INFLUENCE MAXIMIZATION BASED ON PARTIAL NETWORK STRUCTURE INFORMATION: A COMPARATIVE ANALYSIS ON SEED SELECTION HEURISTICS

by

Şirag Erkol

B.S., Industrial Engineering, Boğaziçi University, 2014

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering
Boğaziçi University

2016

ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to *Gönenç Yücel*, my thesis supervisor, for his immense help throughout this study. Whenever I got stuck at some point, which happened frequently, he guided me to clear the path in the process with his true wisdom and experience in modeling. I am grateful for all his support and help both as my thesis and graduation project supervisor, and for introducing me to the world of agent-based modeling.

I would also like to thank *Prof. Yaman Barlas* for all his help and everything he has taught me during my studentship and assistantship. He generously shared his extensive knowledge and widened my perspective, which will be an invaluable asset even if I learned the least I could from him.

I would like to thank *Assist. Prof. Evren Güney* for his help and comments on the study. I also would like to thank *Prof. Kuban Altınel* for taking part in my thesis committee.

Last but not least, I would like to thank all *SESDYN* members, especially *Oylum Şeker* for answering my endless questions on every little detail of everything with her vast knowledge and enthusiasm. Her companionship has surely been a joy in these last two years.

ABSTRACT

INFLUENCE MAXIMIZATION BASED ON PARTIAL NETWORK STRUCTURE INFORMATION: A COMPARATIVE ANALYSIS ON SEED SELECTION HEURISTICS

In this study, the problem of seed selection is investigated. This problem is mainly treated as an optimization problem, which is proved to be NP-hard. There are several heuristic approaches in the literature which mostly use algorithmic heuristics. These approaches mainly focus on the trade-off between computational complexity and accuracy. Although the accuracy of algorithmic heuristics are high, they also have a high computational complexity. Furthermore, in the literature it is generally assumed that complete information on the structure and features of a network is available, which is not the case in most of the times. For the study, a simulation model is constructed, which is capable of creating networks, performing seed selection heuristics, and simulating diffusion models. Novel metric-based seed selection heuristics that rely on partial information are proposed and tested using the simulation model. These heuristics use local information available from nodes in the synthetically created networks. The performances of heuristics are comparatively analyzed on three different networks and two different diffusion models, i.e. six combinations. The results suggest that the performance of a heuristic depends on the structure of a network. A heuristic to be used should be selected after investigating the properties of the network at hand. Also, the approach of partial information provided promising results. It has approximated to the performances of heuristics relying on complete information in most of the cases.

ÖZET

AĞ YAPISI ÜZERİNDE KISMİ BİLGİ KULLANARAK ETKİ ENİYİLEMESİ: ÇEKİRDEK KÜME SEÇİMİ SEZGİSELLERİ ÜZERİNE KARŞILAŞTIRMALI BİR ÇÖZÜMLEME

Bu çalışmada çekirdek küme seçimi problemi incelendi. Bu probleme temel olarak eniyileme yaklaşımı uygulanmıştır ve bu yaklaşımın NP-zor olduğu kanıtlanmıştır. Birçok kaynak bu probleme algoritmik sezgisellerle yaklaşmıştır. Bu yaklaşımlardaki temel amaç hesaplama karışıklığını azaltırken çözümlerin hata payının artışını en düşük seviyede tutmaktır. Algoritmik sezgisellerin hata payı düşük, hesaplama karmaşıklıkları yüksektir. Ayrıca kaynakların çoğunda ağların yapısı ve özellikleri hakkında tüm bilgiye sahip olunduğu varsayılmıştır ancak durum genelde bu şekilde değildir. Bu çalışma için bir benzetim modeli kuruldu. Kurulan model, ağları oluşturma, çekirdek küme seçimini yapabilme ve yayılım modellerinin benzetimini gerçekleştirebilmektedir. Ağ üzerine kısmi bilgi kullanan yeni ve ölçüt temelli sezgiseller oluşturuldu ve benzetim modeli kullanılarak test edildi. Bu sezgiseller, sentetik olarak üretilmiş ağlardaki elemanları ve onlarda var olan yerel bilgileri kullandı. Sezgisellerin performansları üç farklı ağ tipi ve iki farklı yayılım modeli, yani altı kombinasyon üzerinde karşılaştırmalı olarak analiz edildi. Sonuçlar, sezgisellerin performanslarının ağ yapılarına bağlı olduğunu göstermektedir. Bir ağ üzerinde kullanılacak sezgisel, ağın özellikleri incelendikten sonra seçilmelidir. Ayrıca, kısmi bilgi yaklaşımıyla umut verici sonuçlar elde edildi. Kısmi bilgi kullanan sezgiseller, birçok yerde tam bilgi kullanan sezgisellerin performanslarına yakınsadılar.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

AT	Average Threshold
B	Betweenness
C	Closeness
D	Degree
DD	Degree Discount
E	Eigenvector
IC	Independent Cascade
LT	Linear Threshold
PR	PageRank
R	Random
S	Strength
s1	over s Nodes within 1 Step
TS	Twostep
w1S	within 1 Step
w2S	within 2 Steps
w5Gr	with 5 Groups
wD	with Degree Effect
wSD	with Square Root Degree Effect

1. INTRODUCTION

Social networks constitute an important part of people's lives nowadays. They have many different types that are being used by people to stay connected with each other and with the world, such as Twitter and Facebook. People exchange ideas, demonstrate tastes, follow the news, and show their feelings through these social networks. This fact creates a virtual environment in which information and ideas disseminate.

One example of such social networks is the microblogging site Twitter, where people can actively share their opinions on any subject, or can simply follow other people to see what they think. This nature of Twitter helped it become a platform for information sharing, like a news media [1]. Lately, there have been important incidents in Turkey [2] and Chile [3] where the information propagation through Twitter has been crucial to the creation and attendance of protests and public knowledge. People and news media have all had different roles in these movements regarding their level of connectedness, or connecting different clusters of people by being bridges between these and enabling information flow from one to the other [3]. This shows the possible differences between the elements of networks, and how they can be essential in different ways for the network.

These characteristics of network components become important in the case of a diffusion process. By diffusion process, the propagation of any kind of information in a network is meant. This process can be about a marketing campaign, a social movement, or anything else that one can imagine its spread through people. As it is for any kind of information, there will be people who would be the first batch adopting and spreading this information. This brings the issue of who should be these people, and more importantly, whether it matters who is selected to start the dissemination.

Seed selection is the study of dealing with such questions. Seeds are defined as the first batch of people who are 'injected' with the information that is intended to spread

through a network. The problem of seed selection is the decision of who the people in the first batch will be. In a world where all people have similar characteristics and connectedness in a network, seed selection would be a struggle in vain, since obviously any selection would result with almost the same amount of propagation. However, in real life, this is not the case as explained previously. Each and every person in a network has different characteristics with a different connectedness feature, such as the number of people they know or different types of clusters of people they connect, which makes the study of seed selection essential. Practically, the goal of seed selection is to maximize the total gain at the end of diffusion process with a given seed set size.

The problem of seed selection is first defined by Domingos and Richardson [4], but its formulation as a discrete optimization problem was first done by Kempe et. al [5]. It is primarily used in areas such as viral marketing [6], vaccination strategies [7], and (mis)information diffusion [8]. The ultimate goal for all such areas is the same, ending up with the greatest diffusion possible via selecting seeds intelligently. However, there are several issues that come along with the problem of seed selection.

First and most importantly, to make the best seed selection possible, it is necessary to know the whole structure of the network, such as the nodes and links between them, and the characteristics of the components and links. These characteristics, which will be explained in depth later, are closeness of a friendship and proneness of a person to acquiring new ideas. To know the whole information on a network is very hard to accomplish, because the information contained in a network is huge and more critically it is very unlikely to possess it in perfect form. As an example, Twitter has approximately 310 million monthly active users with an average of approximately 200 followers, which creates a network of 310 million nodes with 62 billion links [9]. Even if someone manages to collect all the information and have a computational power that can handle solutions on such a network, there will be a problem with quantifying the qualitative characteristics, such as the strength of links in a network. Secondly, the number of combinations that can form the seed set grows exponentially as the network size increases, and trying to solve this combinatorial problem to see what will be the optimal seed selection will require too much time and computational power. These

problems that have been mentioned brings up the following questions: Is it possible to identify efficiently good seed sets on a network of meaningful size, based on partial information? If the answer is yes, what kind of algorithms should be used for this purpose?

2. LITERATURE REVIEW

2.1. Network Types

The most important distinctions between networks are evaluated using two metrics, clustering coefficient and average path length [10]. The clustering coefficient of node x is defined as

$$CC(x) = \frac{L(x)}{\frac{d_x(d_x-1)}{2}} = \frac{2L(x)}{d_x(d_x-1)} \tag{2.1}$$

where $L(x)$ is the number of links between the neighbors of node x, and d_x is the degree of node x. The clustering coefficient of a network N is

$$CC(N) = \frac{1}{n}\sum_{x=1}^{n} CC(x) \tag{2.2}$$

where n is the number of nodes in the network. This metric shows how clustered is a network, as the value gets higher, it is more possible for the neighbors of a node to know each other. Similarly, the average path length of node x is defined as

$$APL(x) = \frac{1}{n-1}\sum_{y} d_{xy} \tag{2.3}$$

where d_{xy} is the shortest path between node x and any other node y. The average path length of a network N is

$$APL(N) = \frac{1}{n}\sum_{x=1}^{n} APL(x) \tag{2.4}$$

This metric indicates how many steps is necessary on the average to reach a node from another one.

There are mainly three network topologies mentioned and used widely in the literature: random, small-world and scale-free networks.

The random networks are characterized by their low clustering coefficient and low average path length [11, 12]. The algorithm for the construction of this network was first proposed by Erdős-Rényi [13]. In a network with n nodes, there are $C(n, 2)$ possibilities of links in case of an undirected network, and twice this number when the network is directed. The Erdős-Rényi method uses a probability p, which indicates the existence possibility of a link in a network. The probability of existence for a link is independent from the existence of other links. Using such a probability, the expected number of links for a network will be $p \times C(n, 2)$ for an undirected network, and again twice this number for a directed network.

Another network type is the small-world network. The idea of a small-world dates back to the experiment conducted by Milgram [14]. In his experiment, Milgram initiated a letter chain where the people getting the mails were asked to send these to one of the two targets, and if they did not know these targets personally, they had to send these mails to one of their friends whom they thought was most likely to know one of these two targets. At the end of the experiment, using the paths that have finalized by the mail reaching the target, Milgram estimated that median number of steps necessary was six, and this gave rise to the popular phrase of 'six degrees of separation' [15, 16]. This finding states that there is an average of six steps between any two people in the world brought up the name of small-world for such networks, which are mainly characterized by their high clustering coefficient and low average path length [10]. There are several algorithms for creating a small-world network, such as the one by Watts and Strogatz [10], and another by Kleinberg [17]. The method that Watts and Strogatz had come up is as follows: They start with a ring lattice network. This lattice has characteristics of high clustering coefficient and high average path length. After the network is initiated, they rewire a certain portion p of the links. When p=0 the lattice remains the same, and when $p = 1$ it becomes a random network. In this range they look for a certain fragment of values that will create a small-world network from a ring lattice. In their example, where they have used a network with

1000 nodes and an average degree of 10, a p value between 0.005 and 0.15 creates a small-world network. On the other hand, Kleinberg uses a grid-based algorithm, and different from the previous method, unidirectional networks can also be created using this algorithm.

The last type of network is the scale-free network, also known as the preferential attachment network. The characteristics of a scale-free network is its power-law scaling, the low clustering coefficient (yet higher than random graphs) and low average path length [18]. The power-law scaling is defined as the functional relationship between two variables, where one quantity changes as a power of the other. The microblogging site Twitter can be a good example for scale-free networks, in which there are some people acting as hubs with very high degrees, and most users with lower degrees. In Figure 2.1 the degree distribution of a real world network can be seen [19]. k represents the degree of nodes in the network, and p_k represents the portion of nodes that have a degree of k. As it can be seen, there are a lot of nodes with lower degrees and only some with very high degrees. This property of the degree distribution in the graph shows that it follows a power-law scaling.

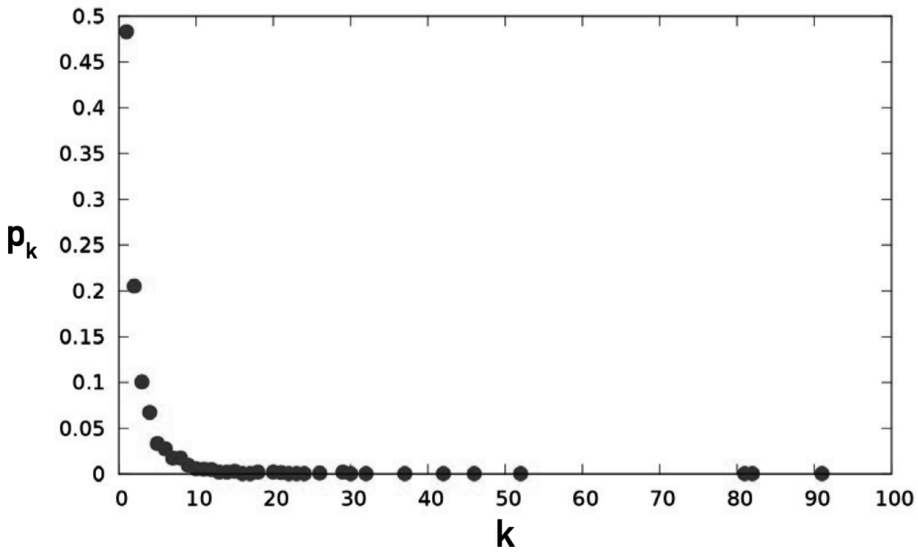

Figure 2.1. Power-Law Scaling of Degree Distribution of a Real World Network

The idea of preferential attachment was first introduced by Price, which he first called 'cumulative advantage' [20]. Later, Barabási and Albert came up with a method, which is now called the Barábasi-Albert model, that creates a scale-free network and follows the power-law scaling observed in real networks [21]. This model starts with m nodes, and continues adding nodes by connecting to m-1 already existing nodes, and this process continues until the necessary number of nodes is reached. The important point here is that, the new-coming nodes are always biased towards the nodes with higher degrees. Let the degree of node x be d_x and the probability of a new-coming node being attached to it be p_x, if $d_x > d_y$ then $p_x > p_y$.

2.2. Diffusion Models

There are mainly two diffusion models that are most widely used in the literature, which are the Linear Threshold (LT) Model and the Independent Cascade (IC) Model.

```
for each inactive node x do
   set s_x ← sum of link strengths with active neighbors
   if s_x > t_x then
      activate node x
   end if
end for
if no node is activated at t then
   stop
end if
```

Figure 2.2. Pseudocode of the Linear Threshold Model

One of the two diffusion processes that have been used in this study is the Linear Threshold Model. The idea of a threshold model was first proposed by Granovetter and Schelling [22,23]. However, a generalized model was first proposed by Kempe, Kleinberg and Tardos [5]. In the model proposed by Kempe *et al.*, each node is influenced by its neighbor y depending on a weight w_{xy} such that $\sum_y w_{xy} < 1$. This weight can be

interpreted as the effect of a node on its neighbor or vice versa. Another characteristics of this model is the threshold values, t_x, which are selected randomly from a uniform distribution in the interval [0,1] in the model by Kempe *et al.* Some other approaches use the value of 0.5 for all the nodes as the threshold value, making the nodes behave in a majority rule way when the link strengths are disregarded [24, 25]. In the LT model, an inactive node becomes active when $\sum_y w_{xy} > t_x$, where node y is an active neighbor of node x. Activeness of a node indicates that the node has absorbed what diffuses in the network. The pseudocode for the LT model can be seen in Figure 2.2.

```
for each activated node x at t − 1 do
  for each inactive neighbor y do
    set n ← a random uniform number in range [0,1]
    if n > w_xy then
      activate node y
    end if
  end for
end for
if no node is activated at t then
  stop
end if
```

Figure 2.3. Pseudocode of the Independent Cascade Model

The second diffusion process used is the Independent Cascade Model. It was first proposed and studied by Goldenberg, Libai and Muller [26,27]. Similar to LT, there are weights assigned for links, but no thresholds for nodes are assumed. The weights are categorized as strong and weak for certain ranges in the model proposed by Goldenberg *et al.*, Alternatively, they can be used as numeric values straight away in the interval of [0,1]. These weights work as probabilities, once a node becomes active, it triggers a propagation towards all its inactive neighbors. When a node x becomes active, it tries to activate all its inactive neighbors, and succeeds with probability w_{xy}, where w_{xy} is the link strength between node x and node y. If node y becomes active, it repeats the

same procedure and this continues until all the active nodes have tried to diffuse to their neighbors. The pseudocode for the IC model can be seen in Figure 2.3.

