Universiteit Leiden

Master Computer Science

Fairness-aware edge blocking
for influence minimization

Name:	Diego van Egmond
Student ID:	s2243458
Date:	01/06/2024
Specialisation:	Artificial Intelligence
1st supervisor:	Dr. Akrati Saxena
2nd supervisor:	Dr. Frank Takes

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

In influence minimization the objective is to change a given (social) network in such a way that the spread of influence from a given set of source nodes is minimized. The problem was initially introduced as a variation of the influence maximization problem. In this new variation, the goal is to limit the spread of misinformation, rather than to spread desired information. Two approaches to address the influence minimization problem are removing nodes from the network, and removing edges from the network. The resulting network is evaluated by finding the number of saved nodes, i.e., those nodes that become activated in a diffusion simulation on the original network, but remain inactive in a diffusion simulation on the new network from which the selected set of nodes or edges has been removed. In this thesis, we specifically look at the fairness of influence minimization by edge removal. We consider a method to be fair if the number of saved nodes for each community is proportional to the size of the community. We introduce heuristic methods, and compare them using different time limits, edge set sizes, and heuristics for estimating the importance of edges in the diffusion of influence. We evaluate the performance by looking at the total proportion of saved nodes, and we evaluate the fairness by comparing the proportions of saved nodes for each affected community. The fairness metrics we use are disparity and maximin. Disparity is the maximum difference of the proportional saved node between any two communities. Maximin fairness shows the minimum of proportional saved node over all communities Based on our experimental results, we find that changing the edge weight heuristic has a much larger effect on the resulting blocking performance and group fairness than changing the edge selection model. There is a clear trade-off between the two metrics, the edge betweenness heuristic giving us the best performance, and the degree heuristic, calculated as the product of the two node degrees, giving us the fairest result.

Preface

This master thesis is my final project as part of my study towards a master's degree in Computer Science at Leiden University. I have been working on this thesis from December 2022 until April 2024, and I am proud of the result.

First, I would like to thank my supervisor Dr. Akrati Saxena for her invaluable support. She has been very helpful in shaping the research, formalizing goals, and brainstorming solutions to any problems I encountered. Her feedback throughout the project has been essential in achieving this final product.

Second, I would like to thank my second supervisor Dr. Frank Takes for helping me find this topic for my thesis, for the feedback he has given me, and for teaching me most of what I know about working with networks. His course on social network analysis (SNA) has given me all the tools I needed to create the experiments used to obtain my results.

I would also like to thank my fellow students and friends Corné Spek and Michael van der Zwart, who I have often worked with on campus. They have provided me with useful insights when I was stuck on problems, brought additional tools to my attention to improve the quality and efficiency of my programs, and have been great in making my time at the university a very pleasant and enjoyable one.

Finally, I would like to thank my friends and family for supporting me unconditionally throughout this whole endeavour. All the support is greatly appreciated.

Contents

1 Introduction

The first section of this thesis will give preliminary information regarding the study of social networks, and the place of this thesis within it. Subsequently, a general overview of information diffusion, its blocking, and the fairness-aware influence blocking is provided. Finally, the contributions of this thesis are listed, and the thesis structure is outlined.

1.1 Social networks

People form connections wherever they go. This can be in the form of friendships, collaborations, business contracts, competitions or any other kind of social relations. Such a system of interconnected people, groups, or organisations is called a social network. Social network analysis (SNA) is the field that looks at these networks. Some of the topics in this field include the growth of networks [1, 2, 3, 4, 5, 6], network anonymization [7, 8, 9], information diffusion [10, 11, 12, 13], and network visualization [14, 15, 16]. The main focus of this thesis is information diffusion, which is the spread of information through a social network.

1.2 Information diffusion

A lot of information and ideas are spread over the internet, particularly social media. However, since everybody is given a voice, the validity of information is not ensured. A big problem we see nowadays is the rapid spread of misinformation through social media. A rumour might start with a small group of people, but each of them tells some of their friends, and the new group of people who believe it go on to spread it even further. The spread of information from one person to another is called information propagation. The process of the spreading of information through a network is called information diffusion. This diffusion is the result of many instances of information propagation. Such information diffusion is simulated in our experiments by a diffusion model. The model we use is the Independent Cascade (IC) model, which gives each edge an individual propagation probability. This model iteratively generates a new subset of activated nodes based on the previous set of activated nodes. These activated nodes are the nodes that have been influenced by the misinformation. One such state of the network in which a set of nodes is active is called a timestep. Another diffusion model we use is the Linear Threshold (LT) model, which gives each node an individual influence threshold. Like the IC model, this model iteratively generates a new subset of activated nodes for each timestep. A new node is influenced if the proportion of its neighbors that have been influenced exceeds its threshold.

1.3 Influence blocking

Blocking the spread of such misinformation [17, 18, 19, 20, 21] is a large area of research within the field of SNA. In this thesis, we represent the social networks as graphs, and the spread of misinformation is simulated using boolean values representing the opinions of individuals within a network, in terms of whether the individual believes the misinformation or not.

There have been several approaches of influence blocking. One proposed solution for this problem has been to spread truthful information in the hopes it arrives sooner to many people than the misinformation does. The problem with this approach is that the misinformation has a head start and will likely spread faster. Another proposed solution is to simply inform a group of people about the misinformation to ensure it does not spread beyond them. However, it is not possible to make sure that everybody we try to convince will actually block this misinformation, and it would not be fair to block or ban these influential people from the network.

The approach we will focus on in this thesis is edge blocking, since blocking, removing or immunizing connections might be easier and more fair to implement in a real social network than blocking or removing users. In this approach, we select a subset of edges of a particular size to block. Once blocked, influence can no longer be propagated by these edges. We implement this blocking of edges by removing them from the network. This will have the same effect as setting their propagation probabilities to 0 for our experimental setup. However, we find that for other diffusion models there is an important difference. For these models, removing an edge increases the weights of other edges. In those cases it is important to differentiate between edge blocking and edge removal. For this reason, we use the IC model to simulate influence diffusion in our experiments, and the terms edge blocking and edge removal are used interchangeably.

1.4 Fair influence blocking

Social networks can often be partitioned into a number of communities. Such a community is a subset of nodes that have more connections to other nodes within that subset than to nodes outside of their respective subset.

A problem that arises in influence blocking is that the optimal solution often disproportionately protects nodes belonging to large communities. Due to this structural bias, smaller communities are often disregarded. To minimize this effect, we search for a solution for which the number of protected nodes per community is proportional to the community size. We call this additional objective group fairness. Saxena et al. [22] give an overview of fairness measures and constraints in several different fields within SNA.

The problem of limiting misinformation was expanded recently by Bierbooms [23] to include group fairness. The proposed approach is a fair truth-campaigning method. This thesis is the first work on fair influence blocking to consider removing edges. In our approach, we assign a weight to each edge using one of four proposed heuristics. Given these weights, we select a set of edges to be removed from the network using one of three proposed edge selection methods. We compare the influence blocking performance and fairness of all pairs, each consisting of a weight heuristic and an edge selection method.

1.5 Results

In our experiments, each proposed method consists of an edge weight heuristic and an edge selection model. The selected set of blocked edges is evaluated using three metrics: a metric for blocking performance, a metric for group fairness, and a combined metric. Our experimental results show that the proposed models are closely grouped together by edge heuristic with regard to each of the used metrics. Given this finding, we conclude that the effect of the used edge selection model on the resulting blocking performance and group fairness is very small. Therefore, it is better to focus only on the edge weight heuristics. Our results show that a trade-off between fairness and performance is made. The betweenness centrality heuristic maximizes the performance, and the node degree heuristic maximizes the fairness. The highest performance is reached at a low number of timesteps, but this is also paired with the lowest fairness. We also see that the expected number of saved nodes increases monotonically as the number of removed edges increases.

1.6 Contributions

In this thesis we have made the following contributions:

1. In Section 4.3, we propose two new fair edge selection methods to select a set of edges to be removed, given a set of edge weights and the network structure. We use the Louvain model to detect the communities, and we consider an edge to belong to a community if at least one of the nodes it connects belongs to this community. Both of these methods aim to select a set of edges such that the number of selected edges belonging to a community is proportional to the total number of nodes belonging to this community.

2. In Section 4.4.1, we propose the HICH-BA2 model for network generation. This new model builds on the existing HICH-BA model by adding a parameter for the preferential attachment strength. Furthermore, we ensure that the resulting number of edges is equal to the desired number of edges.

3. Based on the experiments in Section 5.2.1, we find that it is important to distinguish between edge blocking and edge removal. Removing edges causes the network structure to change, which may affect the influence diffusion process. Blocking edges simply prevents them from propagating influence, without making any changes to the network. The importance of this distinction is further explained in Section 6.1.

4. Based on the initial analysis described in Section 5.2.1, we find that the LT model is not suitable for simulating the diffusion in our experiments. Since we remove edges based on their individual weights, we expect the number of saved nodes to monotonically increase as we remove more edges. This is the case with the IC model, which is therefore a suitable propagation model for our experiments.

5. Based on the initial analysis described in Section 5.2.2, we find that not all communities should be considered when evaluating the fairness of a method. We define the set of affected communities as the set of communities containing at least one node that is reached by influence in a network before any edges are removed. For big networks, there is often at least one community that is not affected. The proportion of saved nodes for such a community is always zero, because none of the nodes in this community are originally reached by influence, and therefore none of these nodes can be saved. This is a problem because our fairness evaluation metrics look at the fairness of the least protected community and the most protected community. If one of these values is a constant value, the fairness of influence minimization methods between different communities can not be properly evaluated.

1.7 Thesis Structure

The background of the field of fair influence blocking is given in Section 2, followed by our problem definition and proposed methods in Section 3. The used methods and resources are explained and listed in detail in Section 4. Section 5 contains unexpected findings made before the final experiments were finished, as well as the results of our final intended experiments. Finally, conclusions and future work are discussed in Section 6.

