos, E., Influential Nodes in a Diffusion Model for Social Networks, 32nd International Colloquium on Automata, Languages and Programming ICALP, 2005.

[22] U. Brandes, On Variants of Shortest-Path Betweenness Centrality and their Generic Computation, Social Networks., pp. 136b 145, 2008.

[23] P. Bonacich, Power and Centrality: A Family of Measures, American Journal of Sociology, Vol. 92, No. 5, pp. 1170-1182, 1987.

[24] P. Bonacich, P Lloyd, Eigenvector-like measures of centrality for asymmetric relations, Social networks 23 (3), 191-201, 2001.

[25] B. Golub, M.O. Jackson, Naive learning in Social Networks and the Wisdom of Crowds, American Economic Journal: Microeconomics, pp. 112-149, 2010.

[26] P.M.DeMarzo, D.Vayanos, J.Zwiebel, Persuasion Bias, Social Influence and Unidimensional Opinion, Quarterly Journal of Economics, pp. 909 - 968, 2003.

[27] M. Granovetter. Threshold models of collective behavior. American Journal of Sociology 83(6):1420-1443, 1978.

[28] J. Coleman, H. Menzel, E. Katz. Medical Innovations: A Diffusion Study Bobbs Merrill, 1966.

[29] E. Rogers. Diffusion of innovations Free Press, 1995.

[30] T. Valente. Network Models of the Diffusion of Innovations. Hampton Press, 1995.

[31] F. Bass. A new product growth model for consumer durables. Management Science 15(1969), 215-227.

[32] J. Brown, P. Reinegen. Social ties and word-of-mouth referral behavior. Journal of Consumer Research 14:3(1987), 350-362.

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

[34] J. Goldenberg, B. Libai, E. Muller. Talk of the Network: A Complex Systems Look at the Underlying Process of Word-of-Mouth. Marketing Letters 12:3(2001), 211-223.

[35] J. Goldenberg, B. Libai, E. Muller. Using Complex Systems Analysis to Advance Marketing Theory Development. Academy of Marketing Science Review 2001.

[36] V. Mahajan, E. Muller, F. Bass. New Product Diffusion Models in Marketing: A Review and Directions for Research. Journal of Marketing 54:1(1990) pp. 1-26.

[37] M. Richardson, P. Domingos. Mining Knowledge-Sharing Sites for Viral Marketing. Eighth Intl. Conf. on Knowledge Discovery and Data Mining, 2002.

[38] L. Blume. The Statistical Mechanics of Strategic Interaction.Games and Economic Behavior 5(1993), 387-424.

[39] G. Ellison. Learning, Local Interaction, and Coordination.Econometrica 61:5(1993), 1047-1071.

[40] S. Morris. Contagion. Review of Economic Studies 67(2000).

[41] H. Peyton Young. The Diffusion of Innovations in Social Networks. Santa Fe Institute Working Paper 02-04-018(2002).

[42] H. Peyton Young. Individual Strategy and Social Structure: An Evolutionary Theory of Institutions. Princeton, 1998.

[43] Ulrik Brandes. On Variants of Shortest-Path Betweeness Centrality and their Generic Computation, 2007

[44] Robert Geisberger. Better Approximation of Betweeness Centrality, 2008

[45] M. Riondato, E. M. Kornaropoulos. Fast Approximation of Betweeness Centrality through Sampling

[46] M. H. Chehreghani. An Efficient Algorithm for Approximate Betweeness Centrality Computation

[47] E. Bergamini, H. Meyerhenke. Fully-Dynamic Approximation of Betweeness Centrality, 2015

[48] Lipton, R., Naughton, J. Estimating the size of generalized transitive closures. In: VLDB. (1989) 165b 171