The mentioned network types and diffusion models are the ones that are most widely used in the literature as mentioned previously. So, in this study, random, small-world, and scale-free networks will be used along with the diffusion models Linear Threshold and Independent Cascade.

2.3. Solution Approaches

Kempe et. al [5] are the first to investigate the algorithmic and computational perspective of the seed selection problem. They proved the problem of optimal seed selection is NP-hard under both the Linear Threshold Model and the Independent Cascade Model. They came up with a greedy algorithm that achieves an approximation guarantee of 1-1/e, which is approximately 63%. A pseudocode for this algorithm can be seen in Figure 2.4. [28]

start with A = $\emptyset$
for $i = 0$ to K **do**
  let v_i be a node that maximizes the marginal gain $\sigma(A \cup \{v\}) - \sigma(A)$
  set $A \leftarrow A \cup \{v\}$
end for

Figure 2.4. Pseudocode of Greedy Algorithm

Later, several papers have been published both improving on the greedy algorithm, being the conventional method, and suggesting new approaches. Leskovec *et al.* [29] developed an efficient algorithm, using the idea of sensor placement for contaminant detection and blogs to read for not missing important stories, and adapted these questions to seed selection problem. Their algorithm, named *CELF*, can be scaled to large problems, and it is reported to performs 700 times faster than the greedy algorithm [5].

Goyal *et al.* [30] came up with the an improvement on the algorithm of Leskovec *et al.* [29], named *CELF++*, by avoiding unnecessary computations existing in the prior algorithm. The results indicate an improvement of 35-55% over *CELF* in the efficiency. The same authors also came up with another algorithm under the LT model, and reported it outperforms the greedy algorithm measured by the running time and the memory consumption [31].

Kimura *et al.* [32] improved on the computation time of the greedy algorithm by approximations based on bond percolations and graph theory. Using both LT and IC, it is reported that they have computed results 4600 times faster for the prior and 1800 faster for the latter diffusion model compared to the conventional method on a large scale network.

An approach using simulated annealing was proposed by Jiang *et al.* [33], being the first of its kind. The reported results suggest that the proposed method over-performs the conventional method in terms of computation time while also improving the accuracy by an average value of 5%.

Another improvement on the greedy algorithm is proposed by Estevez *et al.* [28], named *set covering greedy algorithm*. They use the fact that while selecting nodes, the neighborhoods of the seeds should better not overlap, so that the selected seeds can cover a larger area, which is a case ignored in the widely used degree centrality heuristic. The reports of their findings indicate that the proposed algorithm works very fast, outperforms the greedy algorithm and degree centrality heuristic, and can also be used in large scale networks.

Cheng *et al.* [34] proposed another algorithm building on simple greedy algorithm, focusing on the scalability-accuracy issue. Pointing out that greedy algorithms come with a heavy computational load although they have high accuracy, and heuristics lack accuracy while being computationally fast, their results suggest that their algorithm dramatically reduced the computation without loss of accuracy.

Narayanam *et al.* [35] have come up with an algorithm that approaches the problem from two different perspectives. One is the generally known seed selection problem and the second is the coverage problem, where the goal is to reach a certain diffusion level using the minimum number of seeds instead of maximizing the diffusion using a certain number of seeds. They have reported their findings to be more efficicent and computationally cheaper comparing with to conventional methods.

For some special cases of IC, Kimura *et al.* [36] proposed a method that can efficiently compute estimates. In these special cases, nodes can become active only at a certain step or steps. When the propagation probabilities through links are small, they have shown that their proposed method can come up with good estimates for influential nodes.

Kitsak *et al.* [37] proposed the *k-shell* decomposition analysis, a similar approach to degree centrality heuristic. The method starts with removing nodes with only one link and assigning them to the group 1-shell, and then recursively assigns nodes with 2 links to 2-shell. This process continues until no nodes remain, and the seeds are selected from shells with greater k values. This is similar to degree centrality heuristic since it uses the notion of degrees, but diverges from it since it looks further then one step in the network. Their findings compromise with the proposed theory that the most influential nodes lie in the core of the network which are grouped in shells with greater k values.

Zhang *et al.* [38] came up with a novel method of seed selection, claiming that conventional methods all have their limitations, such as the *k-shell* method, which can only be applied to undirected and unweighted networks. The proposed method, named *k-medoid*, uses the propagation probability between all pairs of nodes and k-medoid clustering algorithm. Its advantage mainly occurs in networks with a community structure, since it takes into account all the communities existing in the network, while the conventional methods, such as the simple greedy algorithm, may tend to select seeds within the same community due to larger size or higher density.

Chen *et al.* [39, 40] proposed an improvement on the simple greedy heuristic for IC model. The results reported indicate that they have achieved better running time, and in most cases a 100-270% better result in accuracy than existing methods. Their algorithm is able to scale beyond million-sized networks. Also, they have enabled a feature to decide on the trade-off between running time and accuracy, thus being able to manipulate these for each case.

Besides the algorithmic approaches mostly based on the simple greedy heuristic, there are other heuristic approaches that simply use characteristics of nodes in a graph, called the centrality measures. These measures are used to find the important nodes in a network, such as nodes connecting different groups of people or nodes very close on average to all the others. Some of these are degree centrality [5], betweenness centrality [41, 42], closeness centrality [43], eigenvector centrality [44], and Google's famous PageRank [45].

The degree centrality is simply the measure of the number of neighbors a node has. The betweenness centrality of a node x is defined as

$$C_B(x) = \sum_{x \neq y \neq z} \frac{\sigma_{yz}(x)}{\sigma_{yz}} \tag{2.5}$$

where σ_{yz} is the total number of shortest paths between nodes y and z and $\sigma_{yz}(x)$ is the number of these paths that pass through node x.

Closeness centrality indicates how close a node is on average to the other components of the network. This metric for a node x is measured as

$$C_C(x) = \frac{1}{\sum_y d_{xy}} \tag{2.6}$$

where d_{xy} is the shortest path between node x and any other node y.

Eigenvector centrality is a measure to decide how well a node is connected to a network, also taking into account the connectedness of its neighbors. It is defined as

$$C_E(x) = \frac{1}{\lambda} \sum_y a_{xy} C_E(y) \tag{2.7}$$

where a_{xy} is the value from the adjacency matrix of the network, thus it is equal to 1 if x and y are neighbors and 0 if not, and λ is some predefined constant.

In this sense, PageRank follows a similar logic of calculation. It can be thought of the time someone would spend on a node, given the network. While at a node, all neighbors have an equal chance of being visited at the next time step, but there is also a 15% chance of a random step, going to a node not necessarily connected to the current one. It is defined as

$$PR(x) = \frac{1-\alpha}{n} + \alpha \sum_{y \in N(x)} \frac{PR(y)}{d_y} \tag{2.8}$$

where n is the number of nodes in the network, d_y is the degree of node y, $N(x)$ is the neighborhood of node x, and α is the predefined constant value of 0.85. This idea of PageRank calculation has stemmed from the internet usage of people. Using a website, someone can either click a link on the website or can type another URL and go to another website, which is the random step in PageRank [45].

Comparing these centrality-based metrics, degree-centrality is a very simple and computationally cheap metric but has a poorer performance as a heuristic than the others since it remains local, regarding the calculation of the metric, whereas the others are more global, meaning the whole network is, directly or indirectly, taken into consideration while calculating these metrics, but have higher computational complexity, especially PageRank, making them harder to use for larger networks [46].

Chen et. al [46] approached these centrality-based heuristics in a combinatorial manner. They proposed a local centrality measure with a trade-off between degree-

centrality and the other computationally complex centrality measures (betweenness, closeness, etc.). The results indicate that the proposed measure performs as good as closeness centrality while having a much lower computational complexity, and performs much better than the degree and betweenness centrality.

There have also been other heuristic approaches proposed, similar to centrality heuristics. Chen *et al.* [39] proposed the *degree discount heuristic*, which works similar to degree-centrality. The difference between the two is that already activated neighbors are not included in the calculation while selecting a node in degree discount. For example, if node x has n neighbors of which k is active, then its degree is n but the measure for degree discount will be n-k. The results of the paper suggest that the proposed heuristic performs no more than 3.5% worse compared to any greedy algorithm while being almost a million times faster. The authors conclude that although heuristic methods have always been thought to be outperformed by greedy approximation algorithm, such a result will shed new light on the research of heuristic algorithms.

An extension to the degree-centrality heuristic is proposed by Stonedahl *et al.* [47]. Named *twostep*, this heuristic takes into account not only the immediate neighbors but also the neighbors of neighbors, thus calculating a measure showing the number of nodes that can be reached in within two steps. Using a genetic algorithm and combining several heuristic methods, they have achieved good results, while also showing that degree turns out to be a very important component for all types of networks, especially those with uneven degree distribution.

As it can be seen from all the literature referenced above, seed selection is predominantly treated as an optimization problem, namely influence maximization problem. The problem of finding the set of nodes that would yield the maximum influence spread is indeed an optimization problem, but it requires perfect information on the structure of the network being studied. Even if the complete structure of a network is known, solving the aforementioned optimization problem is notoriously difficult, as it is shown to be a NP-hard problem [5].

In this study, we will look for simple seed selection heuristics that do not rely on the assumption of complete information about the structure of a network. The benchmarks for the proposed heuristics will be random seed selection and some other heuristics that are known to have good performances, and the aim will be to perform significantly better than random seed selection, while trying to outperform the other heuristics, and meanwhile be computationally cheap. The trade-off between accuracy and computational complexity has been the main focus of the researchers while assuming the existence of perfect information. Here, we will try to eliminate the need for this assumption through several aspects by starting with a random subset of nodes and proceeding based on local information, meanwhile keeping the computational complexity and accuracy of heuristics at an acceptable level. The procedure of this study is to analyze heuristics, try to understand the reason behind the better performing ones under different circumstances, and come up with efficient heuristics that require only partial information.

3. RESEARCH OBJECTIVES AND METHODOLOGY

In this study, we will be looking for metric-based heuristics rather than algorithmic ones such as the simple greedy heuristics, which covers the most of the literature. The metric-based heuristics calculate a metric provided for each node and select n nodes with the best measure. The goal will be to come up with good enough heuristics that will use local information gathered from nodes instead of assuming all information is globally available and also be efficient in terms of computational complexity. The aim of the study is two fold, using partial information on the networks and having heuristics computationally fast.

Heuristics will be comparatively analyzed in the study. For the analysis, experimentations will be conducted for each combination of network type and diffusion model. The results for each heuristic in these combinations will be averaged. The comparative analysis will be carried out through the averages for each of the combinations separately. So, the analysis will be done in six sections for the six combinations.

There are three main steps of the study, constructing the network, selecting the seeds, and simulating the diffusion process (as it can be seen in Figure 3.1). Agent-based modeling platform is used for all the steps mentioned due to the nature of the problem. As the agent-based modeling software NetLogo [48] is used, which is capable of processing all the necessary steps.

Figure 3.1. Procedure Flow Chart of the Model

All the steps are executed in NetLogo. The procedures are explained below in this chapter, except for the seed selection, which will be explained in Chapter 4.

The agent-based model is able to initiate three different types of networks (random, small-world, scale-free) and simulate two different types of diffusion models (Linear Threshold, Independent Cascade). The NetLogo environment is used to specify which network will be used with which diffusion model, by also specifying the necessary parameters. There are two types of parameters that need to be decided for the agent-based model, one is the set of parameters necessary to construct a network, and the second is the set necessary to simulate the diffusion models.

3.1. Network Construction

For the network construction, there are four parameters. These parameters are the number of nodes, the probability of existence for links in random networks, initial number of neighbors for small-world and scale-free networks, and the rewiring probability for small-world networks.

The number of nodes n have been decided as 1000. The reason behind this is that when n is lower than 1000, the outcomes of simulations change, which may result in non-accurate or different results than of those of large-scale networks. Besides, when n is greater than 1000, the results do not change but the necessary computational time for the simulation increases. Due to these drawbacks for both lower and greater values, n has been used as 1000 in the study, so that it will reflect the dynamics of the networks as designated and will not have a high cost of computation.

To initiate a random network, the algorithm of Erdős-Rényi is executed [13]. For this algorithm, the NW extension of NetLogo is used [48]. Here, it is necessary to specify the probability of existence for links, besides the number of nodes. The probability p of existence for links, gives the proportion of possible links expected to exist in the initialized network. In this study, it is decided to use a value of 5% for p, which makes the studied random networks sparse. Sparse networks have a low density,

which is defined as

$$\frac{2 \times (\#\ of\ links)}{n \times (n-1)} \tag{3.1}$$

Here n is the number of nodes in the network. Sparse graphs are intentionally selected to be studied because the social networks, which are the main medium of information diffusion, are in nature sparse [49].

Secondly, the small-world networks are created based on a network construction algorithm provided by Wilensky [50]. To initiate a small-world network it is necessary to specify the initial number of neighbors and the rewiring probability besides the number of nodes. In this model, the ring lattice method is used to create a network [10]. First, the given number of nodes are created and are linked such that they create a lattice ring. When initial number of neighbors is set to $2n$, each node connects itself to the closest n nodes, thus all end up having a degree of $2n$. After this initialization, each link is rewired with a probability, the rewiring probability. If a node is to be rewired, one of the nodes is kept, the link disappears and the node creates a new link to any other node, chosen randomly. At the end of this process, a small-world network will be obtained. The initial number of neighbors for the ring lattice is selected as 10, based on Watts and Strogatz's study [10]. The rewiring probability is also investigated in the same study. Watts and Strogatz have shown that a ring lattice with 1000 nodes and an average degree of 10 will become a small-world network by rewiring its link in the range of 0.5% to 15%. As a result, the rewiring probability in this study has been chosen as 10%.

Finally, the scale-free networks are also created based on a network construction algorithm provided by Wilensky [51]. To initiate a scale-free network the number of initial neighbors should also be specified other than the number of nodes. The number of initial connections means the links that a new coming node will create to already existing nodes, which is the method proposed by Barabási and Albert [21]. So, when this value is m, the NetLogo model will first create a complete graph with $m+1$ nodes. Then nodes will be added one by one until the specified number of nodes is reached, and

each newly added node will be linked to m nodes following the preferential attachment rule. The number of initial neighbors is set to 5 in this study in order to be able to come up with a network similar to social networks, such as Twitter.

3.2. Seed Selection

After the networks are initiated, the seed selection process needs to be executed. Here, it is necessary to decide on the parameter of the seed set size, and the selection criteria to be used. The percentage of seeds is the percentage of nodes that will be activated initially, so these will be the initial propagators in the network.

The decision on the seed set size s is done based on a value that works good enough for this study. By good enough, it is meant a value that creates a difference between the heuristic performances. If s is set too high, then all heuristics are expected to end up with a very high diffusion, and if it is set too low, then none or very little diffusion may occur. Both cases will not be helpful in understanding the better or worse performing heuristics, since a value is needed where heuristic performances differ. So, for the sake of distinguishing the heuristics, experiments are carried out, the results are observed, and the value for s has been set as 5%. This value gives the necessary difference between heuristics performances to distinguish one from another.

On the other hand, the heuristics will decide which nodes should be activated initially, using different approaches and metrics. The heuristics used in this study will be explained in Chapter 4.