2 Related work

In this section we will discuss the previous work upon which our research builds. In Section 2.1 we first give some preliminary information relating to the topic of network science, and define the relevant terms. In Section 2.2 we define the diffusion of influence, and the different models that are used to simulate it. Section 2.3 gives an overview of the research looking at the problem of maximizing this diffusion of influence. Conversely, Section 2.4 gives an overview of the research looking at the problem of minimizing the diffusion of influence. Chen et al. [17] discusses several more different influence minimization methods, as well as some of the fields in which the research is applicable. Furthermore, an overview of fake news propagation methods and mitigation techniques such as influence blocking and truth campaigning is given by Saxena et al. [24]. Finally, Section 2.5 discusses the research of fairness for several problems in SNA.

2.1 Preliminaries and definitions

Network science is the study of networks. The first work in network science is the solution to the Königsberg bridge problem, proposed by Leonhard Euler [25] in 1736. The problem is to find a route through the city of Königsberg in Prussia, which is currently known as the city of Kaliningrad in Russia, such that each bridge in the city is crossed exactly once. A sketch of the city is shown in Figure 1.

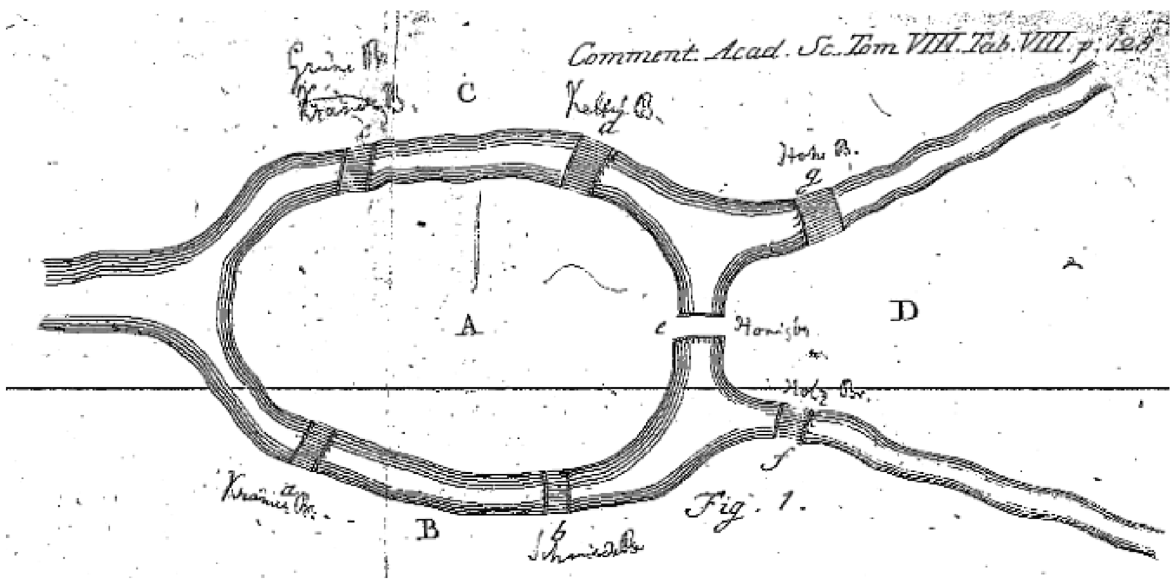

Figure 1: Sketch of the city of Königsberg, taken from [25].

Since the route taken through the city is irrelevant and only the order of bridges crossed matters in this problem, the figure can be simplified to a graph, as seen in Figure 2. In this graph, we represent each land mass as a node, or a vertex. Each bridge connecting two land masses is represented as an edge, or a link. Since each edge connects two nodes, an edge can also be denoted as a pair of nodes $e = (v_1, v_2)$. A set of nodes is denoted $V = \{v_1, v_2, \dots\}$, and a set of edges is denoted $E = \{e_1, e_2, \dots\}$. A set of nodes and a set of edges together make up a network, or a graph, denoted as a pair of sets $G = (V, E)$.

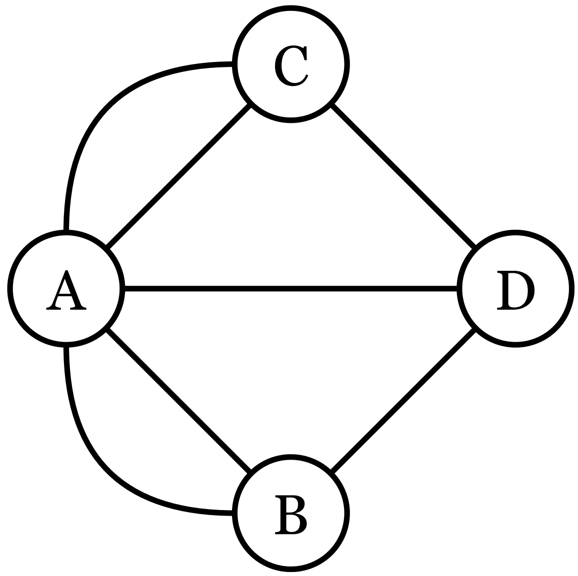

Figure 2: Graph representing the city of Königsberg, based on Figure 1.

In a network, each node v is connected by edges to a set of other nodes. This set of connected nodes is called the set of neighbors of v, denoted $N(v)$. The number of neighbors a given node has is called its degree, which we denote as d_v. The notation of all variables and functions used in this thesis are listed in Table 1.

2.2 Diffusion models

Kempe et al. [26] expanded the field by introducing propagation models to simulate the diffusion of influence through the network. Two such models are the *Linear Threshold* (LT) model and the *Independent Cascade* (IC) model. The LT model is a generalization of many previous works that use node-specific thresholds. Some of the first models that make use of such thresholds were proposed by Granovetter [27] and Schelling [28]. Cascade models are based on works from probability theory such as Durrett [29] and Liggett [30].

The IC model is considered the conceptually simplest kind of cascade model, in which each node has a randomly sampled activation probability. In this model, every node that was activated in the previous iteration will attempt to activate each of its inactive neighbors separately using their respective activation probabilities. Another variation of the IC model was first investigated in the context of marketing by Goldenberg et al. [31, 32]. In this variation, there is a strong or a weak influence probability for every pair of people, and an individual marketing probability. These probabilities are combined to calculate the probability that a node becomes informed at a timestep t. The generalized version of Kempe et al. [26] considers only the strong influence probability, and for each pair of individuals the influence propagation is can only be attempted once at most. A special case of the IC model, the *Weighted Cascade* (WC) model, is introduced by Kempe et al. [26]. This model computes the probability p_{ij} that an active (influenced) node i activates its inactive neighbor j as $p_{ij} = \frac{1}{d_j}$. An active node is only able to activate its neighbors in the first timestep after being activated. The process starts with an initially active subset $M \subset V$, and runs until there are no more inactive nodes that can be activated at the next timestep. The goal is to maximize the spread of influence, denoted $\sigma(M)$. Since not all

v	A node
e	An edge (v_1, v_2) connecting nodes v_1 and v_2
V	Set of nodes $\{v_1, v_2, \dots\}$
E	Set of edges $\{e_1, e_2, \dots\}$
G	A network (V, E)
$N(v)$	Set of neighbors of v
d_v	Degree of node v, defined as its number of neighbors $\lvert N(v)\rvert$
$G(v)$	A network $\{V - \{v\}, E\}$, equal to the network G excluding the node v
$G(V')$	A network $\{V - V', E\}$, equal to the network G excluding the subset of nodes V'
$G(e)$	A network $\{V, E - \{e\}\}$, equal to the network G excluding the edge e
$G(E')$	A network $\{V, E - E'\}$, equal to the network G excluding the subset of edges E'
V_2	The subset of nodes with a degree of 2 or higher
M	An initially active subset of nodes
A	The subset of activated nodes, i.e., the set of nodes influenced by M
A_t	The set A at timestep t
$A_{t,c}(G, M)$	The subset of A_t belonging to community c for a given network G with initially active subset M
D	The set of edges that is removed
k	The desired number of edges to be selected for D
$\mathbb{N}$	The set of natural numbers, i.e. positive whole numbers
c	A community to which a node can belong
c_v	The community to which a given node v belongs
C	The set of all communities $\{c_1, c_2, \dots\}$
r_c	The proportion of nodes that belong to community c
$\vec{r}$	A vector containing all community proportions r_c
k_c	The number of desired occurrences of an edge with community c in the selected set of blocked edges D with total size k
p_t	Probability of generating a triangle in a network generation step
p_{PA}	Probability that preferential attachment is active in a network generation step
α	The strength of preferential attachment in network generation
t	A single timestep in the diffusion process
T	The total number of timesteps in the diffusion process
p_{ij}	The probability that active node i influences its inactive neighbor j
$\sigma_t(G, M)$	The number of active nodes $\lvert A\rvert$ in network G with an initially active set M at a given timestep t
$\sigma(G, M)$	The spread of influence through network G from set M at timestep $t = T$
$\Delta\sigma(G, M, D)$	The number of nodes saved from the influence of M by removing edge subset D from the network G
$\pi_{\Delta\sigma(G,M,D)}$	The value of $\Delta\sigma_t(G, M, D)$ as a proportion of the total number of nodes
σ_{st}	The number of shortest paths between nodes s and t
$\sigma_{st}(e)$	The number of shortest paths between nodes s and t that pass through edge e
$d(v_1, v_2, G)$	The distance between nodes v_1 and v_2 in network G
$x \xleftarrow{\$} X$	Uniformly sampling an element x from a set X

Table 1: Notation

people are equally persuasive in a real world network, this equation was adapted by Hong et al. [33] to take into account the degrees of all neighbors of j as a measure of their persuasiveness. This new *Persuasiveness Weighted Cascade* (PWC) model computes the propagation probability as $p_{ij} = \frac{d_i}{\sum_{k \in N(j)} d_k}$.

