universität wien

MASTERARBEIT / MASTER’S THESIS

Titel der Masterarbeit / Title of the Master‘s Thesis

„The Target Set Selection Problem with Arbitrary Edge Weights“

verfasst von / submitted by

Ludwig Michael Müller BSc

angestrebter akademischer Grad / in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien, 2018 / Vienna 2018

Studienkennzahl lt. Studienblatt / degree programme code as it appears on the student record sheet:	A 066 914
Studienrichtung lt. Studienblatt / degree programme as it appears on the student record sheet:	Masterstudium Internationale Betriebswirtschaft
Betreut von / Supervisor:	Dipl.-Ing. Dr. Markus Leitner, Privatdoz.
Mitbetreut von / Co-Supervisor:	Dipl.-Ing. Dr. Mario Ruthmair

Statutory Declaration

I declare that I have authored this thesis independently, that I have not used other than the declared sources/ resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Munich, 22nd of August 2018

Abstract

This master's thesis reasons why the Target Set Selection Problem (TSSP) is of high relevance to research topics of gaining importance. The impact of graph properties on the complexity of solving the TSSP is shown. For the case of graphs with arbitrary edge weights, a novel integer linear programming formulation is established, and it is shown where the pitfalls of finding target sets in graphs with arbitrary edge weights are. For instances with different structural features, as well as for real-world data, solutions have been found and analyzed. One of the findings is, that even more than the number of nodes, a high graph density leads to a great increase in calculation time for determining the solution of the TSSP.

Abstract

Diese Masterarbeit erörtert, warum das Target Set Selection Problem (TSSP) für Forschungsfelder, die an Bedeutung zunehmen, von großer Wichtigkeit ist. Der Einfluss von Eigenschaften von Graphen auf die Komplexität der Lösung des TSSP wird gezeigt. Für den Fall von Graphen mit beliebigen Kantengewichten wird eine neue Formulierung der ganzzahligen linearen Optimierung eingeführt und gezeigt, wo die Tücken beim Finden eines Target Sets im Falle von Graphen mit beliebigen Kantengewichten sind. Für Instanzen mit unterschiedlichen strukturellen Eigenschaften, sowie für reale Datensätze, wurden Lösungen des TSSP gefunden und analysiert. Eine Erkenntnis ist, dass eine hohe Dichte des Graphen, mehr noch als die Anzahl der Knoten, zu einer hohen Rechenzeit zur Ermittlung der Lösung des TSSP führt.

Contents

1 Introduction

In a increasingly interconnected world, understanding spreading behavior, e.g. the propagation of information or in the spread of diseases is gaining importance. Two global trends are visible in today's world: An increased interconnectedness between people, institutions and devices and greater availability of data on this interconnectivity. The former trend makes spreading phenomena on networks a relevant topic. The latter provides the possibility to do research on this topic using real-world data: As Chen et al. (2013) note in their survey on Information and Influence Propagation in Social Networks, "[e]nabled by the Internet and sparked by the recent advent of online social networking sites such as Facebook, LinkedIn and Tumblr, research on social networks is witnessing an unprecedented growth due to the ready availability of large scale social network data. This has at once led to the development of many exciting applications of online social networks and to the formulation and the subsequent study of many research questions."

One such research question is to find the minimal number of individuals in a network, such that at least a predefined number of individuals will be affected by a propagation process from these initial individuals. This problem, the so-called Target Set Selection Problem (TSSP), is the main topic of this thesis.

This thesis extends the literature on the TSSP mainly influenced by Nichterlein et al. (2013), Ben-Zwi et al. (2011) and Chen et al. (2013) by addressing the case of negative influence between individuals. More precisely, an integer linear programming formulation for finding the target set of networks with edges of arbitrary edge weight is proposed and some of the features of solved instances are being examined.

2 The Relevance of the Target Set Selection Problem

The following subsections will discuss some fields in which the TSSP is particularly relevant.

2.1 Mass Psychology

When considering spreading phenomena in networks, the network topology plays a crucial role: In weakly connected networks, individual changes will not lead to great effects on the overall network. When however the network is strongly interconnected, small changes for the individual can lead to large effects on the large scale. In his work "The Tipping Point", Gladwell (2006) describes how small changes can have a large effect:

> "The tipping point is that magic moment when an idea, trend, or social behavior crosses a threshold, tips, and spreads like wildfire."

One example Gladwell (2006) adduces is the hypothesis, that the introduction of stricter sanctions of the New York authorities against minor offenses like vandalism or evasion of paying the fare resulted in a significantly lower crime rate. Gladwell reasons that these kinds of unlawful behavior, albeit being bagatelles, shape the urban image and lower the psychological threshold of unlawful behavior of citizens. Removing these small but influential acts against the public order, according to Gladwell (2006), increased the psychological threshold of law infringements for some people. The absence of law infringements in turn increased the psychological threshold for more serious crimes, creating a self-enforcing propagation of lawfulness. Gladwell argues that this effect of self-enforcement due to propagation can be harnessed, and emphasizes the importance of the correct starting point in the network: "Economists often talk about the 80/20 principle, which is the idea

that in any situation roughly 80 percent of the work will be done by 20 percent of the participants. In most societies, 20 percent of criminals commit 80 percent of crimes. Twenty percent of motorists cause 80 percent of all accidents. Twenty percent of beer drinkers drink 80 percent of all beer. When it comes to epidemics, though, this disproportionality becomes even more extreme: a tiny percentage of people do the majority of the work."
Spencer and Howarth (2013) establish a moral-motivation model in which the explained effect of propagation of social norms is applied to the adoption of green behavior. They argue that shifts to permanently different mass behavior could be achieved by incentivizing a small minority of the population. How to find this set of individuals (given perfect information about the structure of the social network) is the main topic of this thesis.

2.2 Defense against Terrorism

The need for deeper understanding of the impact of propagation effects in networks has also been recognized as crucial in matters of state security. In the primer "Network Science" published by the National Research Council (2005), co-authored by the Army Science Board, the importance of network science is emphasized:

"A gap exists between currently available knowledge about networks and the knowledge required to characterize, design, and operate the complex global physical, information, biological, and social networks on which the well-being of our citizens has come to depend. Closing this gap is an urgent matter because society has become dependent on the reliable, robust operation of complex global communication, information, transportation, power, and business networks. The disruption or exploitation of these networks by adversarial social networks of terrorists or criminals is a demonstrated threat, making an investment in network science not only strategically sound but also politically urgent."

A potential terrorist attack might be based on the fact that one single malfunction of a component in a connected system can induce a cascade that leads to the malfunction of the entire system. Barabási and Frangos (2014) points out, that systems which are comparatively safe under the circumstance of randomly failing parts can turn out to be very fragile in the case of targeted attacks.

As Cohen et al. (2001) argues, one possible scenario would be the attack against a small number of routers in order to induce a breakdown of huge parts of the Internet.

In order to improve the attack-stability of systems, their vulnerability has to be evaluated. Analyzing ways of finding the ”weak spot” of a system is one step towards that goal. Similar considerations have been published by Asavathiratham et al. (2001) regarding cascading failure in power networks and by Tanizawa et al. (2005) regarding the robustness of the internet. Finding the weak spots in such systems and what role network structure plays in the susceptibility to attack are questions where insights can be gathered in analyzing the TSSP.

2.2.1 Spread of Diseases

Special importance has been attributed to the study on the relationship of network topology and the spread of diseases.

As Shirley and Rushton (2005) ascertain, diseases are spreading quicker in networks that possess a scale-free degree distribution than in locally clustered networks.

Tatem et al. (2006) draw the conclusion that the emerging international interconnectedness through global transport networks contributes to the fact that the network, on which infectious diseases can spread, are adopting scale-free properties.

Other applications within this realm were published by Bearman et al. (2004) examining efficient ways in combating the transmission of sexually transmitted diseases and Dezső and Barabási (2002) examining vaccination strategies.

2.3 Peer-to-Peer Marketing

A commonly used example of a propagation process is a sales promotion relying on consumers recommending the promoted product to their peers. Here, the aim is to find the minimal number of people to incentivize to buy a product and recommend the product to their peers, which in turn will buy and further recommend the product.

The impact of each possible combination of selected initial individuals on the number of consumers for which at least one peer will recommend the product depends on the social network topology. Furthermore, persuading initial individuals to buy the product comes at a cost, therefore there exists reason to find the optimal set of initial costumers.

The existing TSSP model assumes that a buying decision of one individual will have a positive effect on the likeliness that the individual's acquaintances will buy the product as well. As the opposite case is easily conceivable, a generalization in which both positive and negative influence can be exerted between individuals represents a closer approximation to certain cases in the real world. This thesis proposes an ILP formulation for such a generalized model and applies it to several networks instances.

2.4 Financial Contagion

Haldane and May (2011) applies the notion of cascading failure to failure of financial institutions which are interconnected by credit relationships: The interbank market, in which banks engage in credit contracts with each other, is an example of a network where the failure of a small part of the network may lead to the failure of the entire system. The credit relationship between two banks can be understood as a relation between two actors, where the financial stability of one actor is positively related to the financial stability of the other player. The hypothesis of a strictly positive relation might not stand for all kinds of relation. For instance, between two firms competing in the same market, the financial stability of one firm might be negatively related to the financial stability of the other.

3 Preliminaries

This section will introduce concepts which will be needed in later sections.

3.1 Graph Theory Terminology

A network of individuals that stand in relation to each other can be represented by a **graph** $G(V, E)$. Each **node** $v \in V$ represents an individual and two nodes $u, v \in V$ are connected by an **edge** $\{u, v\} \in E$. if there exists a relation between them.

If the assumption of mutual relations is given, the edges are undirected.

If relations may be unilateral, a relation is represented via a directed graph $G(V, A)$, in which an influence of $u \in V$ on $v \in V$ is represented by an arc $\{u, v\} \in A$.

The amount of edges a node u possesses ($\sum_v \{u, v\} + \sum_v \{v, u\}$) is called the **degree** of node u. In case of a graph with directed edges, there exist also **indegree** ($\sum_v \{v, u\}$) and **outdegree** ($\sum_v \{u, v\}$) of a node.