3.3. Diffusion Models

After seed selection, the last step is simulating the diffusion over the network. There are two parameters necessary to simulate a diffusion model, which are the link strengths and node thresholds. For both diffusion models, a link strength is necessary. This strength is defined while creating the links. The node thresholds are only used for the Linear Threshold Model. Both these values are used in a running diffusion,

depending on the selected diffusion model. The LT model is also named as the pull-type diffusion, because the nodes themselves check whether to activate or not. At each time step, all non-active nodes observe their neighbors and check which are active and which are not. For some node i, let us define k_i as the sum of link strengths with its active neighbors, and t_i as its threshold. At each time step, node i checks if the inequality $k_i > t_i$ holds, and if it does the non-active node activates itself, otherwise it stays non-active. On the other hand, the Independent Cascade Model is also named as the push-type diffusion, because here the active nodes attempt to activate their non-active neighbors. As mentioned previously, only link strengths are used in this diffusion process. Once a node becomes active at time t, it tries to activate its non-active neighbors only at time $t+1$. Independent of the success or failure, this attempt is not repeated for a second time. It succeeds depending on a probability w_{xy}, which is the strength of the link between nodes x and y. For example, if some node x is activated at time t, its neighbor node y is non-active, and their link strength is w_{xy}, the probability that node y will be active at time $t+1$ with the attempt of node x is w_{xy}. Here, it is important to point out that this probability is only for the attempt of node x, since other neighbors of node y may be also activated at time t, and these nodes will also attempt to activate node y at time $t+1$ in case node x fails.

For both diffusion models it is assumed that an active node will never be inactive. Based on this assumption, the diffusion process terminates when the number of active nodes does not change from time t to $t+1$. This is because when the diffusion percentage stays the same for one time step, it can not change afterwards. This situation applies for both diffusion models.

In the IC model, the link strength indicates the probability of an active user activating its inactive neighbor. For the LT model, it indicates the portion of importance a node gives to its neighbor considering all its neighbors. The study focuses on equally weighted links, so equal link strengths are assigned to all neighbors of a node. This assumption implies that all neighbors of node x will have equal effects on it. This idea has stemmed from social networks. It is assumed that if a person chooses to follow (in the case of Twitter) or become friends (in the case of Facebook) with someone, the

information shared through that channel can be well assumed to be equally important to those shared by other friends of the person. For example, let node x have degree d, then the strength of its incoming links from its neighbors will be $1/d$. There are other approaches on the decision of link strengths in the literature, but the assumption for this study is setting the strength of all incoming links to a node equal. Here, it is important to note that the incoming link strength from a node to node x is not necessarily equal to the outgoing link strength from x to that node, since these strengths depend on the degree of each node. To be clear, let nodes x and y be neighbors, and let the strength of the link from node x to node y be w_{xy} and w_{yx} for the link from node y to node x. In this case it can be said that if $d_x \neq d_y$, then $w_{xy} \neq w_{yx}$. Also, if a node has an incoming link from a certain node, than it will also have an outgoing link to that node. The relations are symmetrical between nodes, but the strengths are not.

For the node thresholds, they indicate the level of persuasion necessary to activate a node. Let the degree of node i be d_i, and its threshold value be t_i. Since the incoming link strengths of a node are taken equal in this study, node i will become active when it has at least $d_i \times t_i$ active neighbors, otherwise it will stay inactive.

In the literature, the node thresholds are assigned in two ways. One approach is to generate these values from a uniform distribution in range [0,1] [5, 28], and the other is to set them all equal to 0.5 [24, 25]. These methods are approximations since not much information and formalization is available on their quantification [5]. In this study, the threshold values have been set in the way that they will help investigating the difference between heuristics. In some threshold ranges, it is not possible to distinguish heuristics from each other, either due to very low or very high diffusion, so the ranges that they can be distinguished are searched using simulations. To do this, the simulation process is repeated for different ranges of threshold values, and the performance of two heuristics, degree-centrality and random seed selection heuristics are compared. A range that has a significant difference between the two heuristics is chosen. Also, the range is chosen such that there would be a margin to improve for the heuristics to be proposed. For each network, the same process is repeated and three different threshold ranges are used for three different networks. The threshold values for nodes are assigned using a

uniform distribution over these ranges.

Looking for the range of threshold distributions, two values are used. The first one is named *lower threshold*, which indicates the lower bound for the uniform distribution of threshold values, and the second is named *threshold gap*, which indicates the difference between the upper and lower bounds. Several lower thresholds are experimented and it is observed a value of 0.1 works for all network types. The experiments are carried out with 50 replications for each range and heuristic, thus for each network type there are 1900 simulation runs. The following results are obtained using the lower threshold value of 0.1.

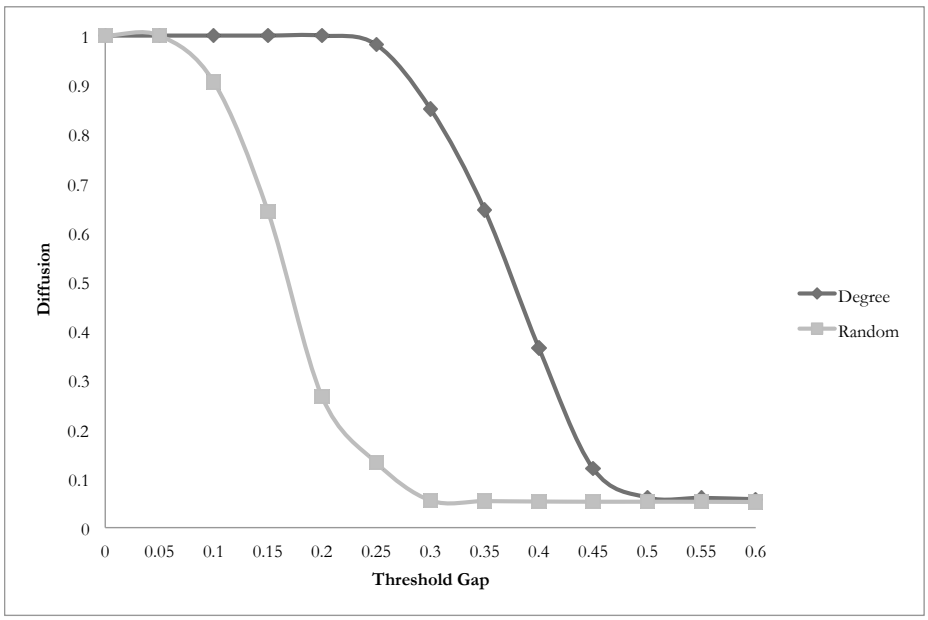

Figure 3.2. Random Network: Threshold Gap vs. Diffusion for Degree and Random Heuristics

For the random network, the lower bound is set to 0.1 and the gap is used as 0.35. So the node thresholds will be assigned from a uniform distribution of [0.1, 0.45]. It can be seen in Figure 3.2 that this range includes both a significant difference and a margin to improve for the heuristics to be proposed, as necessary. If the range for the distribution was set as [0.1, 0.3], the heuristics would still be distinguished, but there would not be a margin to improve for the newly proposed heuristics since the

degree-centrality heuristic already performs a full diffusion. If the range was set as [0.1, 0.7], it would not be possible to distinguish heuristics since both heuristics have very low diffusion in this range of distribution.

In the case of scale-free networks, the lower threshold is again set to 0.1 and the threshold gap to 0.9, so the uniform distribution will be in range [0.1, 1]. It can be seen in Figure 3.3 that both necessary conditions are satisfied. For example, a range such as [0.1, 0.8] does not provide the necessary margin to improve for the heuristics to be proposed.

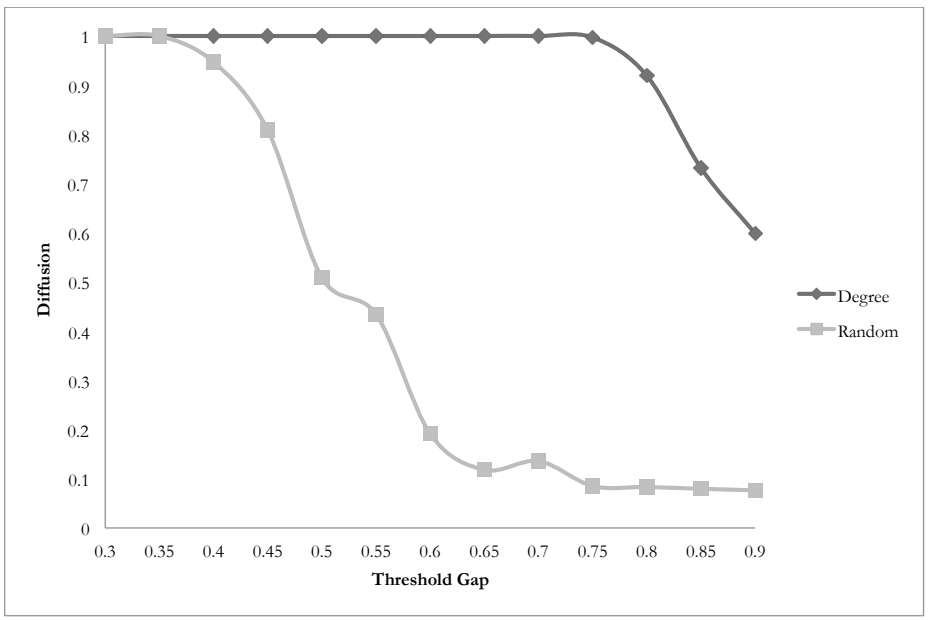

Figure 3.3. Scale-Free Network: Threshold Gap vs. Diffusion for Degree and Random Heuristics

Lastly for small-world networks, the lower bound is set to 0.1, and the gap is set to 0.6. This gives good enough conditions in the sense of distinguishing heuristics and having a margin to improve, as it can be seen in Figure 3.4. The node thresholds are assigned from a uniform distribution of [0.1, 0.7]. A range such as [0.1, 0.6] neither helps in distinguishing heuristics nor provides a margin to improve.

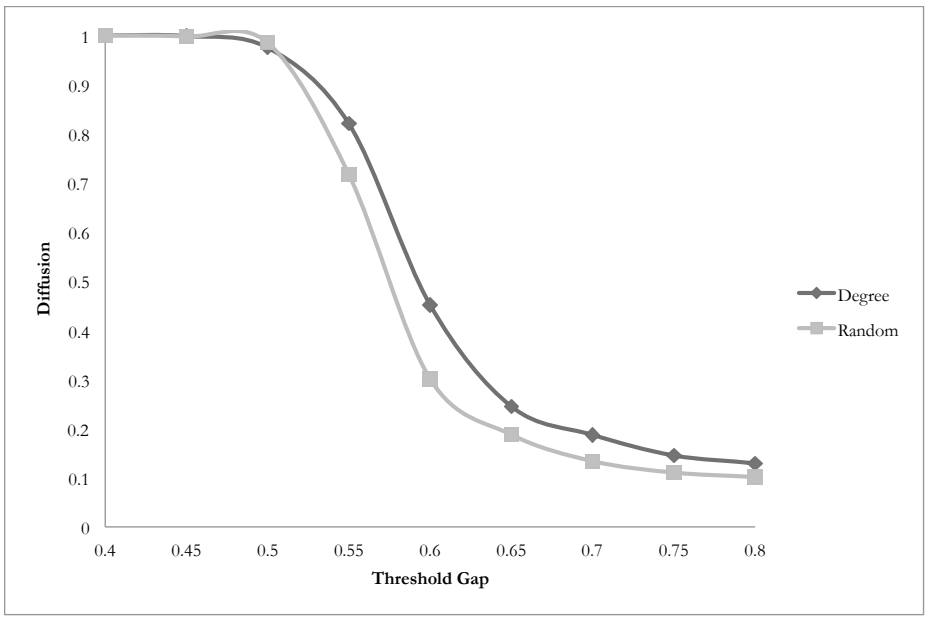

Figure 3.4. Small-World Network: Threshold Gap vs. Diffusion for Degree and Random Heuristics

Overall, these three graphs tell more than just helping to find reasonable ranges to distinguish heuristics. Firstly, trying to find a range for the uniform distribution of threshold values that will serve the purposes of the study means that there are such ranges that does not serve the purposes. There may be two reasons for not serving the purposes. First reason is that the heuristic performances are not distinguished, and secondly there is no margin to improve for the heuristics to be proposed. Here, the first reason is more important in evaluating the graphs. The ranges in which heuristics are not distinguished also means that in such ranges it is unnecessary to struggle to find good seed sets with a given seed set size, since any heuristic will give the same result. For example, in random networks, a uniform threshold distribution in range of [0.1, 0.7] results with almost no diffusion, independent of choosing seeds randomly or with some intelligently designed heuristic. Similarly, in case of scale-free networks, when the distribution range is in [0.1, 0.4] any heuristic results in a full diffusion. Thus the choice of heuristic does not matter and has no advantage over selecting seeds randomly. So, these three graphs tell that heuristics may not always be important, since there can be cases where even random seed selection causes full diffusion or it is impossible

to create diffusion at all.

The second point that these graphs make is that there are differences between types of networks. This has been known already since the characteristics of the types of networks are different. However these graph also imply that the ranges of threshold distributions where intelligent seed selection matters also differs. For example, the range used for random networks is [0.1, 0.45]. If this range is used for scale-free or small-world networks, it will not be possible to distinguish the heuristics because all will perform a full diffusion. Another difference is seen when the graphs are compared. For random and scale-free networks, the heuristics can be distinguished in a number of ranges, their performances are different for a larger range when compared to small-world networks. In small-world networks, the heuristics can be distinguished in only a small portion of threshold gap values. These two differences tell that the characteristics of networks also matter when it comes to deciding whether it is necessary to do intelligent seed selection or it does not matter since selecting seeds randomly gives the same result.

4. HEURISTICS

In this study, the random seed selection heuristic has been used as a benchmark. Besides the random seed selection heuristic, there are two categories of heuristics used, namely the group 1 and group 2 heuristics. The group 1 heuristics assume the whole information on a network is available. Some of the heuristics in this group are taken from the literature on seed selection and some are designed for this study. The group 2 heuristics on the other hand do not assume the whole network information is globally available, they gather information locally from nodes and make the seed selection decision accordingly. The random seed selection does not belong to either of these groups. Next, all heuristics with promising results used in the study will be briefly explained along with their pseudocodes. The remaining heuristics that have been used in the study, which either perform poorly or similar to some of the used heuristics, which can be seen in Appendix A. In the explanations and the pseudocodes, s represents the seed set size, v is used for the metric of a node that the heuristic is using, d for the degree of a node, t for the threshold of a node, tg for the threshold grouping value of a node, and the term *activate* represents a node's selection as a seed. The α constant used for the degree effect is set to be 0.8 in the study, after observation of its performance through a range of values.

Random (R): Ask randomly selected s nodes from the network to activate.

```
for randomly selected s nodes do
  activate
end for
```

Figure 4.1. Pseudocode of R

4.1. Group 1 Heuristics

Degree (D): Ask s nodes in the network with maximum degree to activate.

```
for each node x do
  set v_x ← count neighbors
end for
for s nodes with maximum v values do
  activate
end for
```

Figure 4.2. Pseudocode of D

Degree Discount (DD): This heuristic computes the number of inactive neighbors for each inactive node and will select the one with the maximum metric to be activated. This procedure is repeated s times.

```
for i=1 to s do
  for each node x do
    set v_x ← count inactive neighbors
  end for
  for the node with maximum v do
    activate
  end for
end for
```

Figure 4.3. Pseudocode of DD

Average Threshold (AT): Choose s nodes in the network with minimum average threshold metric to activate. The average threshold for node x is calculated as the mean t of the neighbors of node x. This heuristic is only applicable to the LT model since thresholds are not available in IC.

Average Threshold with Degree Effect (ATwD): The metric used in this heuristic is calculated using the average threshold value of a node combined with its degree along with an effect constant. The heuristic selects s nodes with the minimum metric and

activates them. This heuristic is only applicable to the LT model.

for each node x **do**
  set $v_x \leftarrow$ mean t of neighbors
end for
for s nodes with minimum v values **do**
  activate
end for

Figure 4.4. Pseudocode of AT

for each node x **do**
  set $v_x \leftarrow$ (mean t of neighbors) $\times$ α^{d_x}
end for
for s nodes with minimum v values **do**
  activate
end for

Figure 4.5. Pseudocode of ATwD

Average Threshold with Square Root Degree Effect (ATwSD): Similar to *ATwD*, takes the same values from a node but uses the degree as a square root. Chooses s nodes with the minimum metric and activates them. This heuristic is only applicable to the LT model.

Since quantifying the threshold values of a node is very hard to accomplish in real life, it is decided to categorize nodes into five groups depending on their threshold values and use these in heuristics. When the threshold group of a node is lower, it takes less effort to activate it. The nodes are divided into five equally large groups in terms of thresholds, using the threshold distribution range of each type of network.