Besides these, there are several other extensions of the IC model to simulate influence propagation in real-world, such as shortest path model [34], penta-level spreading model [35, 36], trust-based latency-aware independent cascade [37], conformity-aware cascade model [38], continuous-time Markov chain-independent cascade model [39], and dynamic independent cascade model [40].

In all these propagation models, an activated node remains activated. A propagation model with this property is referred to as a ratcheted dynamical system, progressive system or irreversible system, by Kuhlman et al. [41], Kempe et al. [26], and Dreyer et al. [42] respectively. An example of a dynamic propagation that does not have this property is the SIR model, in which the states are not *active* and *inactive*, but *susceptible*, *infected*, and *recovered*. This propagation model is used for influence minimization by Wang et al. [20]. An overview of different propagation models is given by Chen et al. [17].

2.3 Influence maximization

Influence Maximization is a big area of research that started when Domingos et al. [43, 44] proposed a viral marketing method as an alternative to the traditional mass marketing in which products are promoted to all potential customers indiscriminately. This viral marketing method represents the potential customers and their connections as nodes and edges in a network, and selects a highly influential subset of these nodes to maximize the *expected lift in profit* (ELP). The ELP is defined as the expected increase in the spread of influence when the viral marketing is applied, compared to the expected spread of influence when it is not. This spread of influence is the number of nodes that have become activated at the iteration where the diffusion process stops, i.e. the number of nodes to which the influence has spread. The advantages of this method have been demonstrated on real world data.

Kempe et al. [26] applies a greedy hill-climbing algorithm to their version of the influence maximization problem, which is related to the approach of Domingos et al. [43]. This algorithm starts with an empty set, and repeatedly adds the node that gives the highest gain. The efficiency of this greedy algorithm is improved by Chen et al. [45] using degree discount heuristics.

The competitive multi-campaign influence maximization problem was introduced by Bharathi et al. [46]. Using a network with two campaigns in which players define their seed sets sequentially, Kostka et al. [47] shows that there are networks in which the second player can always obtain a higher spread of influence than the first player. This is shown in Figure 3.

2.4 Influence blocking maximization

Contrary to the influence maximization problem, the goal of the influence blocking maximization problem is to minimize the spread of influence. This shifts our focus from contexts of viral

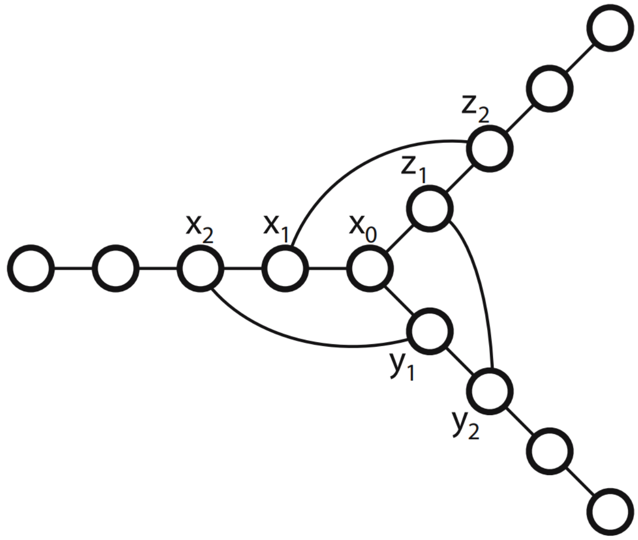

Figure 3: Example network, taken from [47], in which two players can sequentially select a single seed node. After selecting the initial nodes, the campaign spreads iteratively to inactive neighboring nodes until each node has been reached by a campaign. If the first player selects node x_0, the second player can select node x_1. In this situation, the first campaign starting from x_0 spreads to the nodes y_1 and z_1, and the second campaign spreads to the nodes x_2 and and z_2. In the following timesteps, the rest of y branch is reached by the first campaign, and all other inactive nodes are reached by the second campaign. This way, the first campaign reaches six nodes, and the second campaign reaches seven nodes. This shows that the starting node x_0 is outperformed by the starting node x_1. Similarly, x_1 is outperformed by x_2, and x_2 is outperformed by z_1.

marketing to contexts such as the desired limitation of the spread of misinformation. The significant impacts of fake news and misinformation propagation on elections and similar events are widely recognized. Researchers have proposed several methods to detect fake news and misinformation [48, 49]. In Influence blocking, given a set of individuals spreading negative influence, we aim to change the network in a way that minimizes the number of individuals that are reached by this influence.

Initial heuristic approaches on influence maximization used to choose top-k nodes based on a centrality measure [50], such as degree ranking [51], closeness ranking [52], k-shell index [53], and Pagerank [54]. An early approach on influence minimization was proposed by Aspnes et al. [55]. In this paper, the goal is to choose a minimal set of nodes $D \subseteq V$ that do not propagate influence, such that the spread of a randomly selected misinformation node $m \in V$ is minimized. A good strategy to select these nodes is in descending order of degree ranking. Broder et al. [56] shows that the connectivity of the network remains largely unaffected by the elimination of such nodes. However, Albert et al. [57] shows that the diameter of a typical network can be doubled by removing only the top 5% most highly connected nodes from the network. This is due to the inhomogeneity of the connectivity distribution, which means that there is a high variation in the degrees of nodes. This connectivity distribution follows the power-law distribution, as seen in Barabasi et al. [58], which is attributed to the preferential attachment property. This is the property that a node is more likely to form links with more highly connected nodes. Other strategies to select a subset of nodes to block are heuristics based on betweenness and closeness centrality measures, as shown by Habiba et al. [59], and heuristics based on covering or potential, proposed by Kuhlman et al. [60].

A variant of this node blocking problem is introduced by Zhang et al. [61], in which there is a misinformation source set M, and a target node $t \in V$ that is to be protected. For this problem, the greedy algorithm *Minimum Monitor Set Construction* (MMSC) is proposed. Amoruso et al. [62] has developed a heuristic to find a set of blocked nodes D that makes use of an unbalanced cut, introduced by Hayrapetyan et al. [63], to disconnect the target t from a contracted misinformation source node m^* in the graph. Given this cut, a bipartite graph C is defined as the set of edges removed by the cut, and their corresponding vertices. The heuristic finally returns a minimum vertex cover of this graph C. Another variant proposed by Zhang et al. [64] explores settings in which information regarding the misinformation source set is absent or partial. Newman et al [65] shows empirically that targeted node blocking works better than random node blocking in a simulated computer network through which an email virus spreads.

A variation of the influence minimization problem is truth campaigning. This problem is discussed in Section 2.4.1. Another variant of the influence minimization problem is the edge blocking problem, in which we remove edges rather than nodes. This variant is discussed in Section 2.4.2.

2.4.1 Truth campaigning

Based on the multi-campaign influence maximization problem, Budak et al. [66] introduced truth campaigning as an approach to influence blocking maximization. In the truth campaigning problem we have two campaigns that spread simultaneously, a misinformation campaign and a debunker campaign. The goal of truth campaigning is to find a set of initial debunkers $D \subseteq V$ for which $|D| \leq k$, such that the eventual spread of misinformation is minimized, given a network G, a set of initial misinformation spreaders $M \subset V$, and a constant $k \in \mathbb{N}$. This paper proposes three heuristics based on degree centrality, how early a node is expected to become infected, and the extent to which it propagates misinformation after becoming infected. These heuristics are shown to perform comparably to a greedy algorithm.

Several variations of the truth campaigning approach have been investigated. Nguyen et al. [67] introduced a time constraint and experimented with missing information regarding the set M, Li et al. [68] introduced a non-persistent contaminated set, Fan et al. [69] focuses on communities in the network, and Zhang et al. [70] and Hosni et al. [71] extend the approach to not only minimize the spread of misinformation, but also maximize the spread of positive influence.

2.4.2 Edge blocking for influence blocking maximization

Kimura et al. [72] introduced the approach of influence minimization by edge blocking as an edge-focused alternative to node immunization. Edges are blocked by removing them from the network to ensure they can no longer propagate influence. The selection of edges is based on the contamination degree of the network after the removal of an edge. A greedy edge selection method is given by Zhang et al. [73], in which the weight of an edge is the product of the degrees of the nodes on this edge. A different heuristic is proposed by Kuhlman et al. [41] that aims to minimize the number of blocked edges and protect all salvaged nodes by removing a set of edges attached to misinformation source nodes. This heuristic is shown to provide better blocking performance compared to previous works. A salvaged node is a node whose infection is avoided by the choice of blocked edges. The term of salvageable nodes is introduced by Kuhlman et al. [60]. Yao et al. [74] introduce heuristics based on betweenness centrality and out-degree, and show that these heuristics reduce running time.

2.5 Fairness in SNA

Early research into fairness for influence maximization was done by Ali et al. [75], who proposed a fairness constraint that forces the disparity between communities to be below a given maximum. For this constraint, we calculate the proportion of influenced nodes for each community. We define the disparity as the difference between the highest and the lowest of these proportions. Stoica et al. [76] shows that taking into consideration the protected feature on which the communities are based leads to fairer outcomes with higher efficiency. It is also shown by Stoica et al. [77] that promoting equality, as defined by Farnadi et al. [78], often leads to a fairer outcome if the seed set is sufficiently large. However, if the seed set is too small, the improvement of fairness leads to a slight decrease in efficiency. Fairness aware influence minimization was introduced in [79]. A node selection method was proposed that makes use of maximin to promote a fair selection of blocked nodes. For thoroughly studying the group fairness of this method, [79] developed the HICH-BA model, which generates synthetic networks and takes many parameters related to community structure.

3 Research Questions

This section is split up into two parts. Firstly, in Section 3.1, we define the research problem and the corresponding questions that we aim to answer in this thesis. We subsequently give our motivation for these research questions. Secondly, in Section 3.2, we present our methodology, including our approach, for answering the research questions.