A path is a sequence of nodes $v_1, v_2, ..., v_k$ such that between every pair of adjacent nodes there exists an edge $\{v_1, v_2\}, \{v_1, v_2\}, ..., \{v_{k-1}, v_k\}$ and no node repeats itself ($v_i \neq v_j \forall i \neq j$). The **distance** between two nodes $d(u, v)$ is the shortest possible path between two certain nodes.

The **diameter** of a graph $diam(g)$ is the length of the longest distance existing in a graph G for all sets of $u, v \in V$. The network structure can have a huge influence in the diameter of a network: In the case of a network consisting of a chain of nodes, each two nodes connected by one edge, the diameter of a graph equals the amount of edges in the graph. In the case of fully connected graphs, the diameter equals one.

A **closed triplet** is formed by three nodes u, v, w, which are connected by three edges $\{u, v\}, \{v, w\}, \{w, u\}$. The three nodes of an **open triplet** are connected by only two nodes $\{u, v\}, \{v, w\}$.

The **Clustering Coefficient** C is the amount of all closed triplets divided by the amount of all triplets in a graph G. For instance, a full graph (a graph with edges between each pair of nodes) has a clustering coefficient of $C = 1$, a tree has a clustering coefficient of $C = 0$.

A **Connected Component** of a graph is a subset of nodes, of which each node shared an edge with at least one other node in the subset, but no node in the subset shares an edge with a node outside of the subset.

3.2 The Small-World Property

A central publication in the analysis of path lengths of acquaintance networks is the article ”The Small World Problem” by Milgram (1967): The article describes an experiment, in which the participants needed to send a letter to an acquaintance who was to forward it in turn to reach a predefined recipient. Milgram claims that the average number of forwardings per letter was about six.

This raises the question how it comes to be that in a network of several hundred million nodes (the population of the United States) the average path length seems to be just equal to six. Here, it is worthwhile to think about the structure of networks.

Many real-world networks display a combination of high clustering and small average path length. The networks which represent this combination have been characterized as 'Small World Networks' by Watts (2004).

Watts defines these networks as scale-free, this means that the probability of a given node having k edges $P(k)$ exhibits a power law distribution $P(k) \sim k^{-\gamma}$. As a result, while most nodes have a very low degree, a small fraction possesses a very high degree. The average degree in this kind of networks is far higher than the degree of the vast majority of nodes.

3.3 Random Graphs

In static random graph models like the Erdös-Rényi model by Erdös and Rényi (1959) or the network generation algorithm by Molloy and Reed (1995), a graph is created by repeatedly selecting pairs of vertices with a static probability and creating an edge between them. In the Erdös-Rényi Model, the degree distribution can be approximated by a Poisson distribution.

Molloy and Reed (1995) also describe a procedure to generate Random networks with an arbitrary degree distribution.

Erdös and Rényi (1959) introduce the concept of a **Giant Component**: Taking a unconnected random graph, and adding one edge between two random nodes at a time, the number of connected components will gradually decrease from n to 1. What remains striking is that as a rule, when the average degree surpasses 1, one connected component emerges that is much greater than the rest of the components. Erdös and Rényi call this the giant component.

Yet this does not provide any explanation of the perceived short average shortest path between two nodes, or the high local clustering in real networks.

As Barabási and Pósfai (2016) states, in a random network, for a node with a fixed degree, when increasing the size of the network, the clustering coefficient decreases significantly.

Barabási and Pósfai (2016) states that for random networks of a small node count and a fixed number of edges per node, the clustering coefficient is high, because the chance is high that for any node, two of its neighbors will be connected. Random networks of larger node count and the same number of edges per node, the clustering coefficient will be lower, since the probability that for any node, two of its neighbors will be connected, is sinking with increasing size of the network.

This is easy to illustrate: When starting off with three nodes, and an average degree of two, the clustering coefficient equals one, since every triplet is closed. When stepwise adding one node and two edges to the network, the probability of forming a new triplet in each step will continually get smaller. With adding nodes ad infinitum, the clustering coefficient will approximate zero.
The network obtained by this process will bear a low number of triplets, obviously different to many real-world networks such as the network of acquaintances, where it is common that if two individuals both are acquainted to the same third individual, they are also acquainted to each other.

3.4 Fitness Models

The assumption, that the probability of two vertices share an edge is independent of the node's features (i.e. its degree) seems to be somewhat incompatible with reality. For instance, one might think of a graph with nodes representing the world's population, which are joined by an edge if they know each other by name. In this case, determining the chances of a certain individual to know any other simply on a matter of the degrees of each of the two seems far from reality. Factors like the distance between the homes of the individuals play a huge role, as everyday experience shows. Using a model that takes into account intrinsic properties of vertices which have an effect regarding the question to which node they are connected to, Söderberg (2002) defines a certain value for each vertex that assigns it to a type. The probability of two vertices being connected depends on their 'fitness', e.g. the type of the two vertices. Sonderberg shows that using a scale-free fitness distribution together with a non-scale free degree distribution can result in a scale-free network. This suggests that properties inherent to the vertices can play a crucial role in the overall structure of the networks.

3.5 The Watts-Strogatz Model

In the Small World Model by Watts and Strogatz (1998), the construction of the network starts off with a so-called ring lattice with n vertices and k edges per vertex. Hereby all vertices are arranged in a ring and every vertex is connected to its k closest neighbors. Depending on a given rewiring probability p, each edge has the chance to change one of its ends to a different node. This construction allows to vary the 'amount of randomness' of a graph between $p = 0$ and $p = 1$, where the former represents the initial ring lattice, and the latter represents a random network.

Watts and Strogatz (1998) analyse the value of the normalized clustering coefficient and the average path length for graphs with varying rewiring probability:

When starting with a graph with low rewiring probability and gradually increasing it, the initially very high local clustering of nodes decreases with a steady rate. The average path length however reduces significantly already with the first changed edges. That means a moderate rewiring probability does not have a great effect of the local clustering of the graph but leads to far shorter average path lengths than in a ring lattice.

These features are what we were looking for at the beginning of the chapter: A network with high local clustering, similar to the network of acquaintances of the participants of Milgram's study, but at the same time a short average path length, similar to the short chain of letter forwardings.

4 Diffusion Models

In this section different types of diffsion models will be presented.

4.1 Value Types

Diffusion describes the effect of the spread of one node's value to nodes connected to it. These values may be categorical, discrete or continuous. One example of a diffusion model with a categorical value is the model of dissemination of culture by Axelrod (1997), in which culture is being represented by a combination of traits (such as language, religion, etc.). In the model, individuals are represented by nodes on a grid, each connected to its four neighbors. Each node has a vector of values, which represent the traits of the individual. If two neighbouring individuals share any common trait, with a certain probability, one of the not-shared feature values of one of the two individuals replaces the corresponding feature value of the other individual. After a certain number of rounds, each individual can have only neighbors of two kinds: Individuals they do not share any trait with, and individuals which share all of their traits. A group of neighbouring nodes with identical traits are called a zone. Axelrod simulates the effect of different levels in the number of individuals, traits and influence distances on the number of zones in the final state. In an alternative model, Axelrod proposes continuous traits, enabling influence between all nodes. Since in this variant, all nodes are subject to the assimilation process, each node has the identical traits in the final state of the process.

Henceforth, this thesis restrict itself to diffusion models of one single value, which can take one of the two states *activated* and *not activated*.

4.2 Stochastic vs. Deterministic Models

Diffusion models can follow deterministic or stochastic rules. An example of a deterministic model is the model of influence spread by Richardson and Domingos (2002), in which a node is being activated with certainty as soon as the number of active neighbors exceed the threshold of the node.

An example where spreading behavior on networks that does not follow deterministic rules is found is in Percolation Theory by Bollobás and Riordan (2006). Here, a certain probability is being assigned to nodes or edges which determines the probability that it exerts influence.

4.3 Progressive vs. Non-Progressive Models

In progressive models, a once acquired state does not switch back, in non-progressive models, nodes can change their state back and forth.
As will be discussed later, non-progressive diffusion models can lead to difficulties determining the minimum-size target set on graphs with arbitrary edge weight. Since activated nodes can be deactivated later on, the activation process cannot be easily depicted by a non-cyclic diffusion graph for every conceivable instance, since edges might be activated, deactivated, and reactivated again. Furthermore, there exist instances where no steady state is reached: For example, a pair of nodes might activate and deactivate each other in turn ad infinitum. With progressive models however, it is obvious that a diffusion process on a graph of n vertices must have reached a steady state after at most n time steps: The activation process, depicted in a diffusion tree cannot be longer than the longest path of the underlying graph, which is bound by the node count of the graph.

4.4 The Independent Cascade Model

Kempe et al. (2003) provide a definition of the **Independent Cascade Model**:

In the graph $G(V, E)$, a uniform value $p_{v,w}$ is assigned to every edge $p_{v,w}, \forall\{v, w\} \in E$, which determines the probability that node w gets activated by node v in the round after node v has been activated. The process stops when no new nodes have been added in a step.

Granovetter (1978) states, that the Independent Cascade Model is submodular and monotonic. The notion of **Submodularity** states that for any given set of x initially active nodes which have been selected, adding a certain node i to the initial set cannot increase the size of the set activated by the propagation process by more nodes than adding this same node to the set of $n + 1$ initially active selected nodes.

Monotonicity states that for any node, each additional neighbour that is becoming activated, contributes to the node itself getting activated: Let us assume that the function $g_v(\cdot)$ determines if a node v gets activated by its set of neighbors $N(v)$ at a certain time step. Further, $X \in N(v)$ and $Y \in N(v)$ are two possible sets of active neighbors for node v where $X < Y$. If the function $g_v(\cdot)$ is monotone, $g_v(X)$ must never be greater than $g_v(Y)$ for any node v or sets of neighbors X and Y (given $X < Y$).

Submodularity and monotonicity together constitute a feature that can be exploited in the design of algorithms. When establishing the WTSSP with arbitrary edge weights later on, which does not inhibit these two features, it will be shown that a lack of these features constitute a challenge in establishing a solution method to finding the optimal target set.