Average Threshold with 5 Groups (ATw5Gr): Instead of taking the exact t values of nodes, all nodes are divided into equally large five groups depending on their thresh-

for each node x **do**
  set $v_x \leftarrow$ (mean t of neighbors) $\times \alpha^{\sqrt{d_x}}$
end for
for s nodes with minimum v values **do**
  activate
end for

Figure 4.6. Pseudocode of ATwSD

old values. Afterwards, each node calculates its metric similar to AT and s nodes with the minimum metric are activated. This heuristic is only applicable to the LT model.

Average Threshold with 5 Groups with Square Root Degree Effect (ATw5GrwSD): Similar to $ATw5Gr$, only with an addition of square root degree effect. Then the heuristic chooses s nodes with the minimum metric and activates them. This heuristic is only applicable to the LT model.

Strength (S): Choose s nodes in the network with maximum sum of outgoing link strength to activate.

Twostep (TS): Adopted from [47], this heuristic is similar to D, but as a difference, takes all nodes into account that can be reached in 2 steps.

Betweenness (B): Choose s nodes in the network with maximum betweenness centrality metric to activate.

Closeness (C): Choose s nodes in the network with maximum closeness centrality metric to activate.

Eigenvector (E): Choose s nodes in the network with maximum eigenvector centrality metric to activate.

```
for each node x do
  for k=5 to 1 do
    if t_x < 0.2k× (threshold gap) + (lower threshold) then
      set tg_x ← k
    end if
  end for
end for
for each node x do
  set v_x ← mean tg_x of neighbors
end for
for s nodes with minimum v values do
  activate
end for
```

Figure 4.7. Pseudocode of ATw5Gr

```
for each node x do
  for k=5 to 1 do
    if t_x < 0.2k× (threshold gap) + (lower threshold) then
      set tg_x ← k
    end if
  end for
end for
for each node x do
  set v_x ← (mean tg_x of neighbors) × α^√d_x
end for
for s nodes with minimum v values do
  activate
end for
```

Figure 4.8. Pseudocode of ATw5GrwSD

for each node x **do**

  set $v_x \leftarrow$ sum strength of outgoing links

end for

for s nodes with maximum v values **do**

  activate

end for

Figure 4.9. Pseudocode of S

for each node x **do**

  set $v_x \leftarrow$ count nodes within 2 steps of reach

end for

for s nodes with maximum v values **do**

  activate

end for

Figure 4.10. Pseudocode of TS

for each node x **do**

  set $v_x \leftarrow$ calculate betweenness centrality

end for

for s nodes with maximum v values **do**

  activate

end for

Figure 4.11. Pseudocode of B

for each node x **do**

  set $v_x \leftarrow$ calculate closeness centrality

end for

for s nodes with maximum v values **do**

  activate

end for

Figure 4.12. Pseudocode of C

for each node x **do**

  set $v_x \leftarrow$ calculate eigenvector centrality

end for

for s nodes with maximum v values **do**

  activate

end for

Figure 4.13. Pseudocode of E

PageRank (PR): Choose s nodes in the network with maximum PageRank value to activate.

```
for each node x do
  set v_x ← calculate PageRank
end for
for s nodes with maximum v values do
  activate
end for
```

Figure 4.14. Pseudocode of PR

4.2. Group 2 Heuristics

Degree within 1 Step (Dw1S): This heuristic first accesses to an inactive node. Then it selects the inactive neighbor of this node with the highest degree as a seed. This procedure is repeated s times. In real life, this heuristics corresponds to asking someone about his most popular friend and selecting this friend as a seed.

```
for each node x do
  set v_x ← count neighbors
end for
for i=1 to s do
  select an inactive node x randomly
  for the inactive node within 1 step of node x with the maximum v value do
    activate
  end for
end for
```

Figure 4.15. Pseudocode of Dw1S

Degree within 2 Steps (Dw2S): Similar to $Dw1S$, this heuristic accesses to an inactive node. Then it selects an inactive node with the highest degree as a seed which is either a neighbor or neighbor of a neighbor. This procedure is repeated s times.

```
for each node x do
  set v_x ← count neighbors
end for
for i=1 to s do
  select an inactive node x randomly
  for the inactive node within 2 steps of node x with the maximum v value do
    activate
  end for
end for
```

Figure 4.16. Pseudocode of Dw2S

Average Threshold within 1 Step (ATw1S): This heuristic first accesses to an inactive node. Then it selects the inactive neighbor of this node with the minimum average threshold value as a seed. This procedure is repeated s times. In real life, this corresponds to asking someone how easily its neighbors convince their friends. This heuristic is only applicable to the LT model.

Average Threshold within 2 Steps (ATw2S): Similar to $ATw1S$, this heuristic accesses to an inactive node, and then it selects the inactive neighbor of this node with the minimum average threshold value within the reach of 2 steps as a seed. This procedure is repeated s times. This heuristic is only applicable to the LT model.

Average Threshold with Square Root Degree Effect within 1 Step (ATwSDw1S): Similar to $ATw1S$, but also uses the degree of a node. An inactive node is selected and the neighbor with the minimum metric is activated. This process is repeated s times. This heuristic is only applicable to the LT model.

```
for each node x do
  set v_x ← mean t of neighbors
end for
for i=1 to s do
  select an inactive node x randomly
  for the inactive node within 1 step of node x with the minimum v value do
    activate
  end for
end for
```

Figure 4.17. Pseudocode of ATw1S

```
for each node x do
  set v_x ← mean t of neighbors
end for
for i=1 to s do
  select an inactive node x randomly
  for the inactive node within 2 steps of node x with the minimum v value do
    activate
  end for
end for
```

Figure 4.18. Pseudocode of ATw2S

for each node x **do**
  set $v_x \leftarrow$ (mean t of neighbors) $\times \alpha^{\sqrt{d_x}}$
end for
for i=1 to s **do**
  select an inactive node x randomly
  for the inactive node within 1 step of node x with the minimum v value **do**
    activate
  end for
end for

Figure 4.19. Pseudocode of ATwSDw1S

Average Threshold with 5 Groups within 1 Step (ATw5Grw1S): The nodes are grouped according to their threshold values similar to $ATw5Gr$. Then, an inactive node is selected and the inactive node with the minimum metric within the reach of 1 step is activated. This procedure is repeated s times. This heuristic is only applicable to the LT model.

Average Threshold with 5 Groups with Square Root Degree Effect within 1 Step (ATw5GrwSDw1S): Similar to $ATw5Grw1S$, the nodes are grouped according to their threshold values, but the degree of the node is also used while calculating the metric. Then, an inactive node is selected and the inactive node with the minimum metric within the reach of 2 steps is activated. This procedure is repeated s times. This heuristic is only applicable to the LT model.

Strength within 1 Step (Sw1S): This heuristic accesses to an inactive node and selects the inactive node with the maximum sum of outgoing link strength within the reach of 1 step to activate. This procedure is repeated s times.

```
for each node x do
  for k=5 to 1 do
    if t_x < 0.2k× (threshold gap) + (lower threshold) then
      set tg_x ← k
    end if
  end for
end for
for each node x do
  set v_x ← mean tg_x of neighbors
end for
for i=1 to s do
  select an inactive node x randomly
  for the inactive node within 1 step of node x with the minimum v value do
    activate
  end for
end for
```

Figure 4.20. Pseudocode of ATw5Grw1S

for each node x **do**
  for k=5 to 1 **do**
    if $t_x < 0.2k\times$ (threshold gap) + (lower threshold) **then**
      set $tg_x \leftarrow k$
    end if
  end for
end for
for each node x **do**
  set $v_x \leftarrow$ (mean tg_x of neighbors) $\times \alpha^{\sqrt{d_x}}$
end for
for i=1 to s **do**
  select an inactive node x randomly
  for the inactive node within 2 steps of node x with the minimum v value **do**
    activate
  end for
end for

Figure 4.21. Pseudocode of ATw5GrwSDw1S

for each node x **do**
  set $v_x \leftarrow$ sum strength of outgoing links
end for
for i=1 to s **do**
  select an inactive node x randomly
  for the inactive node within 1 step of node x with the maximum v value **do**
    activate
  end for
end for

Figure 4.22. Pseudocode of Sw1S

Twostep within 1 Step (TSw1S): This heuristic accesses to an inactive node. Then it activates an inactive neighbor who has the maximum number of connections within 2 steps. This procedure is repeated s times

```
for each node x do
  set v_x ← count nodes within 2 steps of reach
end for
for i=1 to s do
  select an inactive node x randomly
  for the inactive node within 1 step of node x with the maximum v value do
    activate
  end for
end for
```

Figure 4.23. Pseudocode of TSw1S

Degree over s Nodes within 1 Step (Ds1): A node is selected and the metric for its neighbors is increased by 1. The metric for the selected node is also increased by 0.5 since the choice creates a minor disadvantage for the selected node. The procedure is repeated s times. At the end, s nodes with the maximum metric are chosen to be activated.

```
for i=1 to s do
   select a random node x
   set v_x ← v_x + 0.5
   for all neighbors y of node x do
      set v_y ← v_y + 1
   end for
end for
for s nodes with maximum v value do
   activate
end for
```

Figure 4.24. Pseudocode of Ds1

5. EXPERIMENTATION AND RESULTS

5.1. Validation and Verification

For the validation and verification of the agent-based model, all three steps of the study should be considered. These should look if the created networks are formed as they are intended, if the seed selection process is executed as designed, and if the diffusion models are working properly.

First to validate the accuracy of networks, their clustering coefficients, average path length and the degree distribution are looked at. As it can be seen in Table 5.1, all three networks satisfy the conditions that they are characterized with. Random networks have very low clustering coefficient along with a low average path length, while scale-free networks also have low clustering coefficients with low average path length but yet higher than that of random networks'. Small-world networks, on the other hand, have very high clustering coefficients, along with a low average path length, while it is still higher than the average path length of the other two networks.

Table 5.1. Clustering Coefficients and Average Path Lengths of Networks

	Clustering Coefficient	**Average Path Length**
Random	0.050	2.029
Scale-Free	0.045	2.983
Small-World	0.490	4.423

Another way to check the accuracy of networks created by the agent-based model is looking at the degree distributions. As it can be seen in Figure 5.1, the distribution for the random network is like a normal distribution with a mean of 50, which is the expected number of neighbors for a node in this network.

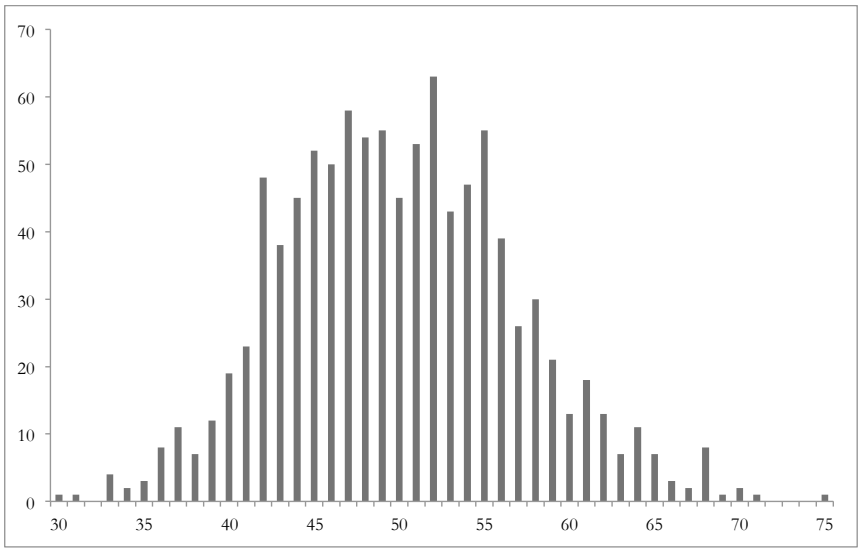

Figure 5.1. Degree Distribution of a Sample Random Network

In Figure 5.2 it can be seen that the degree distribution is following a power-law scaling, which is as expected for scale-free networks. A lot of nodes have very low degrees, but there are some nodes that have very high degrees, which consequently form the hubs in those networks.

In Figure 5.3 it is seen that the degrees of nodes are distributed around a mean of 10 with a low variation, as it should be in a small-world network.

The validation for seed selection and diffusion models is done throughout the designing process of the agent-based model. For each heuristic, it is validated if the selected seeds are indeed the ones that should be in the seed set based on the metric used. For the diffusion models, the LT model is validated via the observation of several simulation runs and checking the accuracy of the process algebraically. In the case of IC model, the simulation runs are observed along with the random numbers generated to apply the probabilistic processes in the nature of the diffusion model, and it is validated as such.

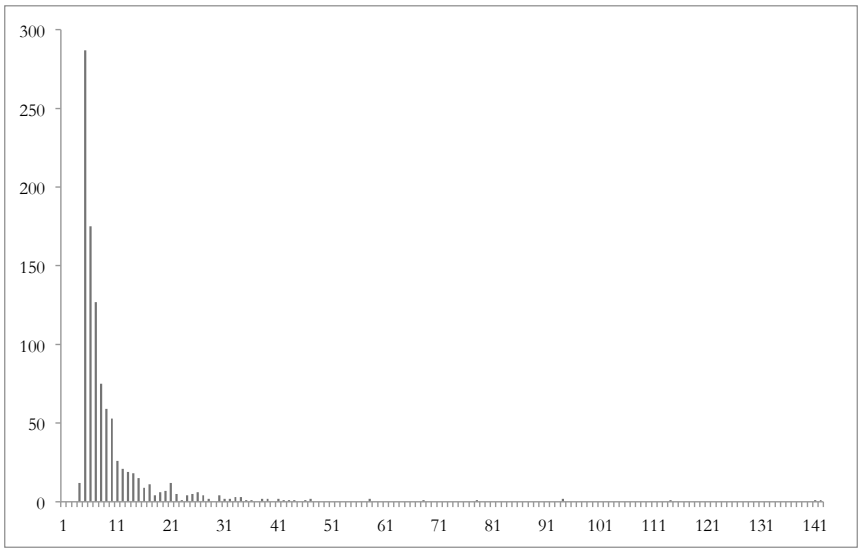

Figure 5.2. Degree Distribution of a Sample Scale-Free Network

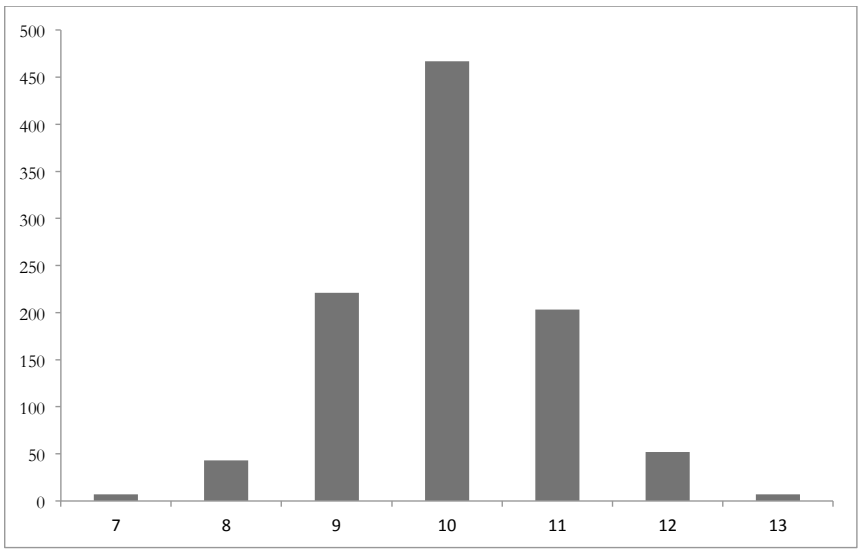

Figure 5.3. Degree Distribution of a Sample Small-World Network

The verification of the agent-based model is done using extreme condition tests. To apply these tests, a sample network is created with 10 nodes as it can be seen in Figure 5.4a, and one seed is selected as seen in Figure 5.4b, where red nodes are inactive and greens are active.