3.1 Research problem

In this thesis, we will look at Time-Critical Influence Blocking Maximization by Edge Removal (TCIBM-ER). Specifically, we will focus on the trade-off between influence minimization and group fairness. We refer to the negative influence that we try to minimize as misinformation. We have the following research question:

Given a size $k \in \mathbb{N}$, a network G, and a set of initial misinformation nodes $M \subset V$, how do we select a set of edges $D \subseteq E$ with $|D| \leq k$, such that the number of nodes reached by the misinformation is minimized?

We have the following sub-questions:

1. How fair are the proposed influence minimization methods?
2. What is the effect of changing the number of removed edges on the fairness and influence blocking performance of the proposed influence minimization methods?
3. How do the fairness and influence blocking performance of the proposed influence minimization methods change during the diffusion process?

By answering our first sub-question, we can rank our proposed edge removal methods from most fair to least fair. By combining these results and the influence blocking performance of the proposed methods, we can confirm whether or not there is indeed a trade-off to be made between fairness and blocking performance. Additionally, we will know which methods are best suited for optimizing fairness and blocking performance respectively.

By answering our second sub-question, we can conclude which methods can best be selected for optimizing fairness and blocking performance based on the number of edges we wish to remove. Additionally, the result may aid us in deciding the number of edges we wish to remove from the network in order to save a given proportion of nodes.

By answering our third sub-question, we can see the fairness and blocking performance of the proposed methods in a time-critical situation. This result may be different from the fairness and blocking performance of these methods in absence of such a temporal constraint.

3.2 Methodology

We propose the following approach for our influence minimization problem. Given a network and a set of initial misinformation nodes, we assign a weight to each of the edges in the network. These weights are calculated using one of four proposed heuristics to estimate the importance of edges for the diffusion of influence. Given these weights, we use one of three proposed selection models to select a subset of edges to be removed from the network. One of these proposed selection models is fairness-agnostic, and selects edges based only on the assigned edge weights. The other two proposed edge selection models are fair models. These models select a set of edges sequentially based on the assigned edge weights, as well as the community structure of the network. In each of these edge selection iteration, the model first selects a community from which an edge will be selected. From this community, the edge with the greatest weight is selected. The first fair edge selection method selects the community for each iteration deterministically to ensure that the number of edges that is selected for each community is proportional to the community size. The second fair edge selection method selects the community for each iteration probabilistically, in which the selection probability of a community is proportional to the community size. This community structure is detected using a community detection algorithm. Each proposed method is the combination of one heuristic to compute the edge weights, and one model to select the set of removed edges. Since we have four heuristics and three selection models, this gives us a total of twelve influence minimization methods to consider.

We evaluate the influence minimization performance of the methods by comparing the number of nodes that is reached by misinformation in two situations. In the first situation, we simulate the diffusion of misinformation through the complete network. In the second situation, we simulate the diffusion of misinformation through a version of the network in which a set of edges has been removed. Since our diffusion models and edge selection methods are not deterministic, we average these results over a number of repetitions.

For evaluating the fairness of a method, we define a fairness metric that looks at the maximum disparity between the proportion of saved nodes in any two communities. Additionally, we also use maximin fairness that is defined as the lowest proportion of saved nodes in any community. We define saved nodes as those nodes that are reached by misinformation in the original network, but not reached by misinformation in the version of this network from which edges have been removed. Using this fairness metric we can answer the first sub-question.

The experiments we use to determine the fairness and the influence minimization performance can be split into two sets. The first set of experiments is focused on the effect of changing the number of removed edges. The purpose of this set of experiments is to answer the second sub-question. In these experiments, we allow the diffusion model to run until it converges to a final set of influenced nodes. We run these experiments with different sizes for the set of blocked edges. The second set of experiments is focused on the effect of a temporal constraint. The purpose of this set of experiments is to answer the third sub-question. In these experiments, we set the number of blocked edges to a constant value. Rather than considering only the number of nodes that is reached by the misinformation after the diffusion model converges to a final set, we consider the number of nodes that is reached by the misinformation in every timestep of the diffusion process.

4 Approach

In this section, we discuss the outline of our experiments. We start by giving a brief overview of our experimental set-up in Section 4.1. Subsequently, we discuss our fairness-agnostic approach for minimizing the spread of misinformation in Section 4.2. Based on the fairness-agnostic approach, we propose a fairness-aware approach in Section 4.3. Finally, we define our approach of evaluating the proposed methods. This includes the new HICH-BA2 network generation model, and the evaluation measures for influence minimization performance, fairness, and the accuracy of these experimental results.

4.1 Overall approach

At the start of an experiment, we are given a network G, a maximum number of removed edges k, a maximum number of timesteps T, and a desired number of misinformation nodes. The network we use for our experiments can be one of two types of networks. The first type is the real world networks, taken from a network science database. The second type is the synthetic networks, which are networks that are generated by a model. The synthetic networks in our experiments are generated by the HICH-BA2 model we propose. We start by selecting a set of nodes $M \subset V$. This is the set of initial misinformation nodes.

Our objective is to select a set of edges $D \subseteq E$ with $|D| \leq k$, such that the spread of misinformation in minimized. This selection process consists of two steps. Firstly, each edge in the network is given a weight, which represents the importance of this edge in the diffusion process. This edge weight is calculated by one of the four proposed heuristics. Secondly, based on these edges, a subset of edges D is selected by one of the three proposed edge selection models. One of these models is fairness-agnostic and simply selects the edges with the highest weights. The other two models are fairness-aware. These models use a community detection model to determine the community structure in the network. Given this community structure, the fairness-aware models select the set of edges D in such a way that the number of edges selected from each community is proportional to the total number of nodes in that community.

In order to determine the effect of the edge removal method, we start by simulating the diffusion of misinformation from the set M through the complete given network. This diffusion is simulated by one of two proposed diffusion models. Following this diffusion simulation, we remove the selected set of edges D from the network. The diffusion of misinformation is simulated again for this new version of the network.

We define the influence minimization performance of the methods as the difference in the number of nodes that are reached by misinformation before and after the set of edges D is removed from the network. This number is normalized to make comparison between the experimental results easier if networks of different sizes are used. To evaluate the fairness of the methods, we consider the proportion of saved nodes for each community. If no ground truth communities are available, we use the community structure found by the community detection algorithm instead. We define the fairness of a method to be the maximum community disparity, i.e., the difference between the lowest and the highest proportion of saved nodes in any community. Finally, we have a combined metric for both influence minimization performance and fairness, called maximin. This metric is the lowest proportion of saved nodes in any community. We aim to maximize this value.

Since not all steps are deterministic, we average the results of our experiments over a number of repetitions. To ensure the accuracy of our resulting average, we compute the 95% and 99% confidence intervals for our results.

4.2 Influence blocking

In this section, we discuss several existing methods used for influence misinformation. In Section 4.2.1, we discuss the method used to select our initial set of misinformation nodes M. In Section 4.2.2, we discuss the selection of the set of edges D to be removed from the network. Finally, in Section 4.2.3, we discuss the two models used for simulating the diffusion of misinformation through the network.

4.2.1 Selecting the misinformation node set

There are two different methods for selecting a misinformation set $M \subset V$ with a given size $|M|$. The first method is randomly selecting a subset M with uniform probability. The second method is to select the nodes with the highest degree. In our experiments, we select the set M by highest degree.

We create this degree-based set by first ordering the node degrees in descending order, and determining the cut-off value d_{cutoff}, which is the degree at position x_{cutoff} with $x_{\mathsf{cutoff}} = |M|$. We then create a subset $M' = \{v \in V \mid d_v > d_{\mathsf{cutoff}}\}$. These nodes are guaranteed to be part of M. We then make a second set $M'' = \{v \in V \mid d_v = d_{\mathsf{cutoff}}\}$ of nodes that may become part of M. From M" we randomly select a subset $M''' \subseteq M''$ with $|M'''| = |M| - |M'|$. Finally, we define M as $M = M' \cup M'''$.

4.2.2 Selecting the blocked edge set

Our objective is to select a blocked edge set D. This set will be removed from the network to minimize the spread of misinformation. We select the blocked edge set based on the value given by one of four proposed heuristics. The four heuristics we compare are listed and defined below.

The degree heuristic, which is proposed by Zhang et al. [73], uses the degree weight function in Equation 1 to compute a weight for each edge $(u, v) \in E$.

$$w_{\mathsf{degree}}(u, v) = d_u \cdot d_v \tag{1}$$

We define the contamination heuristic based on the contamination degree used in the greedy approach proposed by Kimura et al. [72]. It is defined as seen in Equation 2. This equation computes the contamination degree of an edge e as the number of nodes that will be disconnected from M if e is removed. In this equation, $G(e)$ is the network G from which the edge e is removed, and $d(m, v, G(e))$ is the distance between nodes m and v in this network $G(e)$.

The distance between two disconnected nodes is considered to be infinite.

$$w_{\textsf{contamination}}(e) = |\{v \mid v \in V; \forall_{m \in M} : d(m, v, G(e)) = \infty\}| \tag{2}$$

The betweenness heuristic proposed by Yao et al. [74] uses the betweenness centrality measure to rank edges. We select the edges with the highest betweenness centrality to form the set D of blocked edges. Betweenness centrality is computed as seen in Equation 3, where σ_{st} is the number of shortest paths from s to t, and $\sigma_{st}(e)$ is the number of those paths that go through the edge e.

$$w_{\textsf{betweenness}}(e) = \sum_{s,t \in V} \frac{\sigma_{st}(e)}{\sigma_{st}} \tag{3}$$

The propagation degree heuristic is derived from the out-degree based method proposed by Yao et al. [74]. This method starts by finding the first reachable set, which are all the nodes $v \notin M$ that have an edge connecting to a node $u \in M$. We only consider blocking edges from the out-edge set, which we define as the set of all edges between the initial misinformation set M and the first reachable set. For each of these nodes v, we count the number of initially inactive neighbors $w \notin M$. This is the number of nodes that may be saved when v is disconnected from the nodes in M. In the original paper, the weight of an edge (u, v) is computed with a variant of betweenness centrality using source set M and target set $V \setminus M$. However, since we already have a betweenness heuristic and this adaptation has proved computationally expensive, we use a new heuristic for the propagation degree method.