4.5 The Threshold Model

Kempe et al. (2003) provide a definition of the **Threshold Model**:

In a graph $G(V, E)$, each node $v \in V$ is can share an edge $\{v, w\}$ with another node $w \forall v \; neqw$.

All the nodes connecting v to its neighbors exhibit a certain weight $b_{v,w}$, such that $\sum_w b_{v,w} \leq 1$.

Each node exhibits a threshold θ_v between 0 and 1, which defines the weighted fraction of v's neighbors that have to be active in order to activate v. The propagation process happens over a sequence of time steps. At the initial time step t_0, a set of nodes A_0 is active, all other nodes are inactive. At each time step, all inactive nodes that have a weighted level of active neighbors larger than the their own threshold level θ_v are being activated. This is repeated until the active set does not grow anymore.

As does the Independent Cascade Model, the Linear Threshold Model shows the properties of submodularity and monotonicity.

4.6 The Voter Model

The **Voter Model** is an example of a non-progressive model. This means that once a node has been activated, it can be deactivated again. In the Voter Model, each round every node adapts the state of one of its neighbors. This has the effect that a node's state can switch back and forth. An influence diffusion process following the Voter Model is being described by Even-Dar and Shapira (2007). A generalized model is provided by Pathak et al. (2010). This model is being extended to a friend-and-foe-model by Li et al. (2013).

4.7 The Relevance of Topology for finding a set of influential nodes

As has been discussed in the section about Diffusion Models, the topology of a network affects features such as the average path length. This chapter explains the relevance of network features to the question of finding optimal node subsets for propagation processes. The relevance of network topology for the Target Set Selection Problem - which in detail will be defined later on - can be demonstrated more easily by a related problem, the Vertex Cover Problem (VCP).

The objective of the VCP is to find a set of nodes X from a graph $G(V, E)$ such that every node of the graph is either in X or shares an edge with a node in X.

Vázquez and Weigt (2003) show the connection between degree correlations and the VCP. The authors demonstrate that the higher the degree correlation in a network is, the poorer results will be obtained by some heuristic solution approaches.
In the following, the terms 'Assortativity' and 'Disassortativity' will be used. A graph is described as assortative, if the degrees of its nodes are correlated, i.e when the average degree of neighbors of a node with a high degree is higher than the average degree in the entire graph, and the average degree of nodes neighboring a node with a low degree is lower than the average degree in the entire graph. If no such relation can be observed, a network will be defined as unassortative. A network, in which the average degree of nodes neighboring a node with a high degree is below the average degree in the entire graph, et vice versa, is defined as disassortative.

To illustrate Vázquez and Weight's point, we can imagine two networks of identical node and edge count, but with a different degree correlation. In both cases, a heuristic is being used, which starts with an empty subset and then adds the node which possesses the currently highest degree of the remaining graph into the subset. When there remains no edge, which does not neighbor a node that is in the subset, the heuristic stops and has found a feasible solution.

To exemplify, let us look at two graphs in figure 1a and 1b. Both graphs consist of 7 nodes and 6 edges, thus they have an average degree of $6 \cdot 2/7 \approx 1.7$. Yet they have a different topology, in the graph in figure 1a, for every node, the average degree of its neighbors is equal to 2. The network therefore is neither assortative nor disassortative.

In the graph in figure 1b, nodes with a low degree are connected to nodes with a high degree, et vice versa. Following the reasoning of Vázquez and

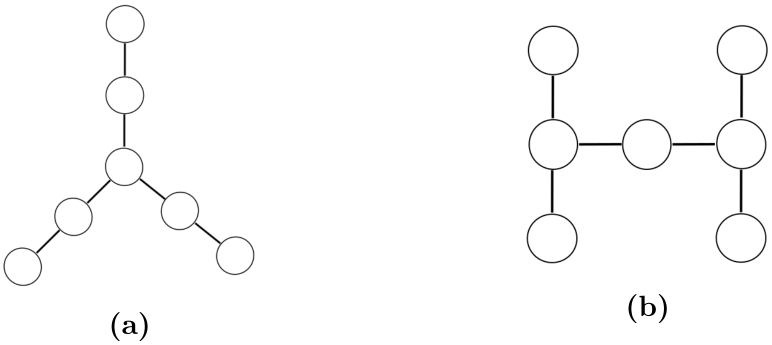


Figure 1: (a) An unassortative graph (b) A disassortative graph

Weight, the heuristic's solution should obtain a result closer to the optimum for the disassortative graph than for the neutral graph.

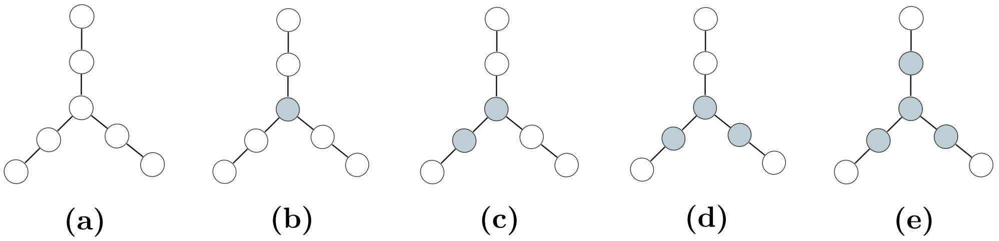


Figure 2: Steps of the heuristic on the unassortative graph

Figure 2 shows from left to right the steps of the heuristic. We start with an empty subset (fig. 2a). In the first step (fig. 2b), the node with the highest degree is being put into the subset. After this, each of the adjacent nodes have still one neighbors in the remaining graph, and they must be put into the subset step by step as well (fig. 2c - 2e). After that, each edge has at least one adjoining node in the subset.

We can see easily here, that the solution is feasible but not optimal. The solution would also be valid without the central node being in the subset of selected nodes.

In the case of the disassortative graph, first one of the two nodes with a

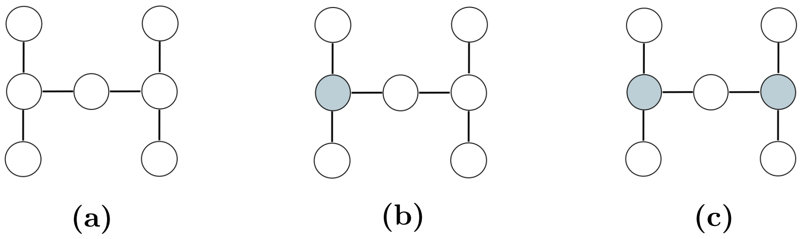


Figure 3: Steps of the heuristic on the disassortative graph

degree of 3 is being picked for the target set (fig. 3b). Then the remaining node of degree 3 is being picked (fig. 3c). After that every edge is adjacent to a node in the subset. It is evident that the solution is not only feasible but optimal.

This example illustrates the conclusion drawn from experiments by Vázquez and Weigt (2003): Heuristics based on the degree of nodes are less promising for finding a solution to the VCP on assortative networks.
This is relevant for applying the related TSSP to graph instances from social network, since this kind of networks generally are assortative networks, as stated in Barabási and Pósfai (2016).

Experiments on the effectiveness of heuristics based on degrees and centrality for the TSSP have been conducted by Kempe et al. (2003): Using a propagation process following the independent cascade model and an infection probability of 10%, for instances with a relatively large optimal target set degree- and centrality-based heuristics provide worse approximations than picking nodes randomly. The author argues that for increasing target set sizes the nodes connected to weakly connected nodes gain importance. These nodes seem to be undervalued in degree- and centrality-based algorithms.

Barabási and Frangos (2014) observes, that in networks with a high aver-

age degree, single nodes are unlikely to infect their neighbors, hindering them from starting propagation process affecting a larger part of the network. He calls these networks **Subcritical Regimes**. Networks, in which single nodes can trigger a global cascade are termed **Supercritical Regimes**.
The effect the average degree has on the number of nodes necessary to induce a cascade affecting the entire network will be one of the topics in the experimentations part of this thesis.

A similar observation is being made by Peleg (2002): Peleg illustrates the impact of network structure on the spread of influence: Assuming a simple majority threshold and no weights with progressive activation (and no deactivation), there exist instances in which a target set of only two nodes suffices to activate the remainder of the entire network, regardless of its size. One such instance is a network where all initially inactive nodes possess only two edges, one to each of the nodes of the target set. Using a model of non-progressive propagation, meaning that nodes will be deactivated with the ratio of their active neighbors being under the nodes' threshold, a minimal target set of $2\sqrt{n}$ suffices to activate the entire network in the most extreme case.

5 The Target Set Selection Problem

Richardson and Domingos (2002) define the **Influence Maximization Problem**, in which a subset of nodes from a Markov random field is active at the start of the propagation process. This node subset is called the **Target Set**. In the propagation process, active nodes exert influence on their neighboring nodes. For any node, when the influence received by its neighbors exceeds its threshold, the node becomes active and exerts influence on its neighbors in turn. The Influence Maximization Problem looks for the best combination of nodes forming a target set of given size that leads to the maximal number of nodes being active after the cascade. In the Target Set Selection Problem defined by Kempe et al. (2003) the target set is not bound, but there exists a lower limit l for the nodes to be activated at the and of the propagation process.

The following subsections illustrate the propagation process and present ILP formulations for different variants of the TSSP.

5.1 Illustration of the Propagation Process

The threshold of a vertex will be depicted by the number on the upper left of each node. The direction of edges on which influence can be exerted from one node to another is being represented by an arrow, edge weights are being depicted by a number next to the edge.