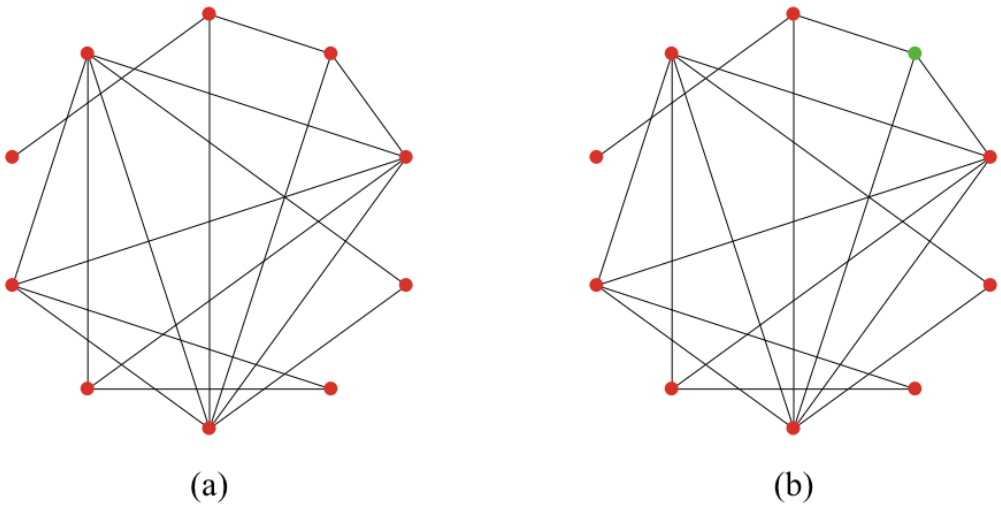

Figure 5.4. A Sample Network and Selected Seed

To verify the agent-based model, the threshold values is set to 0 and 1 and propagation is done using LT model, and separately the link strengths are set to 0 and 1 and propagation is observed using IC model. Before the changes in thresholds and link strengths, in Figure 5.5 it can be seen what would have happened, without changing these values. It is important to point out that prior is the only way it would end up since LT is a deterministic model, whereas the latter may have had a different form since IC is probabilistic. The blue links in Figure 5.5b represent the diffusion attempts happened through these, successful or unsuccessful, and black ones indicate that there was no diffusion attempt.

When the node thresholds are set to 1, it is expected that no node becomes active other than the seed as long as all neighbors of a node are not active. On the contrary, when thresholds are set to 0, a full diffusion is expected. As it can be seen in Figure 5.6, the model simulates the diffusion processes as expected.

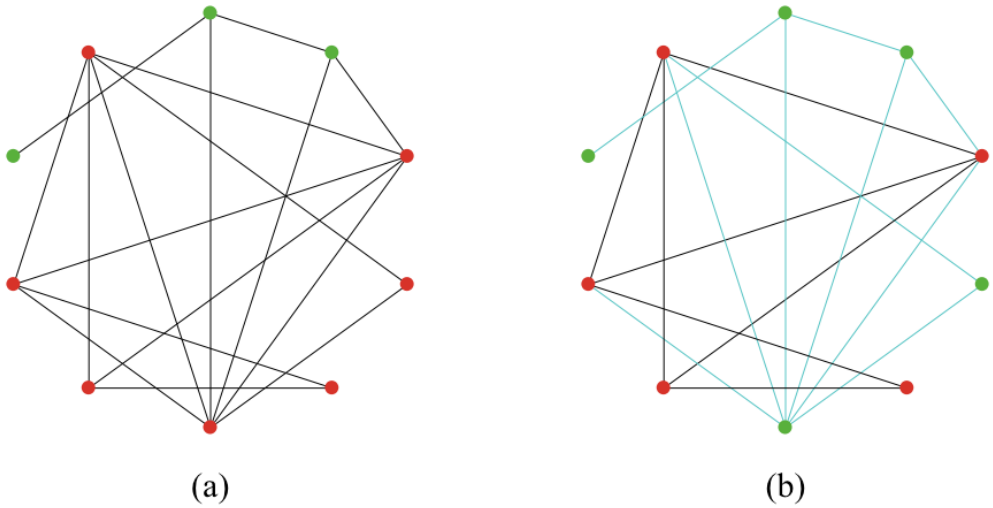

Figure 5.5. Sample Propagations using Linear Threshold and Independent Cascade Models

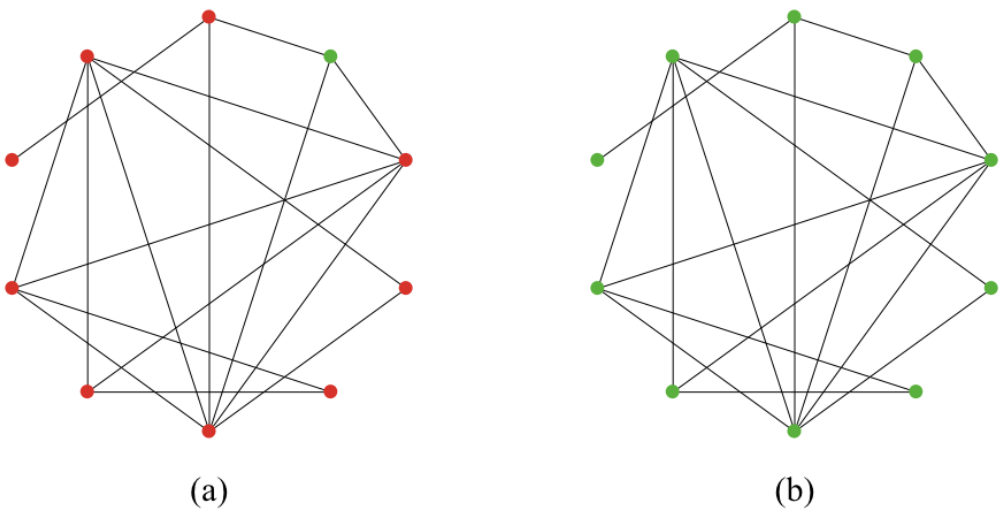

Figure 5.6. Final Diffusion with Thresholds=1 and Thresholds=0

Using the IC model, when link strengths are set to 0, it is expected that no node becomes active other than the seed, and when the strengths are set to 1, the final diffusion is always full. As it is seen in Figure 5.7, the diffusions are simulated and the results are as expected.

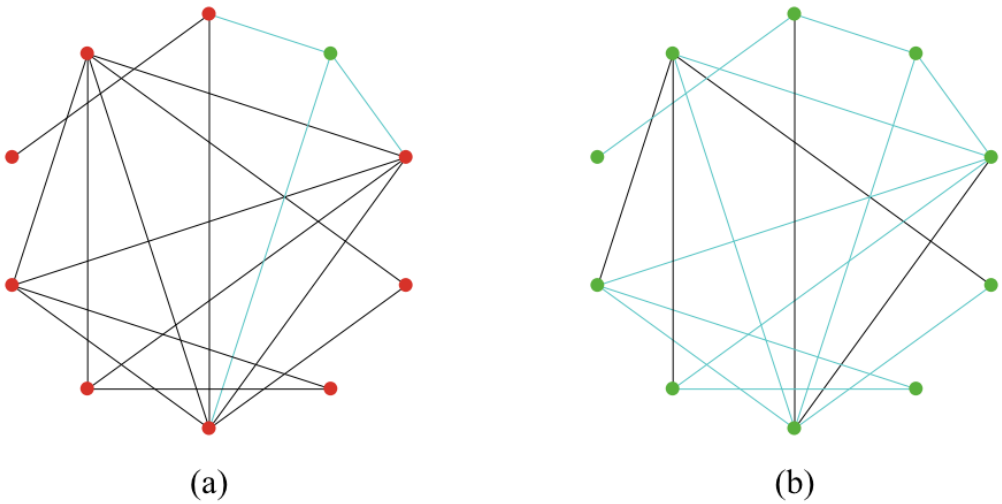

Figure 5.7. Final Diffusion with Link Strengths=0 and Link Strentghs=1

5.2. Experimentation Procedure

The BehaviorSpace tool of NetLogo is used during the experimentation. This tool enables the user to run multiple simulations with replications. Also by defining the necessary parameters, the simulations are executed. These parameters may either be a single value or multiple values, which helps the user to apply sensitivity analysis.

For this study, 100 networks of each type have been generated using 100 different random seeds along with 10 replications for each network and heuristic combination. Thus, there has been 1000 runs for each heuristic and for each network and diffusion model combination. The performance and runtime of each heuristic has been acquired through these 1000 runs.

In BehaviorSpace, each parameter for creating a network or simulating a diffusion model is defined, such as the rewiring probability for a small-world network, or the threshold gap for LT model. Also, all heuristics are given in order to observe their performances. After all, the parameters are defined, and the simulation is run using the functions coded in NetLogo.

After the data for heuristic performances is obtained, the performances of heuristics are analyzed by averaging over 10 replications and 100 different networks of a given type. There are 3 different network types and 2 diffusion models, thus 6 set of results are obtained at the end of the experimentation. There are 114000 simulation runs overall, 24000 for each experiment using LT model and 14000 for each using IC model. IC model has less simulation runs since some heuristics are only applicable to LT model.

5.3. Results

The combinations of network types and diffusion models create six sets of results, which are illustrated and discussed in the following sections. The average runtime of each heuristic across six experiment sets can be seen in Table 5.2.

Table 5.2. Average Heuristic Runtimes (in seconds)

R	0.0002	**S**	0.0052	**ATw1S**	0.0105
D	0.0005	**TS**	0.3330	**ATw2S**	0.0103
DD	0.1009	**B**	8.4030	**ATwSDw1S**	0.0108
AT	0.0048	**C**	17.0212	**ATw5Grw1S**	0.0110
ATwD	0.0054	**E**	0.7542	**ATw5GrwSDw1S**	0.0110
ATwSD	0.0053	**PR**	0.9126	**Sw1S**	0.0102
ATw5Gr	0.0049	**Dw1S**	0.0064	**TSw1S**	0.3363
ATw5GrwSD	0.0054	**Dw2S**	0.0064	**Ds1**	0.7145

It can be observed that most heuristics have a computation time of approximately a tenth of a second or less, while some require a time of up to 1 second, such as *PR*, and some take even longer to compute. *C* takes 8.5 seconds and *B* takes 17 seconds in order to find the seed set. It can also be observed that group 2 heuristics have 0.05 seconds more computation time than their group 1 versions on average. It is important to remind that all heuristics have a diffusion of at least 5% since it is the seed set size.

5.3.1. Random Networks with Linear Threshold Model

Table 5.3. Heuristic Performances on Random Networks with LT Model

R	5.4%	**S**	58.9%	**ATw1S**	83.7%
D	57.2%	**TS**	38.8%	**ATw2S**	87.6%
DD	62.9%	**B**	57.1%	**ATwSDw1S**	95.0%
AT	85.2%	**C**	44.6%	**ATw5Grw1S**	75.9%
ATwD	61.7%	**E**	44.9%	**ATw5GrwSDw1S**	96.0%
ATwSD	98.1%	**PR**	58.9%	**Sw1S**	49.2%
ATw5Gr	83.0%	**Dw1S**	49.3%	**TSw1S**	31.7%
ATw5GrwSD	99.1%	**Dw2S**	55.6%	**Ds1**	7.1%

The performances of heuristics on random networks with LT model is illustrated in Table 5.3. The heuristics *R* and *Ds1* are the ones with the worst performance with 5.4% and 7.1% respectively. While the prior is expected to perform poorly, the result for the latter shows that the designed group 2 heuristic does not provide the intended results in this case.

The heuristic *TS* and its group 2 version *TSw1S* follow *Ds1* in terms of poor performance, while *D* and its group 2 versions *Dw1S* and *Dw2S* perform better. This is interesting since *D* takes into consideration only the nodes within the reach of 1 step, while *TS* has a wider range but performs poorer. This may indicate that it is not always better to have a wider perspective, since looking only to direct neighbors has proven to be better under these conditions. *DD* has performed better than *D* by

10% as expected (D has a performance of 57.2% and DD has a performance of 62.9%), since it is designed to cover more area than D by the elimination of counting already activated nodes.

The heuristic S and its group 2 version *Sw1S* have performed similar to D and *Dw1S*. This may be due to the fact that link strengths are correlated with degrees of nodes due to the design of the study.

The heuristics using centrality measures, B, C, E, and PR have performed worse than expected, especially C, 44.6% and E, 44.9%, since they did not reach the performance of D, while being much slower, respectively by 34000 times and 1500 times. On the other hand B and PR have close performances to D, although they are still much slower than it.

The most significant results in this set are obtained from heuristics using the metric average threshold. *ATwD* is the heuristic with the poorest performance among those using the metric average threshold with 61.7%. This is probably due to the heavy effect of degree on the metric, but it still outperforms all the other heuristics. This is a sign for the metric being promising. The heuristic AT itself proves to perform good, while adding the squared root degree effect to it, increases its performance by 15%, up to 98.1%. Grouping the threshold values itself does not deteriorate the results much, while adding the squared root degree effect, the heuristic *ATw5GrwSD* is the best performing heuristic with 99.1%.

The best performing group 1 heuristics are *ATwSD* and *ATw5GrwSD* with almost full diffusion on average, and their group 2 versions are not much worse, with only a loss of about 3% of performance for both. This is a highly promising results since the group 1 heuristics choose the nodes with the best metric, while the group 2 heuristics use nodes to reach other nodes in a limited range. This shows that reaching information locally, i.e. using group 2 heuristics do not always cause a significant loss of information.

Another important point on these heuristics using the average threshold metric, especially for *ATwSD*, *ATw5GrwSD* and their group 2 versions, is that they are computationally efficient. The process of seed selection takes approximately 0.01 seconds, which is 800 times faster than the heuristic using betweenness centrality, while performing almost twice as better than it, which is known to be a promising centrality measure. One last remark can be made on the variations of their performance, which can be observed in Figure 5.8. Although there are some outliers around 10%, most of the heuristics using the average threshold metric except for *ATw5Grw1S* and *ATwD*, have low variation, especially *ATwSD* and *ATw5GrwSD* which have very few outliers. This is an important point since a low variation means consistently good results in this case. A further analysis and explanation on the box-plot is conducted below.

As it can be seen, the metric average threshold has proved to be an important factor in seed selection and its performance increases with its combination of degree. There can be two reasons behind this metric outperforming degree-centrality. First is that in random networks the degrees of nodes are distributed around a mean with a relatively low variation (as it can be seen in Figure 5.1), thus most of the nodes have close degree values. This brings the necessity of distinguishing nodes through different metrics. However this does not mean that degree notion is completely useless in random networks, since it has been show that the combination of average threshold and degree increases the performance of heuristics.

The second, and seemingly more important one needs more in-depth analysis. When the Figure 5.8 is observed it can be seen that there is an odd situation. Some box-plots only consist of lines, and some extend all the way from 0.05 to 1. When the experimentation outputs are analyzed, it is seen that in each case the diffusion has either stayed in the range of 0.05 and 0.1, or full diffusion has happened. For example, in the case of *ATwSD*, since almost all simulation runs end with full diffusion except for some outliers, its box-plot is almost a line at the level of 1. For *ATwD*, although its median is at 1, since it has relatively more data points around 0.05, its box-plot extends all the way from 0.05 to 1. And finally for *Ds1*, since it performs consistently poor, it has a box-plot near the level of 0.05, which is like a line. This shows the importance of

starting the diffusion, since after a tipping point, which seems to be around 0.1, a full diffusion occurs. This may be why the threshold metric turns out to be an important metric. Since the metric takes into consideration how easily the neighbors of a node are convinced, choosing nodes with a low metric will increase the probability of a kick-off in the diffusion, thus reaching the tipping point more easily and creating the snowball effect to end up with a full diffusion.