In our propagation degree heuristic, we divide the number of initially inactive neighbors of v by the number of initially active neighbors of v. We do this, because the the number of initially active neighbors is equal to the number of edges that must be blocked to disconnect node v from these neighbors, which means each of these edges only contributes a part of the expected decrease of influence. The final computation for an edge (u, v) with $u \in M$ and $v \notin M$ is seen in Equation 4.

$$w_{propagation}(u, v) = \frac{|\{w \mid w \in N(v) \land w \notin M\}|}{|\{w \mid w \in N(v) \land w \in M\}|} \tag{4}$$

Given the weights produced by one of these heuristics, we select a set of blocked edges D. To generate our set D, we start with an empty set $D = \emptyset$. We define a set of options $E' = E$. We iteratively add edges to this set until $|D| = k$. Each iteration we add to D an edge $e \in E'$ with maximal weight $w_e = w_{\textsf{max}}$, and remove this edge from E'. If there are multiple edges sharing the current maximal weight in E', one of these edges is selected randomly.

4.2.3 Diffusion of influence

The diffusion of influence starts from the initial misinformation nodes M. We denote this as $A_0 = M$, where A_t is the set of active nodes at timestep t. Each of the diffusion models takes as its input the network G, the initial misinformation set M, a maximum number of timesteps T, and optionally a set of threshold values. The output of the model is the set A_T. The two diffusion models we use in our experiments are the Linear Threshold model and the Independent Cascade model.

The Linear Threshold diffusion model was proposed by Kempe et al. [26]. The pseudocode is given in Algorithm 1. In this model, each node has an activation threshold, and will be activated when the sum of weights of its active neighbors exceeds this threshold. For the experiments, the thresholds are selected uniformly from $\langle 0, 1\rangle$, and the influence weight from node u to node v is $\frac{1}{|N(v)|}$. Given a set of thresholds, the diffusion model is deterministic.

Algorithm 1 Linear Threshold(G, M, T, $threshold$)

```
current ← M
t ← 0
while current ≠ ∅ and t < T do
    neighborhood ← ∅
    for v ∈ current do
        for w ∈ N(v) do
            if w ∉ previous ∪ current then
                neighborhood ← neighborhood ∪ {w}
    next ← ∅
    for w ∈ neighborhood do
        active_neighbors_w ← {u ∈ N(w) | u ∈ previous ∪ current}
        influence_w ← |active_neighbors_w| / |N(w)|
        if influence_w > threshold_w then
            next ← next ∪ {w}
    previous ← previous ∪ current
    current ← next
    t ← t + 1
return previous ∪ current
```

The Independent Cascade model, proposed by Kempe et al. [26], propagates influence individually and sequentially. In each iteration t, every node that was activated in iteration $t-1$ will attempt to activate each of its inactive neighbors. The probability of activating an inactive node is a constant $p_{v,w}$ that is uniformly sampled from $\langle 0, 1\rangle$ for each edge (v, w) before diffusion starts. The pseudocode of the Independent Cascade model is given in Algorithm 2.

4.3 Fairness-aware influence blocking

We propose two fairness-aware influence blocking methods. Both these methods use knowledge of the community structure in the network. The detection of these communities is discussed in

Algorithm 2 Independent Cascade(G, M, T)

```
previous ← ∅
current ← M
t ← 0
while current ≠ ∅ and t < T do
    next ← ∅
    for v ∈ current do
        neighbors_v ← {w ∈ N(v) | w ∉ previous ∪ current ∪ next}
        for w ∈ neighbors_v do
            Add w to next with probability p_{v,w}
    previous ← previous ∪ current
    current ← next
    t ← t + 1
return previous ∪ current
```

Section 4.3.1. Given the community structure and the edge weights computed by one of the four proposed heuristics, both of these methods use a fairness-aware model for selecting the blocked edge set. These models are discussed in Section 4.3.2.

4.3.1 Community detection

The community structure is used in selecting the blocked edge set D. We use the Louvain algorithm to detect these communities. This algorithm was introduced by Blondel et al. [80], and used by Bierbooms [23] for fairness-aware influence blocking. We have chosen this algorithm to stay consistent with the related research of Bierbooms [23], who shows that this method achieves the highest modularity value and leads to the best results. For the real world networks, we also use the communities detected using the Louvain algorithm to evaluate the fairness of the different methods. For our synthetic networks, we use the ground-truth communities for evaluation instead.

4.3.2 Fairness-aware blocked edge set selection

The fairness-agnostic model selects the blocked edges sequentially by choosing those edges with the highest weights. We introduce two fairness-aware models that aim to improve the community fairness that results from selecting the blocked edges. In one of these models, the community from which we select the next edge is decided deterministically. In the other model, we select this community probabilistically.

The deterministic fairness-aware model limits each of the selection iterations to a subset of edges $E' \subset E$ that connect to at least one node in a given community or set of communities. The deterministic fair edge selection approach consists of two stages. In the first stage we aim to maximize the number of communities for which the set D contains at least one edge. Once D contains at least one edge for each community, we select our set E' to better match the node community proportions r. This model takes as its parameters the set of edges E, the set of communities C, the desired number of blocked edges k, the weights $\vec{w}$ of all the edges, the communities $\vec{c}$ of the nodes, and the node community proportions $\vec{r}$ in the network. The pseudocode is given in Algorithm 3.

Algorithm 3 Deterministic Selection $(E, C, k, \vec{w}, \vec{c}, \vec{r})$

$D \leftarrow \emptyset$
$E' \leftarrow E$ ▷ Set of considered edges
$C' \leftarrow C$ ▷ Set of considered communities
for $c \in C$ **do**
    $d_c \leftarrow 0$ ▷ Number of edges in D per community
while $C' \neq \emptyset$ **and** $|D| < k$ **do**
    $w_{\max} = \max(\{w_e \in \vec{w} \mid e \in E'\})$
    $e \xleftarrow{\$} \{e \in E' \mid w_e = w_{\max}\}$ ▷ Randomly select edge e from E' with $w_e = w_{\max}$
    $D \leftarrow D \cup \{e\}$
    $(u, v) \leftarrow e$
    $C' \leftarrow C' \setminus \{c_u, c_v\}$
    $d_{c_u} \leftarrow d_{c_u} + 1$
    $d_{c_v} \leftarrow d_{c_v} + 1$
    $E' \leftarrow \{(u, v) \in E \setminus D \mid c_u \in C' \vee c_v \in C'\}$
for $c \in C$ **do**
    $k_c \leftarrow 2 \cdot k \cdot r_c$ ▷ Desired number of edges per community
    $d'_c \leftarrow k_c - d_c$
while $|D| < k$ **do**
    $d'_{\max} \leftarrow \max(\{d'_c \mid c \in C\})$
    $C' \leftarrow \{c \in C \mid d'_c = d'_{\max}\}$
    $E' \leftarrow \{(u, v) \in E \setminus D \mid c_u \in C' \vee c_v \in C'\}$
    $e \xleftarrow{\$} \{e \in E' \mid w_e = w_{\max}\}$
    $D \leftarrow D \cup \{e\}$
    $(u, v) \leftarrow e$
    $d'_{c_u} \leftarrow d'_{c_u} - 1$
    $d'_{c_v} \leftarrow d'_{c_v} - 1$
return D

The probabilistic fairness-aware model also selects edges sequentially. At the start of the selection procedure, we set the community probability r_c to be the proportion of nodes that belongs to community c. For each selection iteration, we randomly select a community c with probability r_c. We limit the edge selection to the set of edges for which at least one node belongs to this community c. If there are no more edges containing a node with this community, we select a new community and try the edge selection again. We repeat this process until an edge has been selected.

In both of these fairness-aware blocked edge selection models, an edge is said to belong to a community if at least one of its nodes belongs to this community. This means that each edge can belong to either one or two communities. Since we do not force the community proportions of the edges in D to match the node community proportions r, it is in theory possible to obtain a set D in which all edges belong to the same community c_1. For example, an edge that was added to E' because it belongs to c_2 may also belong to this community c_1. The reason we still choose to select our edges this way, rather than forcing our community proportions in D to better match r, is because the node community proportions do not necessarily match the edge community proportions in the network. For a community in which the nodes have a higher average degree the proportion of edges belonging to this community will be higher. By selecting edges based on just one community, we expect the community proportions of the edges in D to naturally converge to the edge community proportions in the network.

4.4 Model evaluation

To test the proposed influence blocking methods, we use real world networks and synthetic networks. For generating the synthetic networks, we propose the new HICH-BA2 model in Section 4.4.1. The metrics used to measure the blocking performance and fairness of the influence blocking methods, and the accuracy of our results, are discussed in Section 4.4.2.

4.4.1 Network generation

For researching the effect of biases in network structure, Karimi et al. [81] and Lee et al. [82] have developed the *Homophily BA* model for network generation, in which BA stands for Barabási–Albert. This model combines the property preferential attachment attribute proposed by Barabasi et al. [58] with homophily. The term homophily is defined as the tendency to associate with similar others, and it leads to the community structure in the resulting networks. To promote the formation of inter-community links, Wang et al. [83] proposed *Diversified Homophily BA*. This model adds diversified links to a network generated using the Homophily BA model.

To create synthetic networks, we use an improved version of the HIgh Clustering Homophily Barabási-Albert (HICH-BA) model proposed by Bierbooms et al.[23], which we call HICH-BA2. The pseudocode for this model is given in Algorithm 4. The model makes use of a function `get_probability_distribution`, which is seen in Algorithm 5.