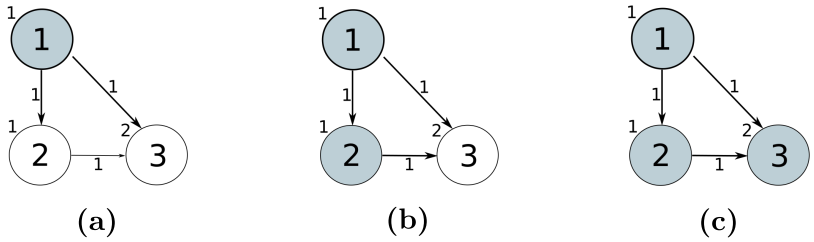
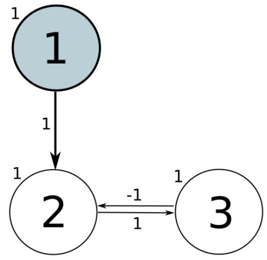


Figure 4: Propagation steps

In figure 4a, a cascade on a network with strictly positive edge weights is shown. The figure shows the graph as the start at the first time step. Node 1 is in the target set and is therefore already active. The threshold of a node is depicted by the number on the upper left of each node. The direction of edges on which influence can be exerted from one node to another is being represented by an arrow, edge weights are being depicted by a number next to the edge.

In the next activation round (fig. 4b), node 2 is activated, since the in-

fluence received from its active neighbor ($d_{1,2} = 1$) is larger or equal to its threshold ($h_2 = 1$). Node 3 is not being activated, since its received influence ($d_{1,3} = 1$) is smaller than its threshold ($h_3 = 2$) and therefore not sufficient to activate it. As a result, the edge from node 2 to node 3 gets activated.
In the next activation round (fig. 64c, node three gets activated, since now the sum of the weights of the active incoming edges ($d_{1,3} + d_{2,3} = 2$) is larger or equal to its threshold ($h_3 = 2$).
Here the minimal-size target set for activation of the entire graph is straightforward. In the case of larger graphs, finding the optimal target set becomes a less trivial task. Another factor that complicates finding the optimal target set is the range of allowed weights of the edge weights. When allowing edge weights to be negative, the order of activation must be taken into consideration, as will be shown later on.



(a)

Figure 5: A graph with a negative edge

The graph in figure 5 the first round of a correct solution to the target set. Even though the sum of the weight of all incoming edges of node 2 is equal to zero and therefore less than its threshold of 1, the node will be active. In the cascade, node 3 is activated after node 2 and therefore can exert influence only after node 2 has been activated. Once activated nodes stay active, even when their activation condition does not hold anymore. Therefore the set of node 1 is a valid target set for this graph.

5.2 WTSS with Node-specific Costs

Zhang and Sahin (2014) propose an ILP-formulation with node-specific costs for selection for the target set:

A graph $G(V, E)$ consists of a set V of nodes numbered by the index i. Nodes may be connected. The subset of vertices that are connected to a vertex i are called i's neighbors. The activation is being depicted by the variable y_{ij}, which is 1 if vertex i influences its neighbour j in the cascading process. Nodes are weighted by a factor b_i which quantifies the cost of having them in the target set. Every vertex inhibits a constant g_i, which denotes the number of neighbors to be active in order to activate i. If a vertex is being selected to be in the target set, its variable x_i is set to 1.

$$\begin{aligned} & \min \sum_{i=1}^{I \in V} b_i x_i && && (1) \\ \text{s.t. } & \sum_j y_{ji} + g_i x_i \geq g_i && \forall i \in V && (2) \\ & \sum_{(i,j) \in C} y_{ij} \leq |C| - 1 && \forall \text{cycles } C \subseteq E && (3) \\ & x_i \in \{0, 1\} && \forall i && (4) \\ & y_{ij}, y_{ji} \in \{0, 1\} && \forall i, j && (5) \end{aligned}$$

The target function (1) sets the objective to finding the least costly target set. Constraint (2) makes sure that vertices only can be activated by either being in the target set, and/or by an amount of neighbors greater than i's threshold g_i already being active. In order to prohibit circular processes, in which a vertex a is activated by another vertex b directly or indirectly being activated by vertex a, constraint 3) is needed. In the case of an undirected graph, each edge must be seen as the smallest possible circle.

5.3 WTSS with Subsidies

Spencer and Howarth (2013) propose an ILP-formulation for a similar model. Here, an node can be 'nudged' towards joining the target set by paying a subsidy y_i:

The binary variable x_{i^t} denotes whether an actor i is adopting the behavior in time step t. A node becomes active either when it is in the target set or when a fraction of its neighbors of at least b_i are active. The size q of the initially active set is to be minimized:

$$\min q \tag{6}$$

$$\text{s.t.} \sum_{I \in V} y_i \leq q \tag{7}$$

$$x_{it} \leq y_i + (1/b_i) \sum_{j \in \delta(i)} x_{j,t-1} \qquad \forall i \in V, t \in \{0, 1, ...1|V|\} \tag{8}$$

For the amount of time steps the trivial upper bound of the number of nodes of the graph is being used. The following constraints guarantee that only yet active nodes i cause their neighbors j to become active:

$$x_{it} \leq (1/b_i) \sum_{j \in \delta(i)} x_{j,t-1} \qquad \forall t \in \{0, 1, ...1|V|\} \tag{9}$$

$$x_{i,|V|} \geq 1 \qquad \forall i \in V \tag{10}$$

5.4 Least Cost Influence Problem

Instead of having to pay a fixed value to an individual for being in the target set, a low amount might be enough if the individual is subject to some level of influence. Thus, in this extension we allow the cost for assigning a vertex to the target set be an incentive p_i whose optimal amount has to be determined. Like previously, each vertex inhibits a threshold b_i to be reached in order to activate the vertex. Each vertex is influenced by an active neighbor by the factor d_{ij}.

For the Least Cost Influence Problem, Fischetti et al. propose the following ILP-formulation:

$$
\begin{aligned}
\min & \sum_{i \in V} \sum_{p \in P_i} w_{ip} x_{ip} && && (11)\\
\text{s.t.} & \sum_{p \in P_i} p x_{ip} + \sum_{(j,i) \in A} d_{ji} z_{ji} \geq h_i x_i && \forall i \in V && (12)\\
& \sum_{p \in P_i} x_{ip} = x_i && \forall i \in V && (13)\\
& \sum_{(i,j) \in C} z_{ij} = \sum_{i \in V(C) \setminus \{k\}} x_i && \forall k \in V(C), \forall \text{ cycles } C \subseteq A && (14)\\
& z_{ij} \leq x_i && \forall (i,j) \in A \text{ s.t. } (j,i) \notin A && (15)\\
& \sum_{i \in V} x_i \geq \lceil \alpha |V| \rceil && && (16)\\
& x_{ip} \in \{0,1\} && \forall i \in V, \forall p \in P && (17)\\
& x_i \in \{0,1\} && \forall i \in V && (18)\\
& z_{ij} \in \{0,1\} && \forall i,j \in A && (19)
\end{aligned}
$$

6 Solution Approaches for the TSSP

In this section, a brief overview of solution approaches to existing variants of the TSSP will be given.

6.1 Heuristics

Wasserman and Faust (1994) propose heuristics based on degree and centrality for solving set selection problems for sociological models. One simple heuristic mentioned is to rank the nodes of a graph by decreasing degree (or centrality), and step-by-step add these nodes to the initially active set until a feasible solution is reached. Kempe et al. (2003) propose a greedy algorithm for solving the TSSP: Starting with an empty target set, each step the node with approximately the highest marginal gain is being selected.

Cordasco et al. (2015) propose a heuristic with a time window constraint: Here, the question whether a node gets activated in time step t is not dependent on the activated neighbors of rounds one to t, but just on the last λ rounds. This approach always produces an optimal solution for trees, cycles, or complete graphs, and produces better approximation for real-world networks than previously published algorithms.

6.2 Exact Solutions

Chen (2009) proves that the TSSP is NP-hard. Ben-Zwi et al. (2011) propose an algorithm for finding an exact solution to the TSSP, which has a complexity of $|V|^{O\omega}$, where ω is the tree width of the graph. Günneç et al. (2016) propose an exact algorithm for the the Least Cost Influence Problem.

7 Complexity of the TSSP

The Target Set Selection Problem is NP-complete. However, there exist kinds of graphs where the TSSP becomes tractable, as Nichterlein et al. (2013) show. Some of these special cases and reduction rules will be shown in the following.

7.1 Reduction Rules for the TSSP

The size of an instance of the TSSP in some cases can be reduced in polynomial time by finding nodes with characteristics that clearly make them either part of the target set, or rule them out as candidates for the target set. One such kind of nodes are **Stubborn Vertices**. Considering the case where the entire graph must be activated ($l = n$), nodes whose threshold $h(i)$ exceeds their degree $deg(i)$ (or in the weighted case the sum of the edge weights of their neighbors $\sum d_{ji}x_{ji}$) cannot be activated by their neighbors. Therefore, they can be added to the target set immediately and the thresholds of their neighbors can be reduced by one. Likewise, vertices which have a threshold $h(v)$ equal to zero can be removed from the candidates for the target set and all their neighbors' thresholds can be reduced by one. Both reductions can be achieved in linear time according to Nichterlein et al. (2013).

Similarly, **Lemming Vertices** can be ruled out as candidates for being in the target set: When a node i exhibits a threshold $h(i) = 1$, it will be activated during the activation process if one of its neighbors j is in the target set. If its neighbour itself possesses $h(j) = 1$, and one of its neighbors is in the target set, vertex j will also be activated. More general, any connected subgraph whose nodes exclusively possess thresholds equal to 1 will be activated if any of its nodes' neighbors outside of the subset is in the target set. Under the assumption that in every connected component of the entire graph G there exists at least one vertex with $h(i) \geq 2$, it follows that there exists an optimal target set which does not contain any vertices with $h(j) \leq 1$. This is the case because in an optimal target set any vertex i with a threshold $h(i) = 1$ can be exchanged by any vertex j not already in the target set which is a neighbour of the subset of vertices connected to i with thresholds equal to one, as is shown by Nichterlein et al. (2013).