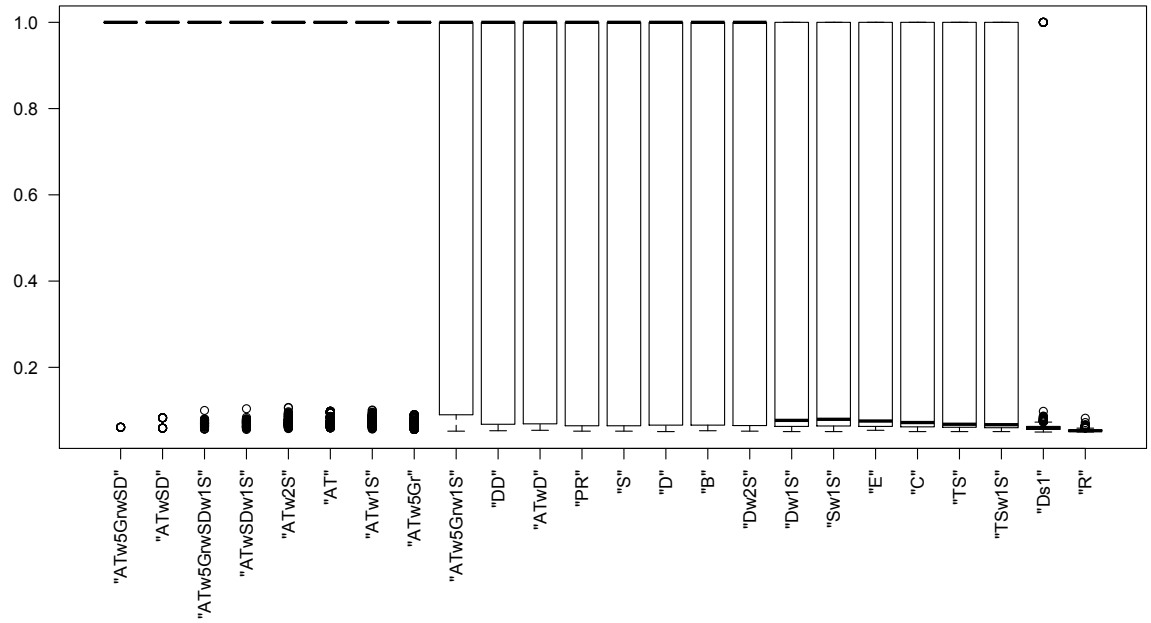

Figure 5.8. Box Plots of Heuristic Performances on Random Networks with LT Model

5.3.2. Scale-Free Networks with Linear Threshold Model

The results on heuristic performances for scale-free networks with LT diffusion model can be seen in Table 5.4. The heuristic R is the worst performing one as expected with 7.8%. Differently from random networks, heuristics using the average threshold metric without the degree effect are the ones with the poorest performances after R. This is an interesting finding since these heuristics had significant performances in random networks, but that is not the case in scale-free networks. This may be due to the structure of networks, which is very much different since scale-free network have power-law degree distributions compared to a normal distribution of degrees in random networks. However, once the degree effect is introduced to the AT-type heuristics, the

performance is enhanced and become very close to the ones with the best performances, especially *ATwD* which is highly effected by degrees of nodes and performs as good as *D* and *DD*. This can again prove the point that the combination between degree and average threshold is promising.

Table 5.4. Heuristic Performances on Scale-Free Networks with LT Model

R	7.8%	**S**	59.7%	**ATw1S**	15.3%
D	59.4%	**TS**	56.0%	**ATw2S**	10.2%
DD	59.7%	**B**	59.3%	**ATwSDw1S**	51.7%
AT	9.7%	**C**	56.5%	**ATw5Grw1S**	15.0%
ATwD	59.5%	**E**	55.9%	**ATw5GrwSDw1S**	51.7%
ATwSD	54.5%	**PR**	59.6%	**Sw1S**	53.0%
ATw5Gr	9.4%	**Dw1S**	52.9%	**TSw1S**	51.0%
ATw5GrwSD	54.7%	**Dw2S**	57.6%	**Ds1**	48.1%

The heuristics *D* and *DD* are among the best performing ones with 59.4% and 59.7% respectively. This indicates that the notion of degree is important in scale-free networks. The group 2 heuristics using the degree measure, namely *Dw1S*, *Dw2S*, and *Ds1*, have significant performances considering their stand against the assumption of globally available information. Especially *Dw2S* performs only 3% worse than *D*, which can be regarded as a significant achievement. *S* and *Sw1S* have similar performances with their counterparts using degree (*D*, *Dw1S*, *Dw2S*), which is expected since the two metrics are correlated. *TS* and *TSw1S* are again outperformed by *D*, *Dw1S*, and *Dw2S*, although the difference now is not very significant.

The heuristics using centrality measures, especially *B* and *PR* are among the best performing heuristics this time with 59.3% and 59.6% respectively. However, since their computational cost is relatively high, 8.4 seconds for *B* and 0.9 seconds for *PR*, they are still not preferred over, for example *D*, which has a performance of 59.4% with a computation time of 0.0005 seconds.

Comparing the variation of results for the heuristics from Figure 5.9, it can be said that most of them are relatively similar, both in the sense of the amount of variation and whiskers, which gives somehow an upper limit for performance.

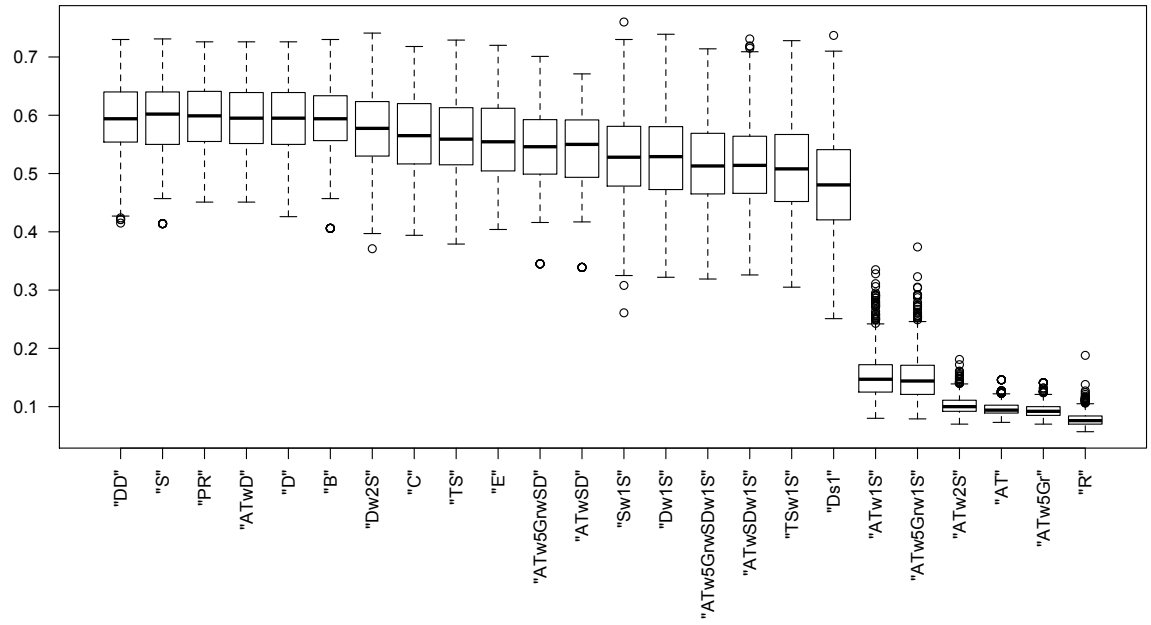

Figure 5.9. Box Plots of Heuristic Performances on Scale-Free Networks with LT Model

Comparing the results discussed until now, these can be interpreted as there is an effect of structure of networks on seed selection performances, which may make one consider if the process of seed selection should be treated differently depending on the structure of a network.

5.3.3. Small-World Networks with Linear Threshold Model

From the heuristic performances illustrated in Table 5.5 it can be seen that this time R is not the worst performing heuristic, which is an interesting result. This means that certain heuristics, which are intended to be designed to serve a purpose, are outperformed by selecting seeds randomly. Another way to look at this can be that this type of networks may be requiring some randomness in the process of seed selection. This claim can only be strengthened or weakened by analyzing the performances of

other heuristics.

Table 5.5. Heuristic Performances on Small-World Networks with LT Model

R	29.3%	**S**	45.6%	**ATw1S**	51.3%
D	44.6%	**TS**	32.9%	**ATw2S**	50.6%
DD	48.8%	**B**	33.2%	**ATwSDw1S**	52.4%
AT	31.7%	**C**	25.7%	**ATw5Grw1S**	50.3%
ATwD	46.9%	**E**	14.9%	**ATw5GrwSDw1S**	52.1%
ATwSD	32.6%	**PR**	45.6%	**Sw1S**	41.4%
ATw5Gr	32.3%	**Dw1S**	40.7%	**TSw1S**	32.9%
ATw5GrwSD	33.6%	**Dw2S**	41.8%	**Ds1**	17.4%

The degree and strength metrics have close performances along with their group 2 versions. This indicates that degree measure can be a deciding factor in the decision of seed selection for small-world networks. *DD* is among best performing heuristics with 48.8%, while *Dw1S* and *Dw2S* perform only 9% and 6% worse than *D* respectively, and *Sw1S* performs 9% worse than *S*. These results show that group 2 heuristics are not much worse than their group 1 versions. The only exception here is *Ds1*, which performs even worse than *R* with 17.4% while *R* has a performance of 29.3%. Similar to the case in random networks, *TS* and *TSw1S* perform around 20% worse than similar heuristics using degree and strength. This is an interesting result since the idea behind these heuristics were to be able to look at a network more globally by increasing the metric range to two steps of reaching, but the results for all network types, including small-world networks, indicate that such a wider range does not have an impact on the performances as intended.

The heuristics *B*, *C*, and *E* have poor performances. Especially *E* and *C* perform among the worst with 14.9% and 25.7%. This performances are 50% and 12% worse than *R*, respectively. *B* is also outperformed by most of the heuristics. This may be due to the nature of small-world networks. Since these networks are created using a ring lattice, at the beginning all nodes are identical, except for their threshold values,

which are not used in these centrality measures. By rewiring a proportion of links, a small-world network is achieved, but it may be the case that these centrality measures do not change enough to distinguish between good and bad seeds. However, this is not the case for *PR*, which has a performance similar to *D* with 45.6%. This may indicate that while other centrality measures are facing the explained situations, *PR* can still manage to distinguish between nodes effectively.

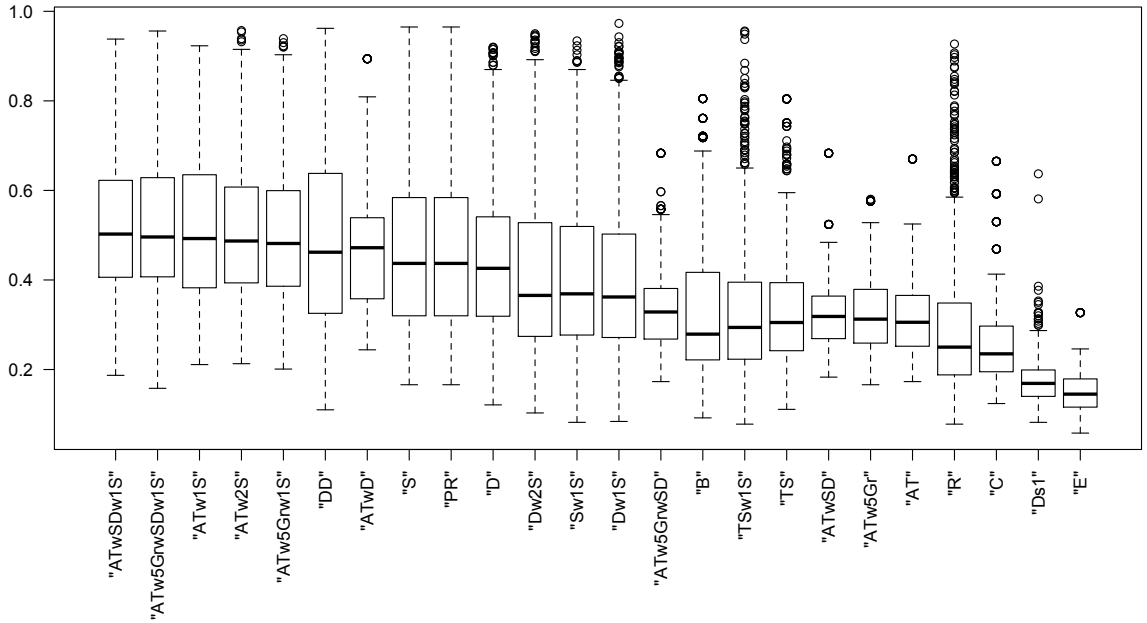

Figure 5.10. Box Plots of Heuristic Performances on Small-World Networks with LT Model

The most interesting results are obtained from heuristics using the average threshold metric. The group 1 heuristics using this metric have performances slightly better than *R* with a performance of 32% on average.The only exception here is *ATwD*, which performs better than others with 46.9%, probably due to its heavier dependence on degree measure. However, the group 2 versions of these heuristics are the ones performing the best, where *ATwSDw1S* is the best one among these with 52.4%. This is interesting because a group 2 heuristic has less information that it can reach than its group 1 version, but it performs better. This fact can be supporting the claim expressed at the beginning of this section, which hypothesized that there may be a need of randomness in seed selection. Selecting seeds by asking nodes to gather information

limits the range of information, and selecting those initially asked nodes is what adds the randomness to the process, and these two are the main basis of the group 2 heuristics. The randomness of group 2 heuristics have somehow enhanced their performances compared to their group 1 versions, and the only rational explanation seems to be that randomness up to some degree is helpful in the seed selection process for small-world networks.

5.3.4. Random Networks with Independent Cascade Model

As it can be seen in Table 5.6, the performances of heuristics are very close. This may be due to the selection of assigning link strengths in this study. Since the performances are too close, t-tests with unequal variances are used between certain heuristics to see if the differences between their means are significant or not. In these tests, p-value is used as 0.05. The formula for t is

$$t = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{5.1}$$

where $\bar{x}$ is the mean of a sample, s is the standard deviation, and n is the sample size. The formula for df is

$$df = \frac{(\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2})^2}{\frac{(s_1^2/n_1)^2}{n_1 - 1} + \frac{(s_2^2/n_2)^2}{n_2 - 1}} \tag{5.2}$$

R is the heuristic with the poorest performance among all heuristics with 27.5%, which is expected. All the other heuristics, except for *Ds1* have close performances. Using a t-test, it is checked if the performance of *Ds1* (28.2%) is significantly better than R. As the results imply, which can be seen in Table 5.7, it performs significantly better than R.

After this conclusion, it can also be concluded that all other heuristics perform significantly better than R, but it is also necessary to check if they perform better than *Ds1*. The t-tests run for all heuristics suggest that the difference is significant, and in

the table a sample t-test for D can be seen.

Table 5.6. Heuristic Performances on Random Networks with IC Model

R	27.5%	**E**	30.4%
D	30.3%	**PR**	30.2%
DD	30.6%	**Dw1S**	30.3%
S	30.5%	**Dw2S**	30.2%
TS	30.0%	**Sw1S**	30.1%
B	30.4%	**TSw1S**	30.0%
C	30.2%	**Ds1**	28.2%

Another important point is to check if group 2 heuristics are outperformed or not. From the performances in Table 5.6, it can be seen that the performances of D, and *Dw1S* and *Dw2S* are almost the same, just like *TS* and *TSw1S*. A t-test is run for S and *Sw1S* and it is seen that the difference is not significant. So, it can be concluded that using group 2 heuristics, thus reaching information locally rather than globally does not deteriorate the performance significantly in this case.

Table 5.7. t-Tests for Heuristics on Random Networks with IC Model

	R & Ds1	Ds1 & D	S & Sw1S
Mean	0.275 - 0.282	0.282 - 0.303	0.305 - 0.301
Sample size	1000 - 1000	1000 - 1000	1000 - 1000
df	1990.8	1994.1	1996.8
t	-3.4536	-11.017	1.9417
p-value	0.0005648	0	0.05231
H_0	$\mu_1 = \mu_2$	$\mu_1 = \mu_2$	$\mu_1 = \mu_2$
Decision	Reject H_0	Reject H_0	Do not reject H_0

The close performances and variations of all heuristics may be related to the distribution of link strengths modeled in this study and the structure of the network. The

random networks have a degree distribution that have relatively low variance, which makes nodes similar in terms of their degrees. Also, the link strengths are modeled depending on the degrees of nodes, which may be the reason for closely performing heuristics. Still, the t-tests have shown that the differences are statistically significant, which means that intelligent seed selection heuristics may not be an endeavor in vain under different link strength distributions.