The HICH-BA2 model takes the following seven parameters. n and m are the number of nodes and edges respectively. r is the list of desired community proportions r_c for each community c. h is the homophily probability, i.e. the probability of adding an edge between two nodes that belong to the same community. p_t is the probability of creating a triangle when adding an edge to the network. Higher values will lead to increased clustering. p_{PA} is the probability of using preferential attachment when selecting a target node for a new edge. Preferential attachment uses node degree to calculate the weight of a node, as opposed to uniform weights. α is the preferential attachment strength when using preferential attachment to calculate weights. The steps of the model are listed below.

1. First we initialize the network by adding one node to each community where the number of communities are equal to the length of r.
2. We define our probability p_N as the number of remaining nodes to add divided by the desired number of edges m.
3. If the number of nodes or number of edges in the network are less than the required number of nodes and edges, respectively, then we determine if we will add and connect a node, or add only an edge to the network. We do this as follows:

(a) If $n - |V| = m - |E|$ (i.e. an equal number of nodes and edges remain to be added), we must add and connect a node. To add and connect a new node, we follow the following steps:

   i. We randomly select a community c with probability r_c.
   ii. We add a new node called $source$ to the network with community c.
   iii. We define a set $options$ as the set of all other nodes with community c. This is the set of nodes to which we may later connect our new node $source$.
   iv. We generate a probability distribution for the set $options$.
   v. Given this probability distribution, we randomly select a node $target$ from $options$.
   vi. We add a new edge between $source$ and $target$.

(b) If $|V| = n$ (the number of nodes $|V|$ is equal to the desired number of nodes n), we must add only an edge.

   i. With probability p_t, we add a triad closure edge as follows:
      A. We define V_2 as the set of nodes with a degree of at least 2.
      B. We generate a probability distribution for the set V_2.
      C. Based on the probability distribution we randomly select a node u from V_2.
      D. We randomly select a neighbor of u with uniform probability, and call this node $source$.
      E. We define a set of nodes opt_all as the set of nodes that is connected to u, but not connected to $source$.
      F. With probability h, we define the set $options$ as the subset of opt_all belonging to the same community as $source$. Otherwise, we define the set $options$ as the subset of opt_all belonging to any other community than $source$.
      G. If the set $options$ is empty, we define $options$ as the set of all nodes in opt_all.
      H. Select target node and add an edge using steps 3(a) (iv)-(vi).
   ii. With probability $1 - p_t$, we add an edge as follows:
      A. We define a set opt_source as the set of all nodes that are not already connected to all other nodes in the network.
      B. We generate a probability distribution for the set opt_source.
      C. Given this probability distribution, we randomly select a node $source$ from opt_source.
      D. We define the set opt_all as the set of nodes that is not connected to $source$. After this follow steps 3 (b)(i) F-H.

(c) If neither of the conditions mentioned above are true, we add and connect a node with probability p_N by following the steps mentioned in 3(a), and with probability $1 - p_N$, we add only an edge following the steps mentioned in 3(b).

A probability distribution is created as follows:

1. With probability p_{PA}, we generate the probability distribution using preferential attachment. Otherwise we generate the probability distribution without preferential attachment.

(a) If we use preferential attachment, each node v is given a weight $(d_v)^\alpha + 1$. This list of weights is normalized to make a list of probabilities.

(b) If we do not use preferential attachment, each node is given the weight 1, which is normalized to create a uniform probability distribution.

Algorithm 4 HICH-BA2 $(n, m, r, h, p_t, p_{PA}, \alpha)$

```
num_com ← length of r
G ← empty undirected graph
for i: 0 to num_com − 1 do
    add a new node v to G with c_v = i
p_N ← (n − num_com)/m
while |V| < n or |E| < m do
    current_max_edges ← |V| · (|V| − 1) · 0.5
    must_add_node ← |E| = current_max_edges or m − |E| = n − |V|
    must_add_edge ← |V| = n
    try_add_node ← True with probability p_N
    add_node ← must_add_node or (not must_add_edge and try_add_node)
    if add_node then
        c ← community that is randomly selected with probability r_c
        add a new node source to G with c_source = c
        options ← {v ∈ V | c_v = c} \ {source}
    else
        can_close_triangle ← True if there exists a node v ∈ V with d_v ≥ 2
        try_close_triangle ← True with probability p_t
        close_triangle ← can_close_triangle and try_close_triangle
        if close_triangle then
            V_2 ← {v ∈ V | d_v ≥ 2}
            probdist ← get_probability_distribution(V_2, p_PA, α)
            u ← random node from V_2 with probability probdist_v
            source ← random neighbor of u with uniform probability
            opt_all ← {v ∈ N(u) | v ∉ N(source)} \ {source}
        if (not close_triangle) or opt_all = ∅ then
            opt_source ← {v ∈ V | deg_v ≠ |V| − 1}
            probdist ← get_probability_distribution(opt_source, p_PA, α)
            source ← random node from opt_source with probability probdist_source
            opt_all ← {v ∈ V | v ∉ N(source)} \ {source}
        try_intra_edge ← True with probability h
        if try_intra_edge then
            options ← {v ∈ opt_all | c_v = c_source}
        else
            options ← {v ∈ opt_all | c_v ≠ c_source}
        if options = ∅ then
            options ← opt_all
    probdist ← get_probability_distribution(options, p_PA, α)
    target ← random node from options with probability probdist_target
    add a new edge to G between source and target
return G
```

The get_probability_distribution function takes as its parameters the probability and strength preferential attachment, and a subset of nodes $V' \subseteq V$. We decide to use preferential attachment with probability p_{PA}. If we do, each node gets a connection probability based on its degree and the preferential attachment strength α. Otherwise, each node is given the same probability $\frac{1}{|v'|}$.

Algorithm 5 get_probability_distribution(V', p_{PA}, α)

```
1: PA ← True with probability p_PA
2: if PA then
3:     probdist_v ← (d_v)^α + 1 for v ∈ V'
4: else
5:     probdist_v ← 1 for v ∈ V'
6: normalize list probdist
7: return probdist
```

A few changes and additions are made to the HICH-BA model to obtain the current HICH-BA2 model to avoid errors, and to improve or expand functionality. These changes are listed and explained below.

The probability p_N of adding a node to the graph is changed from $p_N = 2n/m$ to $p_N = (n - num\_com)/m$. Since in each iteration we either add only an edge, or we add a node and connect it with an edge, the expected number of edges we get is equal to the number of iterations T it takes to generate the network. The network generation continues until the number of nodes $|V|$ is equal to the desired number of nodes n. These iterations start after adding an unconnected node for each community. Since $|V| = num\_com + T \cdot p_N$, we calculate the expected number of timesteps T' as $T' = (n - num\_com)/p_N$. To get the expected number of edges $m' = T'$, we therefore compute p_N as $p_N = (n - num\_com)/m$, where m is the desired number of edges. For any other value of p_N, we get $m' \neq m$, so the expected number of edges m' is not the same as the desired number of edges m.

The condition for the main while loop now also contains $|E| < m$, and the choice between adding a node or adding an edge is extended using the Boolean variables $must\_add\_node$ and $must\_add\_edge$. This is done to make sure that a generated network contains exactly n nodes and m edges, and the chosen option is possible. Furthermore, a single edge can not be added if $|E| = |V| \cdot (|V| - 1) \cdot 0.5$, since each pair of nodes would already be connected.

The path for not creating a triangle is now also reached if $options = \emptyset$ after trying to create a triangle. This happens when $N(source)$ - $\{v\} \subseteq N(v)$. In this notation, $N(x)$ is the neighborhood of x, i.e. the set of nodes $\{y \in V \mid (x, y) \in E\}$.

The value α is added to the calculation of the weights in the case of preferential attachment. This value is the preferential attachment strength. Setting this value to $\alpha = 1$ would give the same weights as the calculation done by the HICH-BA model.

4.4.2 Evaluation measures

We will evaluate the general performance of the methods using the proportion of nodes we can save, and the fairness is evaluated using inter-community disparity. A combined metric called Maximin looks at the performance of the blocking method for the least protected community. We expect the fair methods to have a better group fairness, but at the cost of a slightly lower general performance.

We denote the spread of influence at timestep t as $\sigma_t(G, M)$ for a network G and corresponding initial misinformation node set M. This spread is defined as $\sigma_t(G, M) = |A_t|$, where A_t is the set of activated nodes at timestep t. The effect of removing the edge subset D from a network is measured by looking at the number of saved nodes. This number of saved nodes, denoted $\Delta\sigma_t(G, M, D)$, is calculated as the difference in spread before and after removing D from the network, as shown in Equation 5. For this comparison, we denote the set G from which the subset of edges D is removed as $G(D)$. To make the comparison of results from different networks easier, we normalize our result by dividing it by the number of nodes in the network. This gives us a saved node proportion, as calculated by Equation 6.

$$\Delta\sigma_t(G, M, D) = \sigma_t(G, M) - \sigma_t(G(D), M) \tag{5}$$

$$\pi_{\Delta\sigma_t}(G, M, D) = \frac{\Delta\sigma_t(G, M, D)}{|V|} \tag{6}$$

Since the edge selection methods and the diffusion models are nondeterministic, we define our performance value as the average number of saved nodes over multiple repetitions. Each repetition, a new set D is selected, and the influence diffusion is simulated.

A higher saved proportion indicates a better performance of the method that selects the set of blocked edges D. Since the node thresholds of the Linear Threshold model and the propagation probabilities of the Independent Cascade model are generated randomly before the diffusion process starts, we use one set of values per iteration. This way, the difference between the resulting saved node proportions before and after removing D are not affected by the random values in the diffusion model, and can be attributed to the removal of D from the network.