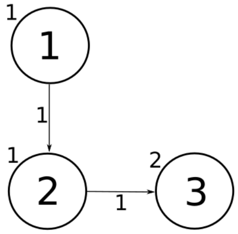


Figure 6: A graph with a stubborn vertex and a lemming vertex

An example of a graph with a stubborn vertex and a lemming vertex can be seen in figure 6. The threshold of node number three ($h_3 = 2$) is higher than the maximum of influence it could receive from its neighbors ($d_{2,3} = 1$). Therefore, it can only be activated by making it part of the target set. Node 2 on the other hand can be ruled out as a candidate for the target set, since it has one incoming edge and has a threshold of 1. It will become active once its neighbouring node 1 will be active. Therefore, there exists no reason to favor node 2 over node 1 for the target set, while node 1 will not be activated when only selecting node 2. Thus, node 2 is a Lemming Vertex.
Furthermore, Nichterlein et al. (2013) establish the rule of **Vertex Addition**: Having a valid target set S for a graph G, when adding a new vertex i to the graph and connecting it to $h(i)$ arbitrarily chosen vertices, the set S will still be a valid target set for the new entire graph G'. Furthermore, any target set S' for the graph G' where i is not in the target set will also be a valid target set for the graph G. These reductions can be done to a graph in arbitrary order without impacting the validity of the solution. Also, subdividing an edge i, j by introducing a new vertex k with threshold $h(k) = 1$ and edges i, k and k, j does not change the cardinality of the optimal target

set of the entire graph. A node i in the target set can be exchanged with a node outside of the target set j with identical neighborhood.
In cliques, the following rule can be applied: The nodes of a clique are being ordered by their degree, the nodes with highest degree are added stepwise to the target set, until the number of nodes of the clique which are part of the target set equals the threshold of the node of the clique outside of the target set with highest degree. This process takes linear time, as Nichterlein et al. (2013) state.

7.2 Parameterized Complexity

The notion of Parameterized Complexity analyses the complexity of problems dependent on multiple parameters. For instance, in general complexity theory, the TSSP is not tractable with increasing input size n. However, as we will see, for the restriction that the diameter of the graph is being restricted to 1, any instance of the problem will be a complete graph, whose target set can be calculated in linear time. In the case of a problem becoming tractable under the restriction of a fixed factor k, the problem is **fixed parameter tractable** for the respective parameter k, as Downey and Fellows (2013) show.
Nichterlein et al. (2013) enumerate various parameterized reductions for the TSSP. A parameterized reduction P' is obtained by applying a function which is bound by the factor k to a problem P in order to make the problem fixed parameter tractable with the factor k. Such parameters are the Cluster Editing Number, the Cluster Edge Deletion Number, the Feedback Edge Set Number, and the Vertex Cover Number. The Cluster Edge Editing Number ζ denotes the minimum number of edges which have to be added or deleted in order to transform a graph into a disjoint union of cliques. Since the clustering coefficient is generally small in small-world graphs, it can be assumed that the Cluster Editing Number is generally low on real-world graphs. The

TSSP is shown to be solvable in $O(16^{\zeta}) \cdot m + n^3$ time, shown by Nichterlein et al. (2013).

The Cluster Edge Deletion Number ξ denotes the minimum number of edges that have to be deleted from a graph in order to transform the graph into a disjoint union of cliques. The TSSP is solvable in $O(4^{\xi} \cdot m + n^3)$ time according to Nichterlein et al. (2013).

The Feedback Edge Set Number f denotes the minimal number of edges which have to be removed from a graph in order to make it acyclic. The TSSP can be solved in $O(4^f \cdot f + m)$ time, shown by Nichterlein et al. (2013). This graph measure seems promising, since the optimal feedback edge set can be computed efficiently by computing a spanning tree. For the example of the Internet, empirical measurements on the graph structure of the known nodes many times consist of a spanning tree. When the connections from one server to another is being measured, packets are being sent from a certain number of servers to a huge number of destinations. The addresses of every node being traversed is being recorded in the packet. Doing this, the shortest path from any of the source nodes to any known node can be measured. Since generally the path of the packets are cyclic only in few exceptions, the feedback edge set number is very low.

7.3 Vertex Cover Number

The Vertex Cover Number τ is the number of nodes of the smallest subset of a graph where at least one of the ends of each edge of the graph is part of the subset. In other words, there must not be any node in the graph that either is in the subset or/and is not neighboring at least one node of the subset. The TSS is shown to be solvable within $O(t_{max} \cdot 2\tau)$ time, where t_{max} is the maximal degree of the graph, shown by Nichterlein et al. (2013).

8 Target Set Selection with Arbitrary Edge Weights

In the following, a variant of the TSSP will be described in detail and a - to the best knowledge of the author - novel ILP formulation given.

8.1 Explanation of the Model

It is assumed that in a network peers that share a connection can influence each other both positively as well as negatively. The act of influence may be non-mutual. I.e. individual A might influence a value of individual B positively, whereas individual B might influence A negatively. In a scenario with both positive and negative influence, the activation effect of changes in centrality measures and number of degree of vertices is not as straightforward as in the case where there is only positive influence. A higher average degree and a higher betweenness do not necessarily mean a smaller target set to influence the same number of nodes. The concepts of submodularity and monotonicity do not hold for this kind of model, as will be described in the following.

The Target Set Selection Problem with Arbitrary Edge Weights is not to be confused with the Minimum-Sized Positive Influential Node Set by He et al. (2017), in which the edges possess positive weights, and a node exerts positive influence on its neighbors if it is activated, and negative influence if it is not activated.

8.2 Adaptions for the Case of Arbitrary Edge Weights

For the case of arbitrary edge weights, i.e. edge weights that can take positive or negative values, some considerations have to be made when establishing a ILP formulation.

8.2.1 Maintaining the Order of Activation

One challenge in finding an ILP-formulation for the Weighted Target Set Problem with Arbitrary Edge Weights lies in the fact that constraints regarding the order of activation must be met in order to maintain the validity of the solution: When a node A with an outgoing edge to node B which has a negative weight is being activated early in the propagation process, node B will have to receive a higher amount of influence in order to be activated than it would have had to before it had been exerted to the negative influence from node A.

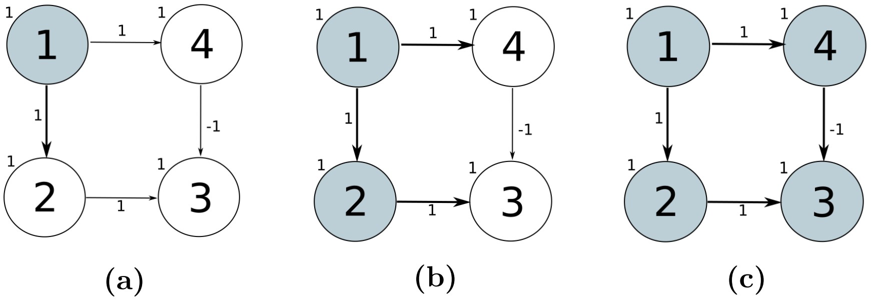


Figure 7: Propagation process with order Violation

An example of a case where the negligence of the order of activation leads to invalidity is depicted in the figure 7a. The figure shows the graph with the minimal solution found with the ILP-formulation for graphs with nonnegative edge weights. This solution is not valid since it violates the rule that edges have to become active as soon as their starting nodes become active. The cascading behavior in this graph starts at node 1, which is in the target set.

In the next round (figure 7b), only node 2, but not node 4 is being activated (which is a violation of the propagation model).

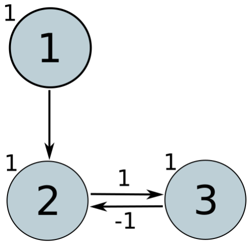
In the final round (figure 7c), node 4 is activated by the edge starting at node 1 and node 3 is being activated by the edge starting at node 2. After that, all the nodes have been activated. The edge between node 3 and node 4 does not do any "harm" anymore since when the edge becomes active, node 3 has already been activated. However, the shown propagation violates the rule that edges must become active as soon as their starting nodes become active. When taking into account this rule, node 4 would become active at the same time as node 2, and therefore node 3 would receive $+1$ plus -1 influence from nodes 2 and 4, respectively, which in sum is not enough to overcome its threshold equal to 1 and activate it. Therefore, the shown solution is not valid.

When respecting the rule that edges must transmit influence as soon as their starting edges become active, no valid target set smaller than two nodes can be found for this graph.

Without the appropriate constraints, the LP algorithm will activate edges with negative edge weights as late as possible. Yet the propagation model demands a node to become active as soon as its received influence reaches or exceeds its threshold. In order to ensure this to be the case in the chosen solution, the LP model has a to have a constraint ensuring there is no lag between activation of a node and its outgoing edges.

8.2.2 Keeping Activated Nodes Active

It must be possible for a vertex, on which negative influence is being exerted, to be active in case it has been activated in a prior round. In figure 9, node 2 has been activated in round 2 because it has received influence of an amount greater or equal to its threshold. The same is the case for node 3 in round 3. In round 4, the total influence received by node 2 is now equal to zero. A constraint strictly demanding that the influence received must be greater or equal than the threshold of a node would forbid the node 2 to be active



(a)

Figure 8: A Graph with a negative edge

in round 4. This would violate the rule that a once activated node must stay active. The problem can be solved by introducing a constant l_i, which for every node i equals the sum of its negative ingoing edges. In the graph depicted in figure 9, the constant l_2 is equal to -1.

The term $l_i \cdot x_i^t$ is being added to the right hand side of the of the influence constraint:

$$\sum_{j}^{N} d_{ji} \cdot x_j^{t-1} \leq (h_i - 1) + (h_i + l_i) \cdot x_i^t \qquad \forall i \in N, t \in (2..n) \tag{20}$$

This has the effect that once a node is activated, its total received influence is allowed to drop below the node's threshold.

8.2.3 Adaption of the Activation Rule

The constant D_i, which denotes the maximum of negative influence a node can receive from its neighbors, is also introduced. Both constants l_i and D_i are products of a preprocessing algorithm that can run in $O(n^2)$ time.

If a vertex i gets activated in time step t, its variable x_i^t is equal to 1, it is 0 otherwise.

The value to minimize is the number of nodes to be active initially. A node is active initially if its value s_i is equal to 1, it is 0 otherwise.