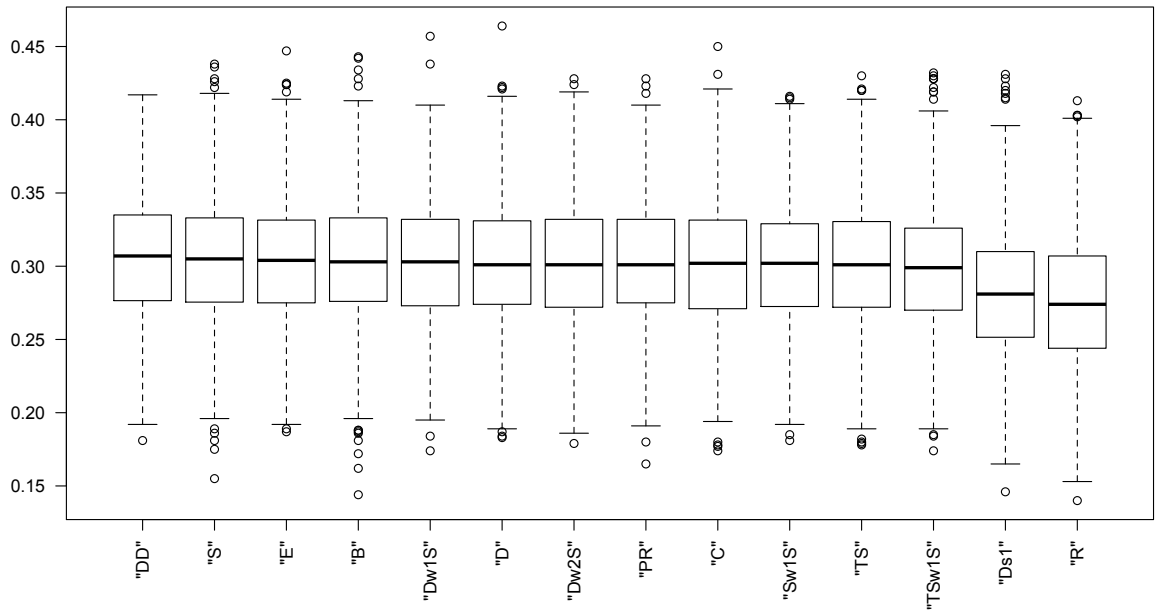

Figure 5.11. Box Plots of Heuristic Performances on Random Networks with IC Model

5.3.5. Scale-Free Networks with Independent Cascade Model

The results of heuristic performances on scale-free networks with IC diffusion model are illustrated in Table 5.8. R is outperformed by almost 90% by all the other heuristics, and these all have performances within a range of 4%. There is a similar case here with random networks, although R is the worst performing heuristic by far with 24.0%, the others are not distinguished too much. To comment on the performances, it can be said that D, S, DD, and PR are the best performing heuristics with a slightly greater performance than 46.0%.

Table 5.8. Heuristic Performances on Scale-Free Networks with IC Model

R	24.0%	**E**	44.9%
D	46.1%	**PR**	46.2%
DD	46.1%	**Dw1S**	44.1%
S	46.2%	**Dw2S**	45.6%
TS	45.1%	**Sw1S**	44.1%
B	45.9%	**TSw1S**	43.4%
C	45.1%	**Ds1**	42.6%

To observe the performance of group 2 heuristics, t-tests are run and the results are shown in Table 5.9. As it can be seen, all group 2 heuristics have performed worse than their group 1 versions, and this difference is statistically significant, although the differences are in the range of 1-4%, which is only a slightly worse performance and can be regarded as acceptable since these only use local information.

Table 5.9. t-Tests for Heuristics on Scale-Free Networks with IC Model

	D & Dw1S	D & Dw2S	S & Sw1S	TS & TSw1S
Mean	0.461 - 0.441	0.461 - 0.456	0.462 - 0.441	0.451 - 0.434
Sample size	1000 - 1000	1000 - 1000	1000 - 1000	1000 - 1000
df	1993.4	1996.1	1962.1	1989.4
t	14.814	3.5558	16.624	12.007
p-value	0	0.0003856	0	0
H_0	$\mu_1 = \mu_2$	$\mu_1 = \mu_2$	$\mu_1 = \mu_2$	$\mu_1 = \mu_2$
Decision	Reject H_0	Reject H_0	Reject H_0	Reject H_0

Although the heuristic R is outperformed by all the other heuristics, they are not distinguished much among themselves. The outperformance is most probably related to the degree distribution of scale-free networks. Different from random networks, there are some nodes with very high degrees in these networks, which act as hubs and

are very important. While R may skip to select such nodes, other heuristics will most likely put such nodes in their seed sets, which creates the somehow constant difference between R and all other heuristics. On the other hand, the close performance and variation of heuristics may again be related with the distribution of link strengths.

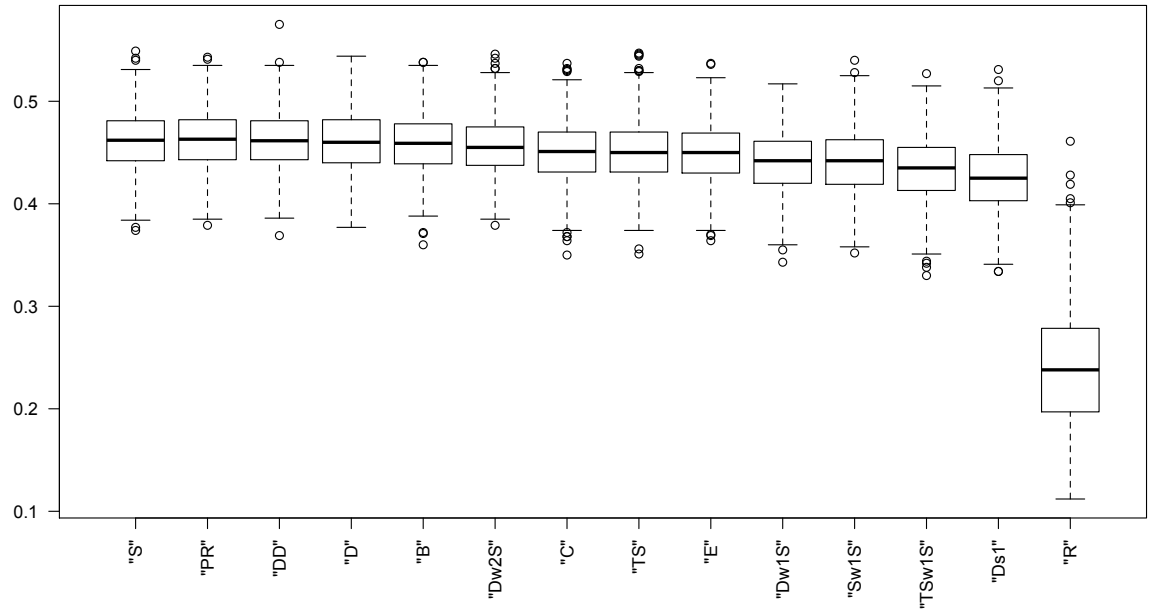

Figure 5.12. Box Plots of Heuristic Performances on Scale-Free Networks with IC Model

5.3.6. Small-World Networks with Independent Cascade Model

As the results are shown in Table 5.10, it can be observed that R is not the worst performing heuristic, similar to when LT diffusion model was used for small-world networks. E, *Ds1*, and C are all outperformed by R, with performances of 13.3%, 14.1%, and 18.1% respectively. The other heuristics with higher performances are very close to R, so it is necessary to conduct a t-test to see if the differences are significant or not.

Comparing R and D, it is seen that the difference between their performance is statistically significant as D performs almost 13% better than R.

Table 5.10. Heuristic Performances on Small-World Networks with IC Model

R	19.7%	**E**	13.3%
D	22.2%	**PR**	22.5%
DD	23.1%	**Dw1S**	21.7%
S	22.6%	**Dw2S**	21.8%
TS	20.5%	**Sw1S**	21.8%
B	21.1%	**TSw1S**	21.1%
C	18.1%	**Ds1**	14.1%

Another important point to check is the significance of difference between group 1 heuristics and their group 2 versions. Conducting the t-tests, although the group 2 versions perform only 2-4% worse, it can be seen that the differences are statistically significant.

Table 5.11. t-Tests for Heuristics on Small-World Networks with IC Model

	R & D	D & Dw1S	S & Sw1S	TS & TSw1S
Mean	0.197 - 0.222	0.222 - 0.217	0.226 - 0.218	0.205 - 0.211
Sample size	1000 - 1000	1000 - 1000	1000 - 1000	1000 - 1000
df	1997.9	1987.3	1995.8	1988.9
t	-19.169	3.6804	5.7963	-4.2249
p-value	0	0.0002391	7.86×10^{-9}	2.499×10^{-5}
H_0	$\mu_1 = \mu_2$	$\mu_1 = \mu_2$	$\mu_1 = \mu_2$	$\mu_1 = \mu_2$
Decision	Reject H_0	Reject H_0	Reject H_0	Reject H_0

Similar to the case in random networks, the nodes in small-world networks are even more homogenous regarding the degree distribution, which is normally distributed with a very low variance. The homogeneity due to the explained network structure and the distribution of link strengths, prevents the heuristics to be distinguished significantly from each other regarding their performances. Still, the t-tests show that the

differences are statistically significant, which means intelligent seed selection heuristics may be useful under different link strength distributions.

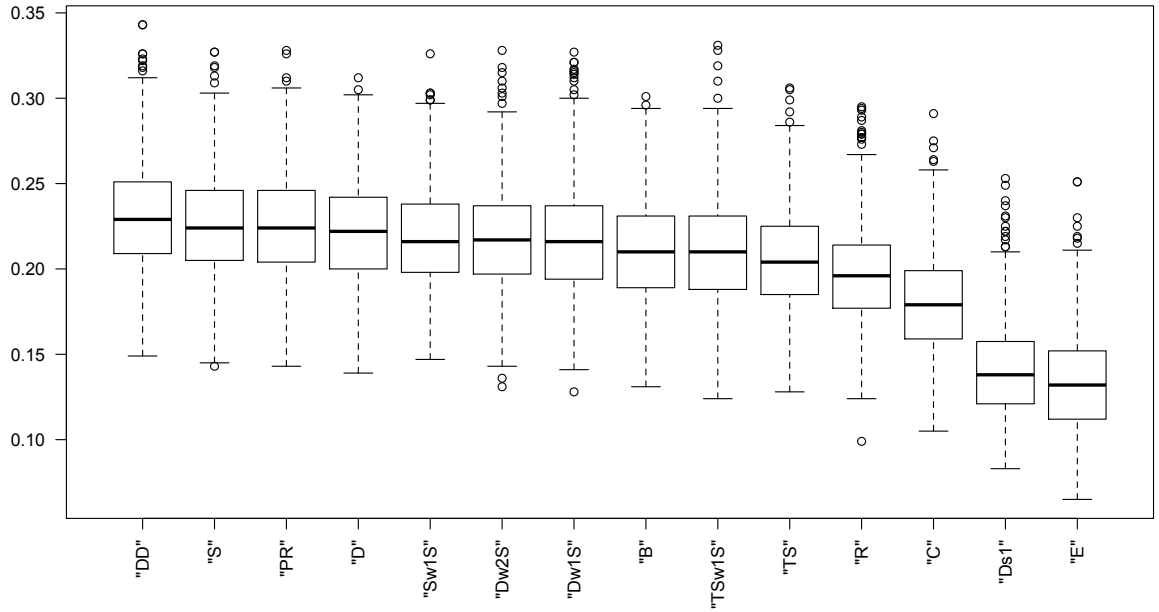

Figure 5.13. Box Plots of Heuristic Performances on Small-World Networks with IC Model

6. CONCLUSION

In this study, the problem of seed selection is investigated with a heuristics approach. The problem is mainly treated with the assumption of complete information about the structure and characteristics of network. In this study, it is aimed to eliminate the need for this assumption by investigating and testing seed selection methods that do no require complete information. The problem of seed selection is primarily treated as an optimization problem, and it is proved to be NP-hard. In the literature, most studies use algorithmic heuristics to come up with near optimal solutions, assuming complete information. While such heuristics have good approximation guarantees, they also have high computational complexity. For example, the simple greedy heuristic had a computation time of more than 10 hours for the networks analyzed. In this study, metric-based heuristics are analyzed, which have low computational costs. There are two groups of heuristics. Group 1 heuristics are the ones which use complete information, and group 2 heuristics rely only on partial and local information. Group 1 heuristics are mainly designed to get an insight on which metrics give promising results and these results are later used for developing group 2 heuristics.

An agent-based social network model is constructed for the study that is capable of creating networks, applying the seed selection depending on the rules of heuristics, and simulating the diffusion processes. Three different types of networks (random, scale-free, small-world) and two different types of diffusion models (Linear Threshold, Independent Cascade) are used, along with several heuristics. The threshold values are assigned using uniform distribution. The link strengths are assumed to be equal for all friends of a node, with a value of $1/degree$.

Prior to primary experiments, a set of simulations are conducted to identify approximate ranges to be used for threshold distributions. These simulations are done for all the three network types separately. A range of uniform threshold distribution for each network type has been choosen in which the heuristics can be distinguished.

Analyzing the threshold distribution ranges, it has been observed that there are ranges where doing intelligent seed selection with a given seed set size is an endeavor in vain. This is because selecting seeds randomly gives the same outcomes with designed heuristics, either resulting in no diffusion or full diffusion. These ranges differ between network types. A range for a network type which helps in distinguishing heuristics may not be working for another network type. For random and scale-free networks the ranges where heuristic performances can be distinguished are wider compared to small-world networks. These ranges are 0.1-0.45 in random networks, 0.4-0.9 in scale-free networks, and 0.55-0.7 in small-world networks. All these indicate that before deciding on a heuristic to find good seed sets, it is important to analyze the structure of a network and decide to which type it resembles. After the decision on the network type, it is also important to check the distribution of thresholds. If they are distributed in such a way that heuristic performances will not be distinguished, than it would be an unnecessary struggle to use heuristics, since a randomly selected seed set will provide similar results. If this is not the case, then one can apply intelligent seed selection heuristics to come up with better performing seed sets.

The experimentation is carried out by simulating each combination of network type and diffusion model. The simulation are conducted with replications and the results for each heuristics are averaged. The comparative analysis of heuristic performances are done for each combination separately.

Overall, the results obtained indicate that although there are some heuristics that constantly have good performances, generally the performances of heuristics depend on the network type. For example, with LT diffusion model, while AT has a very good performance and outperforms D in random networks, it performs worse than D in scale-free and small-world networks. AT is a heuristic that uses the metric average threshold directly. All 10 heuristics that use average threshold as an input has provided promising results, especially when average threshold is combined with the degree metric. In all network types using LT model, a heuristic using this combination had a performance among the best performing heuristics. However, most heuristics had varying performance between networks, which means that before using a heuristic

to find a seed set, it is important to analyze the network in hand, and decide what type of a network it may be.

An observation on random networks using LT model is worth to be discussed. In this experimentation set, when the performances of all heuristics are analyzed, it has been seen that the final diffusions are either between 0.05 and 0.1 or close to 1. This may indicate that there is a tipping point in such networks, which seems to be around 0.1 in this case. Once this tipping point is passed, the diffusion will carry on up to a full diffusion. This observation can also help in explaining why the average threshold metric works great in random networks. Since this metric tries to find nodes who have friends that can be 'convinced' easily, the seed set found helps to give a kick-off to the diffusion. This kick-off helps the diffusion to pass the tipping point, which enhances the performances of heuristics using this metric greatly.

Using IC model, it has been observed that in random and small-world networks, the difference between the performances of random seed selection and other heuristics is very small. In scale-free networks, there is a significant difference but heuristics other than R are not distinguished again. The reason behind this is probably the design of link strengths in the study. Since these are assigned equally for the incoming links of a node, the difference between characteristics of nodes which can help in finding good and bad seeds are minimized. This may be causing the fact that selecting seeds in any way results in more or less the same diffusion. The difference in the situation of scale-free networks is most likely due to its degree distribution. Since this distribution follows a power-law scaling, there are a lot of nodes with low degrees next to very few nodes with very high degrees, which are the hubs. Because R has no logical basis, it might not select such hubs as seeds, whereas all other intelligently designed heuristics mostly select seed sets which include these hubs. These hubs have very high connectivity, and they are likely to increase the diffusion, thus causing all other heuristics to perform significantly higher than R.