To evaluate the fairness of the influence blocking methods, we use the disparity and maximin metrics. For both metrics, we compute a separate saved node proportion for each community, looking only at the nodes belonging to that community. These proportions are obtained by limiting the activated set A to A_c, which contains only nodes belonging to community c. We denote the set of activated nodes in a community c at timestep t as $A_{t,c}(G, M)$, where G is the network, and M is the initial misinformation set. We define the number of saved nodes for a community c as the number of nodes that is reached by the influence in the original network, but not reached by the influence in the adapted network. The new calculations are shown in Equation 7 and Equation 8. Just like the general performance metric, we also average the saved community proportions over iterations.

$$\Delta\sigma_{t,c}(G, M, D) = A_{t,c}(G, M) \setminus A_{t,c}(G(D), M) \tag{7}$$

$$\pi_{\Delta\sigma_{t,c}}(G, M, D) = \frac{\Delta\sigma_{t,c}(G, M, D)}{|\{v \in V | c_v = c\}|} \tag{8}$$

The disparity metric looks only at group fairness. We calculate the disparity as the difference between the highest and lowest saved proportion, as shown in Equation 9. This disparity is the difference between the proportions of saved nodes in the most protected and the least protected community. We aim to minimize this disparity. Since there is a big chance that some communities are affected neither before nor after removing edges, we use a subset C', defined as $C' = \{c \in C \mid \exists_{v \in V} : c_v = c \wedge v \in A_G\}$. This is the set of communities that contain at least one node which was activated before the set D was removed from the network. The decision to use this subset of communities is further explained in Section 5.2.2.

$$\text{disparity} = \max_{\substack{c_1, c_2 \in C' \\ c_1 \neq c_2}} |\pi_{\Delta\sigma_{t,c_1}}(G, M, D) - \pi_{\Delta\sigma_{t,c_2}}(G, M, D)| \tag{9}$$

The maximin metric combines group fairness with general performance. The maximin value is the lowest of all saved node proportions for the communities, as shown in Equation 10. This metric shows how well protected the least protected community is from misinformation. We aim to maximize this minimum proportion. We only use the initially active communities C' as defined for disparity.

$$\text{maximin} = \min_{c \in C'} \pi_{\Delta\sigma_{t,c}}(G, M, D) \tag{10}$$

For each data point we plot in graphs, we compute the interval range for the confidence levels 95% and 99%. For each graph, we define a highest error value at a given confidence level as the maximum margin of error over all the data points in this graph. Each graph therefore has two highest error values, one for each of the confidence levels. These values are used to estimate the validity of our data. To minimize this error value, we run multiple repetitions per network in which our methods select the blocked edge sets. We require these repetitions and the corresponding calculation of the maximum margin of error due to the probabilistic elements in edge selection and diffusion.

5 Experimental results

In this section, we will discuss the details and results of our experiments. In Section 5.1, we start by giving the experimental setup. In Section 5.2, we cover the initial analysis of the proposed methods. Based on this analysis, we determine which methods are used in our main set of experiments. The results of experiments using synthetic networks are given in Section 5.3. Finally, the results of experiments using real world networks are given in Section 5.4.

5.1 Experimental setup

In this section, we give the details of our experimental setup. Section 5.1.1 discusses the hardware used for network generation and the execution of experiments. Section 5.1.2 discusses the methods we use in our experiments.

5.1.1 Server details

The computational load consists of synthetic network generation, early runs for debugging, and the final experiments. We generate multiple sets of networks of different sizes. Originally, we were planning to take the average results over a set of networks with the same properties. However, early experiments showed that the runtime of a batch of experiments exceeded the expected runtime. For this reason, we use one network for each batch of experiments. Three machines were used for the generation of all networks. The first set of networks was generated sequentially on a Dell Inspiron 15 7000 laptop with a Intel Core i5-8265U CPU. The second set of networks was generated sequentially using the Octiron server, which is one of the generic server machines of the Data Science Lab. It is affiliated with the Leiden Institute of Academic Computer Science (LIACS). The machine has 512GB RAM, and a Dual Intel Xeon E5-2630v3 CPU with 32 threads and a processor base frequency of 2.40GHz. The third set of networks was generated in parallel using the Mithril server, which is the computing machine of the Data Science Lab available to students. This machine has 1TB of RAM and 64 cores of the Intel Xeon E5-4667 v3 that has a total of 128 threads and a processor base frequency of 2.00GHz. The experiments were run in parallel on the Drogium server, which is a machine of the Data Science Lab with 256 threads. It consists of the same hardware as the Mithril server.

5.1.2 Used methods and models

For the generation of synthetic networks, we use the HICH-BA2 network generation model. The initial misinformation set is selected as the set of nodes with the highest degree.

For the initial analysis in Section 5.2 we use an initial misinformation set containing one misinformation node. For the experiments using synthetic networks, we use the sizes $|M| = 5$ for the small network, and $|M| = 50$ for the large network. For the experiments using real world networks, we use the sizes $|M| = 100$ for the Facebook network, and $|M| = 250$ for the Astrophysics collaboration network.

The diffusion models used in the experiments in Section 5.2.1 are the Independent Cascade model the Linear Threshold model. Based on this initial analysis of diffusion models, we decide to use the Independent Cascade model in all other experiments.

The edge blocking method used to in the example in Section 5.2.1 is the fairness-agnostic blocked edge set selection model, using edge weights computed by the degree heuristic. All experiments discussed before Section 5.3.3 use all methods, each consisting of one of the four proposed edge weight heuristics, and one of the three proposed blocked edge set selection models. The experiments in Section 5.3.3 and later sections use only the methods in which the fairness-agnostic model is used to select the blocked edge set.

The influence minimization performance is evaluated as the proportion of saved nodes. The fairness is evaluated using the disparity measure. The maximin value is a combined metric for the performance and the fairness. Based on the conclusion in Section 5.2.2, the disparity and maximin metrics are changed to not consider all communities, but only the affected communities. We define an affected community as a community for which at least one node is reached by misinformation in the diffusion simulation on the original network. This is the set of communities for which it is possible to save at least one node.

Since not all steps in the experiment are deterministic, we average our results over a number of repetitions. The accuracy of these average values is established by calculating the corresponding 95% and 99% confidence intervals. The maximum error of the mean for these confidence levels is included in the experimental results.

5.2 Comparison of Diffusion Models

This subsection covers the initial analysis. A comparison of diffusion models is discussed in Section 5.2.1. Based on this comparison, we decide to use the Independent Cascade model for further experiments. The evaluation of group fairness for influence minimization is discussed in Section 5.2.2. In this section, we decide to exclude unaffected communities from our fairness evaluation metrics.

5.2.1 Linear threshold diffusion models

Before implementing and running the final experiments, we compare different diffusion models. As expected, the probabilistic factors in the diffusion models cause variance in the resulting spread of influence. For any given network with a set M of initially active nodes, we expect the average spread of influence to converge as the number of repetitions increases.

For the Independent Cascade model, removing an edge from the network has the same effect as setting its propagation probability to zero. This leads to a lower spread of influence, and therefore a better performance of the blocking method. We expect a higher number of blocked edges k to result in a better performance of the blocking method. A decrease in this performance with an increasing k could possibly be attributed to the probabilistic factors of cascade propagation in combination with an insufficient number of repetitions.

For the Linear Threshold model we have the same expectations of an increasing mean performance, where a decrease in performance for an increased k could be explained by the randomly generated node thresholds. However, the Linear Threshold model is deterministic for a given set of thresholds. Our finding occurred when it became clear that the performance could also decrease with an increasing number of removed edges k in the case where the thresholds are kept the same, i.e. no probabilistic factors could explain the absence of a positive correlation between the number of removed edges k and the blocking performance of the used edge selection method.

Upon further investigation, we found that removing the k selected edges does not only remove the possibility of node activation between the corresponding nodes, but the remaining edges of these 2 nodes also get an increase in their influence contribution.

For example, a node has four neighbors of which two are active, and its activation threshold is 0.6. This means that a proportion of at least 0.6 of its neighbors must be active for the node itself to become activated. However, in this case the proportion of active neighbors is 0.5, and each of the two neighbors contributes 0.25 to this influence. If an edge to one of the inactive nodes is removed, the proportion of active neighbors is now 0.67, as two of the three neighboring nodes are active. As we see in this example, not only did the influence contribution of the disconnected node decrease from 0.25 to zero, but each of the remaining nodes have their contribution to the proportion of active nodes increased from 0.25 to 0.33. In this new situation, the active proportion 0.67 is now higher than the activation threshold 0.6, and the node will become activated. This leads to an increased spread of influence, and hence a lower performance. A similar example is shown in Figure 4 and Figure 5.

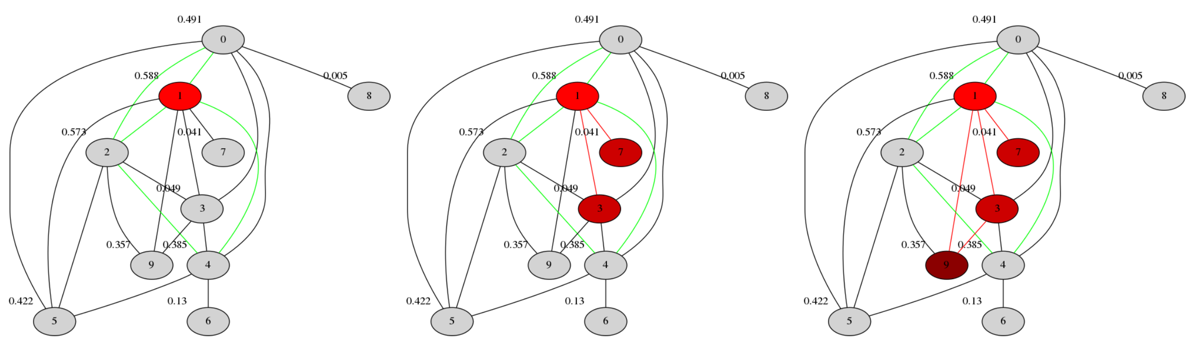

Figure 4: Linear Threshold spread before removing the full green lines. Full red lines are the edges used to influence currently active nodes.