The entire LP formulation is like follows:

$$\min \sum_{i \in N} s_i \tag{21}$$

$$\text{s.t. } l \leq \sum_{i \in N} x_i^l \tag{22}$$

$$\sum_{j \in N} d_{ji} \cdot x_j^{t-1} \leq (h_i - 1) + (h_i + l_i) \cdot x_i^t \qquad \forall i \in N, t \in (2..n) \tag{23}$$

$$h_i \cdot x_i^t \leq (h_i + D_i) \cdot [x_i^{t-1} + (1 - x_i^t)] + \sum_j^N d_{ji} \cdot x_j^{t-1} \quad \forall i \in N, t \in (2..n) \tag{24}$$

$$x_i^{t-1} \leq x_i^t \qquad \forall i \in N, t \in (2..n) \tag{25}$$

$$x_i^1 \leq s_i \qquad \forall i \in N \tag{26}$$

$$s_i, x_i^t \in \{0, 1\} \qquad \forall i \in N, t \in N \tag{27}$$

The size of the target set is to be minimized.(21) The sum of nodes activated in round l has to be at least as big as the predefined value l.(22) Nodes are active if they have been so in the previous time step, or if the sum of their active incoming nodes reaches their threshold, adjusted by the maximum of negative influence the node can receive. (23) Nodes may only change from inactive to active if the amount of influence received exceeds their threshold. The constant D_i is needed to account for already active nodes that become subject to negative influence of neighbors. (24) Nodes cannot be deactivated. (25) In the first round, only the target set nodes are active. (26) The variables for the target set and activation are binary values. (27)

9 Experiments

Above model has been run as a CPLEX script in IBM CPLEX Optimization Studio, Version 12.7.1.0.

9.1 Graph Instances

In the following experiments data, some differences between graphs with strictly positive edge weights and arbitrary edge weights shall be shown. Two kinds of graphs have been created:

- **Random Graphs** generated by a C++-script, in which the number of vertices and directed edges and a range for edge weights can be entered. In the script, the predefined edges get assigned randomly to a pair of vertices. Graph instances have been generated with degrees in the range between 2 and 8, edge weights are distributed equally between -5 and 5 in the case of graphs with arbitrary edge weights, and between 1 and 5 in the case of positive edge weights. The threshold of each node is a random number between 1 and the node's indegree.

- **Small-World Graphs** generated with the Boost (2018) C++ library. As for the Random Graphs, the number of nodes and the range for the edge weights can be defined. The parameters *rewiring probability* and *degree* can be entered. The degree must be a multiple of 2. For the case of the degree being equal to two, a graph, in which each vertex i is connected to the nodes $i - 1$ and $i + 1$ (resembling a ring) is being generated. For the degree being equal to four, each vertex i is being connected to $i - 2$, $i - 1$, $i + 1$, $i + 2$. After constructing the network, each edge changes one of its ends to any other random node with the probability equal to the rewiring probability. As with the random graphs, the edge weights are drawn from a uniform distribution over the range between -5 and +5, unless defined otherwise, and the threshold of each node is a random integer between 1 and the node's indegree.

9.2 Computation Time of the Algorithm

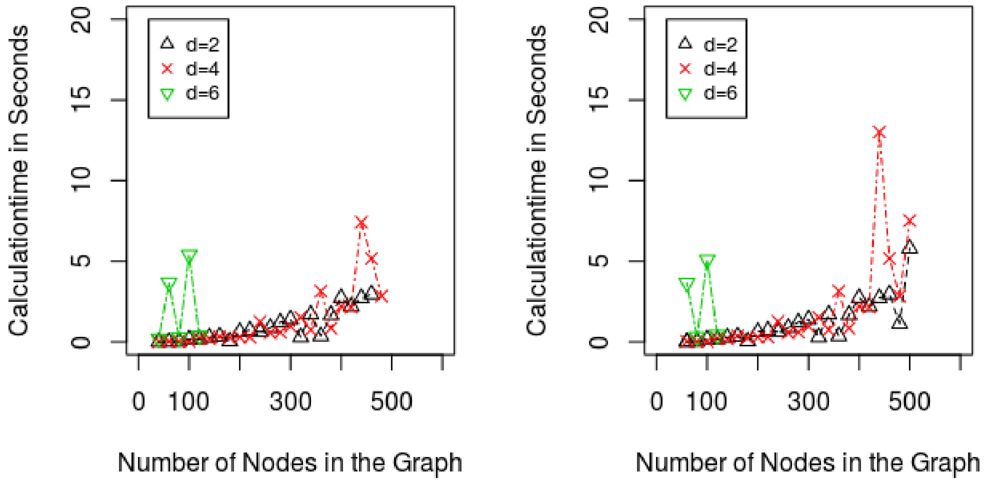
In the following plots each point represents the calculation time in ticks of graph instances of sizes up to 500 nodes. As can be seen, there is a positive relation between the node count of an instance and the time to compute its target set, but what is a far more decisive factor is the density of the graph. The value d represented in the plot is the upper bound for the average node count in each graph. (In each graph, each dth node had a chance to receive a non-zero edge.)

The graphs in figure 10 show the computation time in seconds (one second is supposed to equal 10000 ticks) for graphs with randomly assigned edges (where self-edges are not allowed) and edge weights between -5 and +5 (fig. 9a) and Small-World Graph instances with a rewiring probability of 10%(fig. 9b). The parameter d represents the approximate average degree (for some graphs, the actual average degree deviates marginally as a result of the random graph generation).

It can be observed that the node count of a graph seems to influence the time needed for calculating the target set by a large amount. It also can be observed that the calculation time varies stronger with higher density of the graph.
In figure 11, the calculation of the same instances is shown by their edge count:

9.3 The Effect of Rewiring Probability and Nodes to be Activated

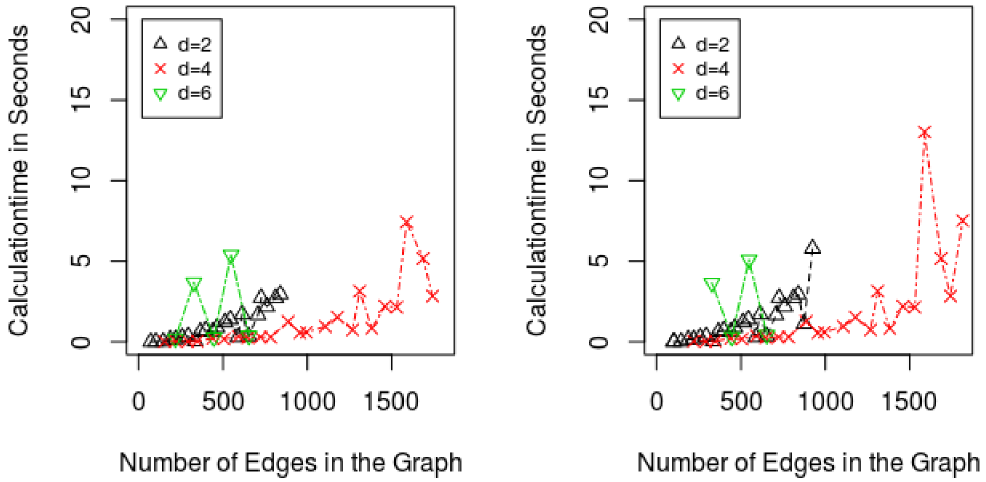
Figure 12 shows graph instances with a size of 100 nodes. The graphs are small-world graphs, in which the nodes have an average degree of 2. The plot



(a)

(b)

Figure 9: Calculation time per number of nodes for (a) Random-Graph instances with arbitrary edges(b) Real-World Graph instances with arbitrary edges



(a) **(b)**

Figure 10: Calculation time per number of edges for (a) Random-Graph instances with arbitrary edges(b) Real-World Graph instances with arbitrary edges

shows both graphs with positive edge weights (equally distributed between 1 and 5) and arbitrary edge weights (between -5 and 5). For each of the two types of graphs, the rewiring coefficient has been set either to zero (the graph resembles a circle), to 0.1 (10% of the edges have been randomly changed) or to 0.5 (50% of the edges have been changed).
The x-Axis shows the number of nodes to be activated (l/n), the y-axis the average size of one optimal target set of all instances of that graph variant and activation constraint. It is evident that the rewiring constraint does not have significant influence on the size of the target set in graphs of the selected parameters, whereas the weight of the edges has.

Figure 13 shows for Small-World graphs with edge weights between -5 and +5, a rewiring probability of 10%, an average degree of 2 and various size, how the size of the optimal target set increases with raising the amount of the graph to be activated (l/n). It is evident that a certain activation constraint requires a certain ratio of nodes in the target set, no matter the graph size. For instance, to activate 100% of the nodes, 60% of the nodes must on average be in the target set, which is true for instances of all observed sizes for these graph parameters.

9.4 Influence of Graph Size and Structure on the Size of the Target Set

For low average degree values, Small-World Graphs (other than Random Graphs) consist of on single connected component.
The following plots show the size of the Target Set for Graphs of a node count of up to 500 and average degrees between 2 and 6. In figure 13, optimal solutions to instances of varying sizes of Random Graphs and Small-World Graphs with rewiring probability of 0.1 are being compared. All graphs have an average edge degree of 4. As we can see, the consequences of the graph

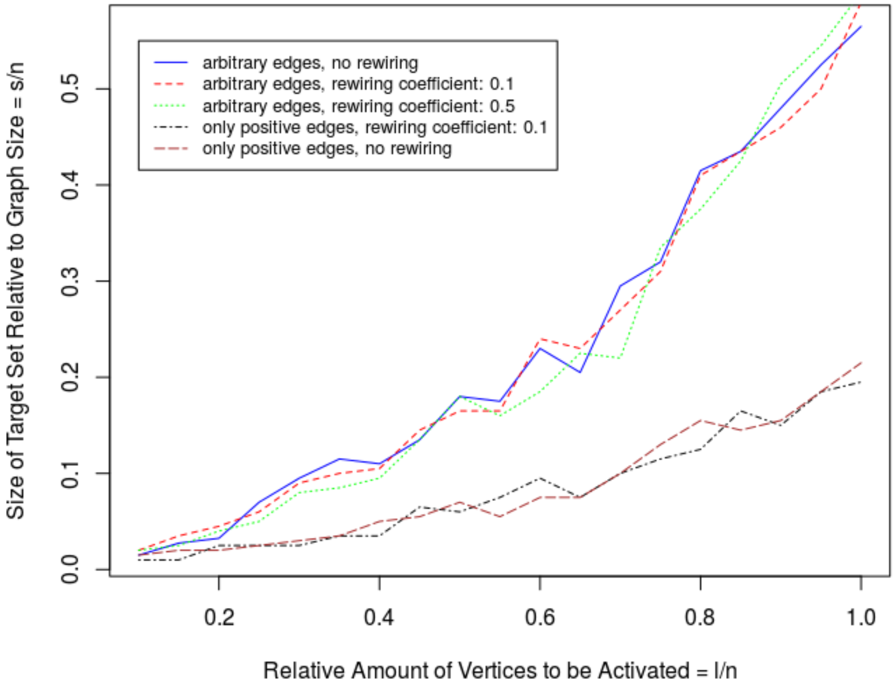


Figure 11: Target set sizes for graphs of different rewiring coefficients and number of nodes to be activated.