The performances of group 2 heuristics, which do not assume complete information on a network are promising. Comparing them to their group 1 versions, it is

observed that most of the time group 2 heuristics are outperformed. However, the difference between group 1 and group 2 heuristics are mostly in a range of 1 to 10%, which can be thought as an acceptable difference since group 2 heuristics are not using the assumption of complete information. There are some cases where group 2 heuristics perform as good as their group 1 versions. Interestingly, there are cases where group 2 heuristics outperform their group 1 versions, such as when it happened in small-world networks using LT model. In this case, the group 2 heuristics that use the average threshold metric have outperformed their group 1 versions by almost 60%. This is an interesting finding. The better performance of group 2 heuristics may be related to the network structure. Small-world networks have high clustering coefficients, which increases the probability of a certain node's neighbors being neighbors with each other. Since the group 1 heuristics select nodes with the best metric among all nodes, they might be selecting nodes from the same cluster. However, the group 2 heuristics reach nodes randomly, which increases their chance to reach different clusters of nodes. Such a difference between two groups may be the reason for the outperformance of group 2 heuristics. This may be indicating that a certain level of randomness may actually be helpful. It can be seen that the relative performance of R for small-world network is not as bad as its performances in scale-free and random networks. It may be the case that, small-world networks can benefit from selecting seeds randomly for a certain portion of their seed set.

One last important observation on the results can be made on heuristics that have performed worse than R in certain cases. For example, in small-world networks for both diffusion models, E, *Ds1*, and C have all performed worse than R. This is an interesting result, since these heuristics, especially E and C, are based on metrics which mainly aim to find influential/central nodes in networks. It might be the case that these metrics are not suitable to find good seeds when the network has small-world characteristics.

At the end, the heuristics performing the best in each combination can be seen in Table 6.1. Heuristics using the average threshold metric as an input are dominating random and small-world networks with LT diffusion model. There are also group 2

heuristics having the top three performances in these combinations. For all the other combinations, *DD*, *S*, and *PR* are seen most among the best performing heuristics.

Table 6.1. Best Performing Heuristics

	Linear Threshold	Independent Cascade
Random	1. ATw5GrwSD 2. ATwSD 3. ATw5GrwSDw1S	1. DD 2. S 3. E
Scale-Free	1. DD 2. S 3. PR	1. S 2. PR 3. DD
Small-World	1. ATwSDw1S 2. ATw5GrwSDw1S 3. ATw1S	1. DD 2. S 3. PR

There are several future research directions. Firstly, the tipping point case in random networks can be studied. If such a tipping point exists, it might be important to find out how it can be discovered. Such a discovery may help in the study of seed selection, by making this tipping point the goal to be reached instead of maximizing the diffusion. Secondly, the boost of performance created by randomness in small-world networks can be investigated. While it can be said that selecting the whole seed set is not the best strategy, there may be a certain level of randomness which enhances the performances of heuristics. Thirdly, an investigation can be done of the distribution of link strengths. Rather than assigning link strengths equally, an analysis similar to node thresholds may be conducted. Such an analysis would give the ranges for link strengths where intelligent seed selection matters and where it doesn't. Finally, the approach of partial information may be developed. In this study, it is applied as collecting information locally from nodes. By doing the same information collection, the links and neighbors of certain nodes can be collected and the obtained subgraph may be reconstructed using intelligent reconstruction algorithms. This way, the performances of heuristics may be increased by supplying them more information on a network.

REFERENCES

1. Kwak, H., C. Lee, H. Park and S. Moon, "What is Twitter, a social network or a news media?", *Proceedings of the 19th international conference on World wide web*, pp. 591–600, 2010.

2. Budak, C. and D. J. Watts, "Dissecting the Spirit of Gezi: Influence vs. selection in the Occupy Gezi movement", *Sociological Science*, 2015.

3. Hilbert, M., J. Vasquez, D. Halpern, S. Valenzuela, and E. Arriagada, "One Step, Two Step, Network Step? Complementary Perspectives on Communication Flows in Twittered Citizen Protests", *Social Science Computer Review*, pp. 1–18, 2016.

4. Domingos, P. and M. Richardson, "Mining the network value of customers", *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 57–66, 2001.

5. Kempe, D., J. Kleinberg and É. Tardos, "Maximizing the spread of influence through a social network", *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 137–146, 2003.

6. Hinz, O., B. Skiera, C. Barrot and J. U. Becker, "Seeding strategies for viral marketing: An empirical comparison", *Journal of Marketing*, Vol. 75, No. 6, pp. 55–71, 2011.

7. Christakis, N. A. and J. H. Fowler, "Social network sensors for early detection of contagious outbreaks", *PloS one*, Vol. 5, No. 9, p. e12948, 2010.

8. Nguyen, N. P., G. Yan, M. T. Thai and S. Eidenbenz, "Containment of misinformation spread in online social networks", *Proceedings of the 4th Annual ACM Web Science Conference*, pp. 213–222, 2012.

9. *Twitter*, 2016, `https://about.twitter.com/company`, accessed at June 2016.

10. Watts, D. J. and S. H. Strogatz, "Collective dynamics of small-world networks", *Nature*, Vol. 393, No. 6684, pp. 440–442, 1998.

11. Newman, M. E., "Random graphs as models of networks", *Handbook of Graphs and Networks: From the Genome to the Internet*, 2002.

12. Golbeck, J., *Analyzing the social web*, Newnes, 2013.

13. Erdős, P. and A. Rényi, "On random graphs I", *Publ. Math. (Debrecen)*, Vol. 6, pp. 290–297, 1959.

14. Milgram, S., "The small world problem", *Psychology today*, Vol. 2, No. 1, pp. 60–67, 1967.

15. Watts, D. J., "Networks, Dynamics, and the Small-World Phenomenon", *American Journal of sociology*, Vol. 105, No. 2, pp. 493–527, 1999.

16. Watts, D. J., *Six degrees: The science of a connected age*, WW Norton & Company, 2004.

17. Kleinberg, J., "The small-world phenomenon: An algorithmic perspective", *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pp. 163–170, 2000.

18. Wang, X. F. and G. Chen, "Complex networks: small-world, scale-free and beyond", *Circuits and Systems Magazine, IEEE*, Vol. 3, No. 1, pp. 6–20, 2003.

19. Barabási, A.-L., *Network science*, Cambridge University Press Cambridge, 2016.

20. Price, D. d. S., "A general theory of bibliometric and other cumulative advantage processes", *Journal of the American society for Information science*, Vol. 27, No. 5, pp. 292–306, 1976.

21. Barabási, A.-L. and R. Albert, "Emergence of scaling in random networks", *science*, Vol. 286, No. 5439, pp. 509–512, 1999.

22. Granovetter, M., "Threshold Models of Collective Behavior", *American journal of sociology*, pp. 1420–1443, 1978.

23. Schelling, T. C., *Micromotives and macrobehavior*, Norton, 1978.

24. Morris, S., "Contagion", *The Review of Economic Studies*, Vol. 67, No. 1, pp. 57–78, 2000.

25. Berger, E., "Dynamic monopolies of constant size", *Journal of Combinatorial Theory, Series B*, Vol. 83, No. 2, pp. 191–200, 2001.

26. Goldenberg, J., B. Libai and E. Muller, "Talk of the network: A complex systems look at the underlying process of word-of-mouth", *Marketing letters*, Vol. 12, No. 3, pp. 211–223, 2001.

27. Goldenberg, J., B. Libai and E. Muller, "Using complex systems analysis to advance marketing theory development: Modeling heterogeneity effects on new product growth through stochastic cellular automata", *Academy of Marketing Science Review*, Vol. 2001, p. 1, 2001.

28. Estevez, P. A., P. Vera and K. Saito, "Selecting the most influential nodes in social networks", *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pp. 2397–2402, 2007.

29. Leskovec, J., A. Krause, C. Guestrin, C. Faloutsos, J. VanBriesen and N. Glance, "Cost-effective outbreak detection in networks", *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 420–429, 2007.

30. Goyal, A., W. Lu and L. V. Lakshmanan, "Celf++: optimizing the greedy al-

gorithm for influence maximization in social networks", *Proceedings of the 20th international conference companion on World wide web*, pp. 47–48, 2011.

31. Goyal, A., W. Lu and L. V. Lakshmanan, "Simpath: An efficient algorithm for influence maximization under the linear threshold model", *Data Mining (ICDM), 2011 IEEE 11th International Conference on*, pp. 211–220, 2011.

32. Kimura, M., K. Saito, R. Nakano and H. Motoda, "Extracting influential nodes on a social network for information diffusion", *Data Mining and Knowledge Discovery*, Vol. 20, No. 1, pp. 70–97, 2010.

33. Jiang, Q., G. Song, G. Cong, Y. Wang, W. Si and K. Xie, "Simulated Annealing Based Influence Maximization in Social Networks.", *AAAI*, Vol. 11, pp. 127–132, 2011.

34. Cheng, S., H. Shen, J. Huang, G. Zhang and X. Cheng, "Staticgreedy: solving the scalability-accuracy dilemma in influence maximization", *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pp. 509–518, 2013.

35. Narayanam, R. and Y. Narahari, "A shapley value-based approach to discover influential nodes in social networks", *Automation Science and Engineering, IEEE Transactions on*, Vol. 8, No. 1, pp. 130–147, 2011.

36. Kimura, M. and K. Saito, "Tractable models for information diffusion in social networks", *Knowledge Discovery in Databases: PKDD 2006*, pp. 259–271, 2006.

37. Kitsak, M., L. K. Gallos, S. Havlin, F. Liljeros, L. Muchnik, H. E. Stanley and H. A. Makse, "Identification of influential spreaders in complex networks", *Nature physics*, Vol. 6, No. 11, pp. 888–893, 2010.

38. Zhang, X., J. Zhu, Q. Wang and H. Zhao, "Identifying influential nodes in complex networks with community structure", *Knowledge-Based Systems*, Vol. 42, pp. 74–

84, 2013.

39. Chen, W., Y. Wang and S. Yang, "Efficient influence maximization in social networks", *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 199–208, 2009.

40. Chen, W., C. Wang and Y. Wang, "Scalable influence maximization for prevalent viral marketing in large-scale social networks", *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 1029–1038, 2010.

41. Freeman, L. C., "Centrality in social networks conceptual clarification", *Social networks*, Vol. 1, No. 3, pp. 215–239, 1978.

42. Friedkin, N. E., "Theoretical foundations for centrality measures", *American journal of Sociology*, pp. 1478–1504, 1991.

43. Sabidussi, G., "The centrality index of a graph", *Psychometrika*, Vol. 31, No. 4, pp. 581–603, 1966.

44. Bonacich, P., "Power and centrality: A family of measures", *American journal of sociology*, pp. 1170–1182, 1987.

45. Page, L., S. Brin, R. Motwani and T. Winograd, "The PageRank citation ranking: bringing order to the web.", *World Wide Web - Internet And Web Information Systems*, 1999.

46. Chen, D., L. Lü, M.-S. Shang, Y.-C. Zhang and T. Zhou, "Identifying influential nodes in complex networks", *Physica a: Statistical mechanics and its applications*, Vol. 391, No. 4, pp. 1777–1787, 2012.

47. Stonedahl, F., W. Rand and U. Wilensky, "Evolving viral marketing strategies", *Proceedings of the 12th annual conference on Genetic and evolutionary computa-*

tion, pp. 1195–1202, 2010.

48. Wilensky, U., "NetLogo", *Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL*, 1999.

49. Huberman, B. A., D. M. Romero and F. Wu, "Social networks that matter: Twitter under the microscope", *Available at SSRN 1313405*, 2008.

50. Wilensky, U., *NetLogo small worlds model*, 2005, `http://ccl.northwestern.edu/netlogo/models/SmallWorlds`, accessed at March 2016.

51. Wilensky, U., *NetLogo preferential attachment model*, 2005, `http://ccl.northwestern. edu/netlogo/models/PreferentialAttachment`, accessed at March 2016.

APPENDIX A: OTHER HEURISTICS

Betweenness over s Nodes within 1 Step (Bs1): s nodes are selected from network G and a subgraph G' is created using these nodes and their links. Using these links also means using the neighbors of each selected node, since links exist with two nodes. Then, the betweenness centrality of each node in G' is calculated, and s nodes in G with the maximum metric in G' is activated.

```
choose random s nodes from network G
create a subgraph G' with the selected nodes and their links
for each node x in G' do
   set v_x ← calculate betweenness centrality in G'
end for
for s nodes with maximum v values from G' do
   activate in G
end for
```

Figure A.1. Pseudocode of Bs1

Betweenness over s Nodes within 1 Step with Graph Modification (Bs1wGM): Similar to $Bs1$, but as a difference the subgraph G' is modified using certain probabilities for each node. After the modification, s nodes with the maximum betweenness centrality in this new subgraph is selected to be activated.

Average Degree (AD): The heuristic uses the metric of mean d of neighbors for a node, and activate s nodes with the maximum metric.

Average Degree within 1 Step (ADw1S): This heuristic will access to an inactive node and select the inactive node with the maximum average degree value within the reach of 1 step to activate. This procedure is repeated s times.

choose random s nodes from network G
create a subgraph G' with the selected nodes and their links
for each node x in G' but not in subset s **do**
  set $m_x \leftarrow$ (count neighbors in subset s) / s
end for
for each pair of nodes y and z in G' but not in subset s **do**
  if no link between y and z **then**
    create a link between y and z in G' with probability $(m_y + m_z)/2$
  end if
end for
for each node x in G' **do**
  set $v_x \leftarrow$ calculate betweenness centrality in G'
end for
for s nodes with maximum v values from G' **do**
  activate in G
end for

Figure A.2. Pseudocode of Bs1wGM

for each node x **do**
  set $v_x \leftarrow$ mean d of neighbors
end for
for s nodes with minimum v values **do**
  activate
end for

Figure A.3. Pseudocode of AD

for each node x **do**
  set $v_x \leftarrow$ mean d of neighbors
end for
for i=1 to s **do**
  select an inactive node x
  for the inactive node within 1 step of node x with the maximum v value **do**
    activate
  end for
end for

Figure A.4. Pseudocode of ADw1S

One Minus Threshold (1mT): The heuristic use the metric for a node as the sum of $1-t$ of its neighbors and activates s nodes with the maximum metric. This heuristic is only applicable to the LT model.

for each node x **do**
  set $v_x \leftarrow$ sum $1-t$ of neighbors
end for
for s nodes with minimum v values **do**
  activate
end for

Figure A.5. Pseudocode of 1mT

Average Threshold with 5 Groups within 2 Steps (ATw5Grw2S): The nodes are grouped according to their threshold values similar to $ATw5Grw1S$. Then, an inactive node is selected and the inactive node with the minimum metric within the reach of 2 steps is activated. This procedure is repeated s times. This heuristic is only applicable to the LT model.

```
for each node x do
  for k=5 to 1 do
    if t_x < 0.2k× (threshold gap) + (lower threshold) then
      set tg_x ← k
    end if
  end for
end for
for each node x do
  set v_x ← mean tg_x of neighbors
end for
for i=1 to s do
  select an inactive node x
  for the inactive node within 2 steps of node x with the maximum v value do
    activate
  end for
end for
```

Figure A.6. Pseudocode of ATw5Grw2S

Average Threshold-Minimum and Maximum (AT-MM): The heuristic first activate some pre-defined β proportion of nodes with minimum average threshold, and the remaining $1-\beta$ proportion of nodes for the seed set is selected from the nodes with maximum average threshold. This heuristic is only applicable to the LT model.

for each node x **do**
  set $v_x \leftarrow$ mean t of neighbors
end for
for βs nodes with minimum v values **do**
  activate
end for
for $(1-\beta)s$ nodes with maximum v values **do**
  activate
end for

Figure A.7. Pseudocode of AT-MM

Average Threshold-Minimum and Random (AT-MR): The heuristic first activate some pre-defined β proportion of nodes with minimum average threshold, and the remaining $1-\beta$ proportion of nodes for the seed set is selected randomly. This heuristic is only applicable to the LT model.

for each node x **do**

  set $v_x \leftarrow$ mean t of neighbors

end for

for βs nodes with minimum v values **do**

  activate

end for

for randomly selected $(1-\beta)s$ nodes **do**

  activate

end for

Figure A.8. Pseudocode of AT-MR