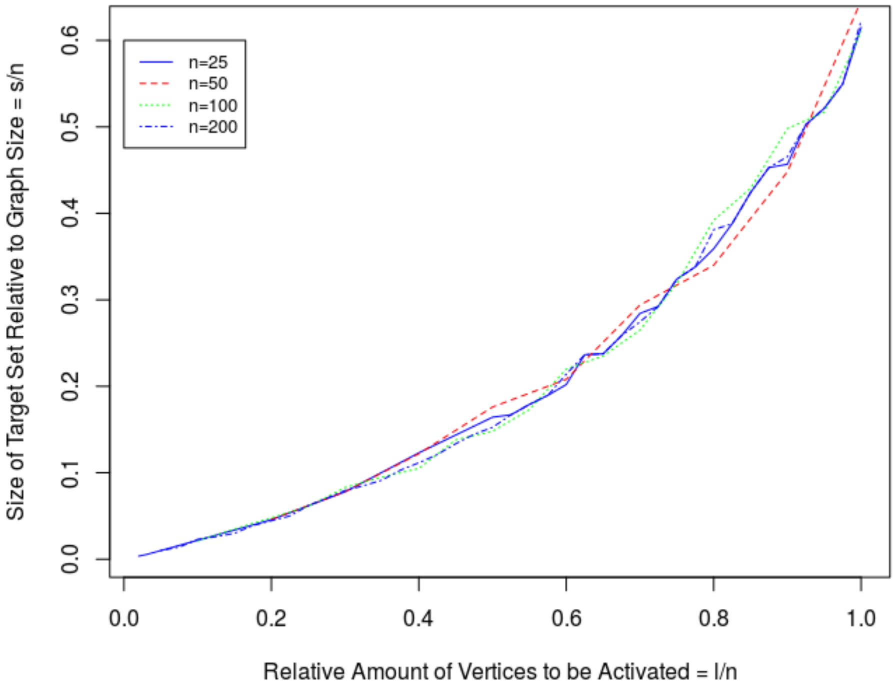


Figure 12: Relative target set sizes for graph instances with different number of nodes to be activated.

topology discussed earlier do not seem to apply for graphs of arbitrary edge weights. The feature of possessing a Small-World-like structure seems to have the effect that more nodes need to be taken into the target set in order to counterbalance the effect of negative influence.

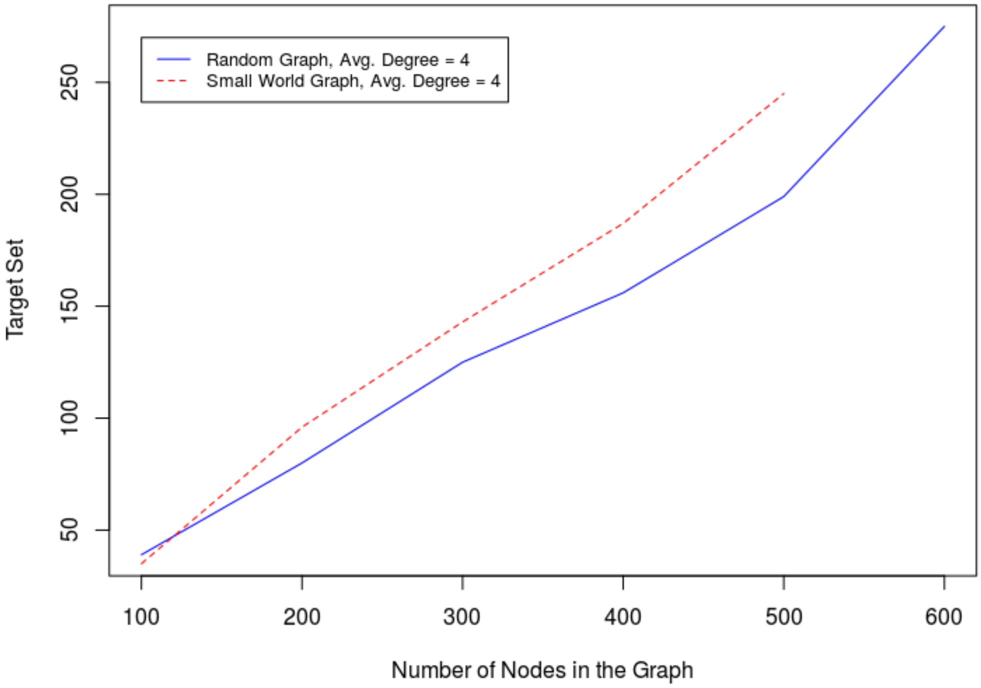


Figure 13: Size of the target sets of Random and Small-World Graphs

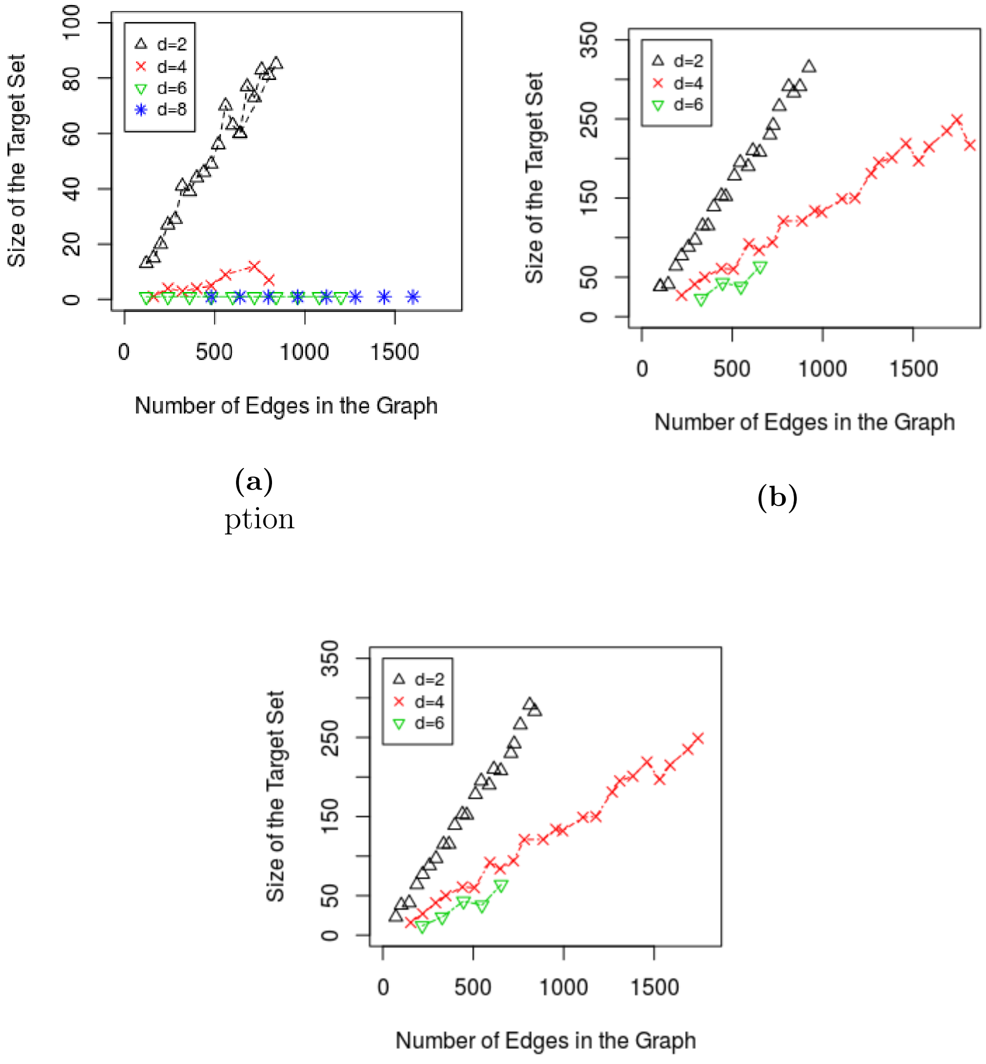
The Size of the Target Set of a Graph instance depends on parameters like size and density of the graph, edge distribution, nodes to be activated, and edge weights.

As we can see, there is a positive relation between graph size and size of the target set, and graphs with a higher average degree have small target sets (at least for a rewiring probability of 10 per cent). I can be seen that for the graph with purely positive edges, as soon as the degree exceeds the value of vertices, one node is sufficient to activate all nodes of the target set. For an average degree of 2, the size of the target set increases almost linear with the node count of the graph. With the examined graphs with arbitrary edges, the size of the target set seemed to be independent of the degree.

Having a look at the size of the target set by node size for larger instances (fig. 13), the overall impression is the same.

9.5 Real-World Instances

The following graph instances were taken from a dataset provided by Leskovec et al. (2010). It is data from the consumer review site 'Epinions.com'. On this site, consumers can post reviews on products as well as rate each other on their trustworthiness. It is possible to give either a positive or a negative rating. The data was remodeled in the following form: The weight of a directed edge between i and j equals $+1$, if user j trusts user i (or equal to -1 in the case of distrust). The threshold for each user to be convinced was set to 1. The entire graph contains 131828 nodes and 841372 edges, i.e. there exist on average about 6 ratings per person. In the plot the size of the target set with an increasing subset of the graph can be seen. For each subset, a predetermined number of nodes together with their respective edges have been chosen arbitrarily from the entire set.

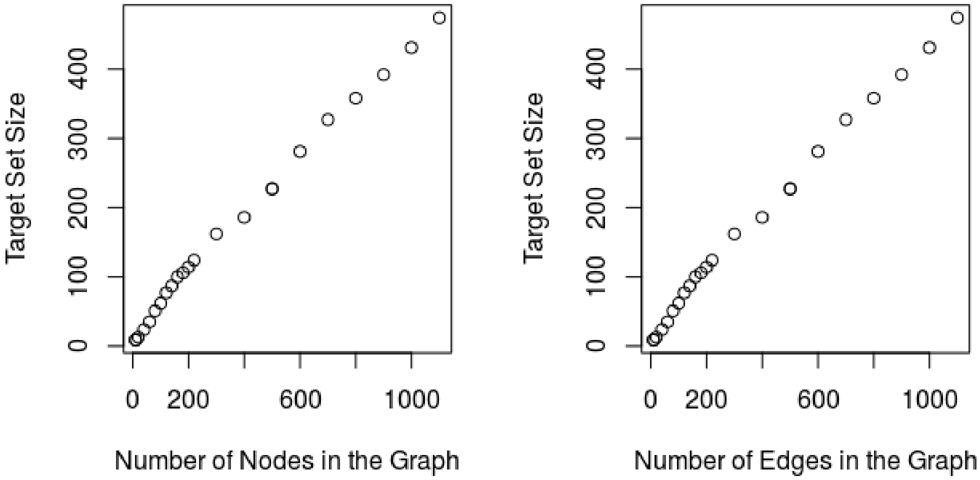


(a)
ption

(b)

(c)

Figure 14: Target set sizes for (a) Real-World Graph instances with positive edges (b) Real-World Graph instances with arbitrary edges (c) Random-Graph instances with arbitrary edges



(a) **(b)**

Figure 15: Target set sizes for subsets of the 'Epinions.com' dataset by (a) number of nodes (b) number of edges

10 Conclusion

This master's thesis has reasoned, why the Target Set Selection Problem (TSSP) is of high relevance to research topics of gaining importance. The impact of graph properties on the complexity of solving the TSSP was shown. For the case of graphs with arbitrary edge weights, a novel integer linear programming formulation has been established and it has been shown where the pitfalls of finding targets sets in graphs with arbitrary edge weights are. For instances with different structural features, as well as for real-world data, solutions have been found and analyzed. One of the findings is, that even more than the size of the instance, the network density leads to a great increase in calculation time.

In an increasingly interconnected world, small causes can have large effects due to propagation processes. Therefore, it is important to develop models that are closer to reality and to deepen the understanding of the effect of network features on propagation. This thesis is meant to do its humble contribution to this great challenge.

References

Chalee Asavathiratham, Sandip Roy, Bernard Lesieutre, and George Verghese. The influence model. *IEEE Control Systems*, 21(6):52–64, 2001.

Robert Axelrod. The dissemination of culture: A model with local convergence and global polarization. *Journal of conflict resolution*, 41(2):203–226, 1997.

Albert-László Barabási and Jennifer Frangos. *Linked: the new science of networks science of networks*. Basic Books, 2014.

Albert-László Barabási and Márton Pósfai. *Network science*. Cambridge university press, 2016.

Peter S Bearman, James Moody, and Katherine Stovel. Chains of affection: The structure of adolescent romantic and sexual networks. *American journal of sociology*, 110(1):44–91, 2004.

Oren Ben-Zwi, Danny Hermelin, Daniel Lokshtanov, and Ilan Newman. Treewidth governs the complexity of target set selection. *Discrete Optimization*, 8(1):87–96, 2011.

Béla Bollobás and Oliver Riordan. *Percolation*. Cambridge University Press, 2006.

Boost. Boost c++ libraries. 2018. URL `http://www.boost.org`. Last accessed 2018-05-02.

Ning Chen. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics*, 23(3):1400–1415, 2009.

Wei Chen, Laks VS Lakshmanan, and Carlos Castillo. Information and influence propagation in social networks. *Synthesis Lectures on Data Management*, 5(4):1–177, 2013.

Reuven Cohen, Keren Erez, Daniel Ben-Avraham, and Shlomo Havlin. Breakdown of the internet under intentional attack. *Physical review letters*, 86(16):3682, 2001.

Gennaro Cordasco, Luisa Gargano, Marco Mecchia, Adele A Rescigno, and Ugo Vaccaro. A fast and effective heuristic for discovering small target sets in social networks. In *Combinatorial Optimization and Applications*, pages 193–208. Springer, 2015.

Zoltán Dezső and Albert-László Barabási. Halting viruses in scale-free networks. *Physical Review E*, 65(5):055103, 2002.

Rodney G Downey and Michael R Fellows. *Fundamentals of parameterized complexity*, volume 4. Springer, 2013.

Paul Erdös and Alfréd Rényi. On random graphs, i. *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

Eyal Even-Dar and Asaf Shapira. A note on maximizing the spread of influence in social networks. In *International Workshop on Web and Internet Economics*, pages 281–286. Springer, 2007.

Matteo Fischetti, Michael Kahr, Markus Leitner, Michele Monaci, and Mario Ruthmair. Least cost influence propagation in (social) networks; a publication of the mathematical optimization society. *Mathematical Programming, 2018, Vol.170(1), pp.293-325*. ISSN 0025-5610.

Malcolm Gladwell. *The tipping point: How little things can make a big difference*. Little, Brown, 2006.

Mark Granovetter. Threshold models of collective behavior. *American journal of sociology*, 83(6):1420–1443, 1978.

Dilek Günneç, S Raghavan, and Rui Zhang. Tailored incentives and least cost influence maximization on social networks. Technical report, Technical report, 2016.

Andrew G Haldane and Robert M May. Systemic risk in banking ecosystems. *Nature*, 469(7330):351–355, 2011.

Jing Selena He, Ying Xie, Tianyu Du, Shouling Ji, and Zhao Li. Influence spread in social networks with both positive and negative influences. In *International Computing and Combinatorics Conference*, pages 615–629. Springer, 2017.

David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 137–146. ACM, 2003.

Jure Leskovec, Daniel Huttenlocher, and Jon Kleinberg. Signed networks in social media. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 1361–1370. ACM, 2010.

Yanhua Li, Wei Chen, Yajun Wang, and Zhi-Li Zhang. Influence diffusion dynamics and influence maximization in social networks with friend and foe relationships. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 657–666. ACM, 2013.

Stanley Milgram. The small world problem. *Psychology Today*, 1:61–67, 1967.

Michael Molloy and Bruce Reed. A critical point for random graphs with a given degree sequence. *Random structures & algorithms*, 6(2-3):161–180, 1995.

National Research Council. *Network Science*. The National Academies Press, Washington, DC, 2005. ISBN 978-0-309-10026-7. doi: 10.17226/11516. URL `https://www.nap.edu/catalog/11516/network-science`.

André Nichterlein, Rolf Niedermeier, Johannes Uhlmann, and Mathias Weller. On tractable cases of target set selection. *Social Network Analysis and Mining*, 3(2):233–256, 2013.

Nishith Pathak, Arindam Banerjee, and Jaideep Srivastava. A generalized linear threshold model for multiple cascades. In *2010 IEEE International Conference on Data Mining*, pages 965–970. IEEE, 2010.

David Peleg. Local majorities, coalitions and monopolies in graphs: a review. *Theoretical Computer Science*, 282(2):231–257, 2002.

Matthew Richardson and Pedro Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 61–70. ACM, 2002.

Mark DF Shirley and Steve P Rushton. The impacts of network topology on disease spread. *Ecological Complexity*, 2(3):287–299, 2005.

Bo Söderberg. General formalism for inhomogeneous random graphs. *Physical review E*, 66(6):066121, 2002.

Gwen Spencer and Richard Howarth. Maximizing the spread of stable influence: Leveraging norm-driven moral-motivation for green behavior change in networks. *arXiv preprint arXiv:1309.6455*, 2013.

Toshi Tanizawa, Gerald Paul, Reuven Cohen, Shlomo Havlin, and H Eugene Stanley. Optimization of network robustness to waves of targeted and random attacks. *Physical review E*, 71(4):047101, 2005.

Andrew J Tatem, David J Rogers, and SI Hay. Global transport networks and infectious disease spread. *Advances in parasitology*, 62:293–343, 2006.

Alexei Vázquez and Martin Weigt. Computational complexity arising from degree correlations in networks. *Physical Review E*, 67(2):027101, 2003.

Stanley Wasserman and Katherine Faust. *Social network analysis: Methods and applications*, volume 8. Cambridge university press, 1994.

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

Duncan J Watts and Steven H Strogatz. Collective dynamics of small-worldnetworks. *nature*, 393(6684):440, 1998.

Rui Zhang and Mustafa Sahin. Combinatorial optimization games arise in social networks. 2014.

Appendices

A Generated Graph instances

This is an extract of the solutions for the instances with mixed edge weights.

This is an extract of the solutions for the instances with arbitrary edge weights:

l	n	edges	targetset	negativeedges	positiveedges
100	100	370	35	156	214
200	200	734	96	352	382
300	300	1,106	143	528	578
400	400	1,478	187	724	754
500	500	1,812	245	906	906
700	700	2,568	360	1,316	1,252

These are the solutions for the graphs with positive edge weights:

l	n	edges	targetset	negativeedges	positiveedges
100	100	200	21	0	200
200	200	400	39	0	400
300	300	600	68	0	600
400	400	800	83	0	800
500	500	1,000	113	0	1,000
600	600	1,200	130	0	1,200