In the cases of the weighted cascade model and the persuasiveness weighted cascade model, the activation probability of a node is inversely correlated to its number of neighbors. Similar to the Linear Threshold model, we can expect that removing the wrong edges can in this way lead to an increased spread of influence, and a decreased mean performance after convergence with a sufficient number of repetitions. For our experiments we use the Independent Cascade model, as it is the simplest variant, and it is the method most used by previous papers.

5.2.2 Evaluation for community fairness

The general performance of the blocking method is measured by calculating the proportion of nodes that are considered saved. This number of saved nodes is defined as the mean difference between the number of activated nodes before and after edges are blocked or removed. The community performance is calculated in the same way, but only looking at the nodes belonging to a given community. The relative performance of different communities is used to define fairness.

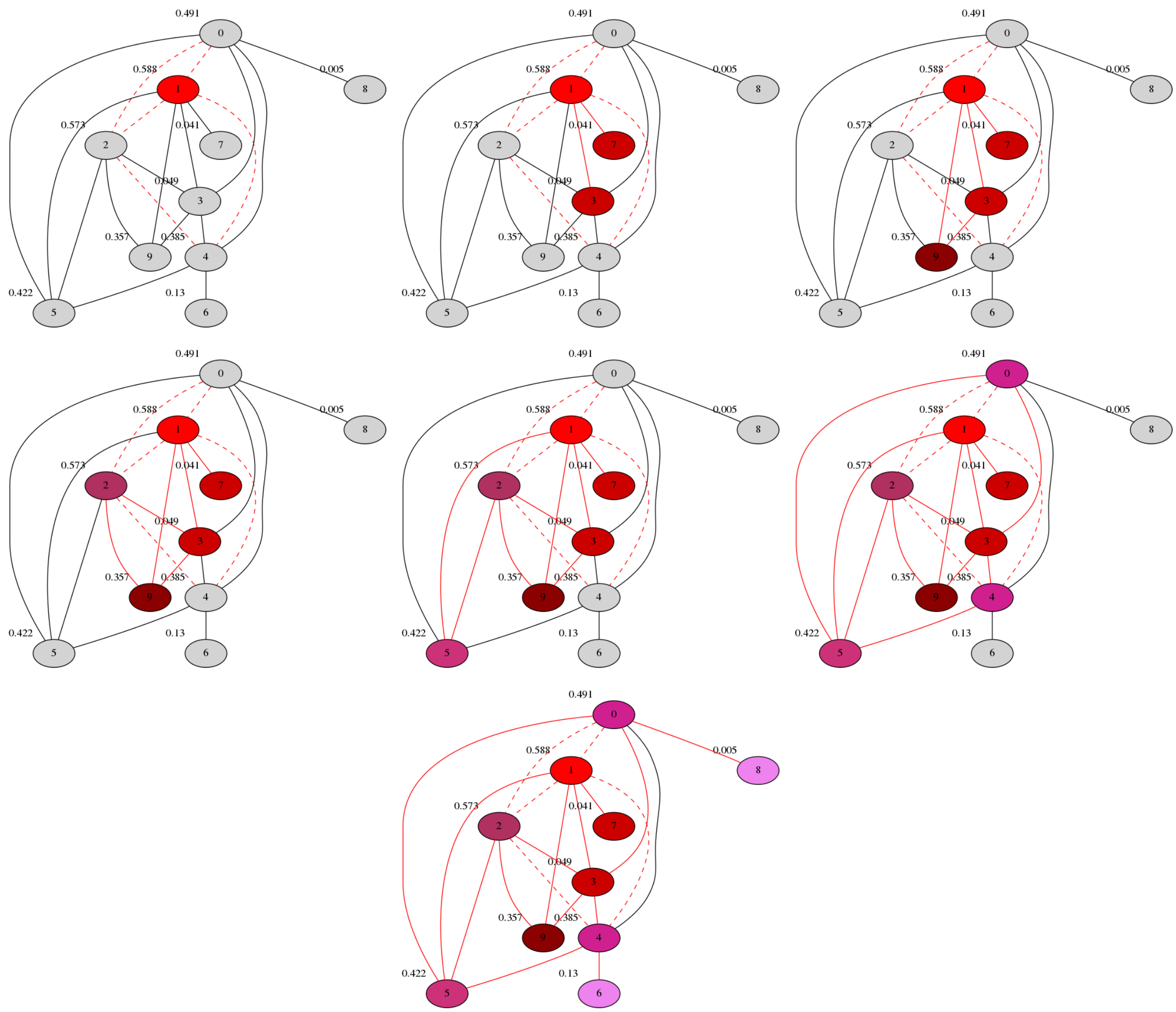

Figure 5: Linear Threshold spread after removing the dotted red lines. Full red lines are the edges used to influence currently active nodes.

A noticeable result is that for large networks, the maximin metric would be very close to zero, and the disparity to one. This is unlike the results we have seen for smaller networks.

Since the Louvain community detection algorithm divides the network into a large number of communities, there are many communities for which none of their nodes are reached by the influence. In any case where there is at least one such community that is reached neither before nor after the removal of edges, the number of activated nodes in both cases are zero, and will thus give a proportion of zero for the saved nodes of that community. This will result in the maximin value zero. In the cases where at least one community has at least one activated node in the original graph, and remains completely inactive after removing edges, this community will have a saved proportion of one. This could be the result of one or two critical edges getting removed from the network. This presence of both a community that is never influenced and a community that is only completely inactive with k edges removed leads to a disparity value of one. In our experiments, our fairness evaluation methods will consider exclusively the communities for which at least one node was activated in the baseline diffusion simulation on the network in which the set D is not yet removed.

5.3 Experimental results on synthetic networks

We test our methods on two different synthetic networks, generated by our HICH-BA2 model. Table 2 shows the generation parameters and final network properties for both networks. We use the abbreviations GT and PA for Ground Truth and Preferential Attachment respectively. Avg WEI is the average Weighted External-Internal Index as defined by Coleman et al. [84].

Network	**Small**	**Large**
Nodes	100	10 000
Edges	500	200 000
Nr Communities GT	2	5
Community proportions	0.3, 0.7	0.54, 0.3, 0.15, 0.005, 0.005
Homophily	0.9	0.9
Triangle Probability	0.9	0.9
PA Probability	0.9	0.9
PA Strength	1.0	1.0
Diameter	5	7
Avg Shortest Path	2.39333	2.97145
Avg Degree	10	40
Nr Communities Louvain	5	101
Avg Clustering Coefficient	0.55381	0.59010
Avg WEI	-0.08228	-0.79229
Assortativity Coefficient	0.72353	0.50609

Table 2: Network properties for synthetic networks

The results of experiments using the small network are presented and discussed in Section 5.3.1. Subsequently, the results of experiments using the large network are presented and discussed in Section 5.3.2. These results show that the used heuristic is much more important than the model used to select the blocked edge set. To simplify our results, we therefore only use the fairness-agnostic blocked edge selection model in future experiments. This model is the fastest of the three proposed blocked edge set selection models. Section 5.3.3 presents and discussed the results of experiments in which only this blocked edge selection model is used. In this set of experiments, the number of repetitions is increased.

5.3.1 Small network

The following figures are the results of experiments run on the small network, averaged over 10000 repetitions. The default values are $T = 10$ and $k = 100$
The marker shape indicates the used heuristic, and the line colors represent selection methods. For each of the corresponding graphs, the maximum error for each metric is below 0.0019 with 95% confidence, and below 0.0025 with a 99% confidence. This shows that our resulting graphs have converged sufficiently to closely represent the actual average values.

Figure 6 shows the proportion of saved nodes over multiple values of k. We see that early on, the different selection methods have little effect on the performance while the heuristics have a clear ranking of which are the best. From best to worst, they are betweenness, contamination, propagation, and degree. Between 100 and 200 we see that the propagation degree heuristic rises to the optimal performance before any other heuristics. This is due to the fact that this heuristic only considers the out-edge set, which we can assume contains between 100 and 200 edges. If each of these out-edges is removed, the set M is disconnected from the rest of the network. The degree heuristic also increases slightly earlier than the betweenness and contamination heuristics. This can be explained by the fact that the set M is selected as the set of nodes with the highest degrees. Therefore, similar to the out-edge set, the degree heuristic also has an advantage here. In the middle of the graph, the contamination heuristic seems to be performing slightly better without fairness considerations, but overall the results seem to depend mostly on the used heuristic.

Figure 7 shows the maximin values. The results mostly appear to match those of the saved proportions, with the difference that any differences in performance from the selection methods seems slightly larger. This may however just be because the y-scale is slightly more stretched.

Figure 8 shows the maximum community disparity. The propagation heuristic performs the worst, followed by the degree heuristic. Not a lot can be said about the influence of the different selection methods here.

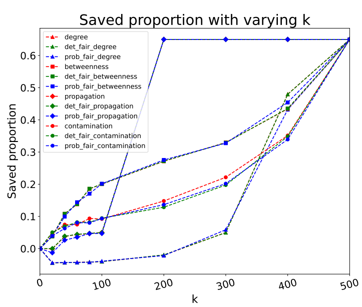

Figure 6: Number of nodes saved in small network with $T = 10$

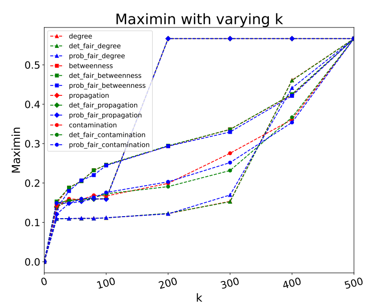

Figure 7: Maximin value in small network with $T = 10$

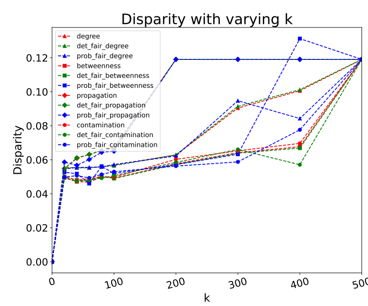

Figure 8: Disparity in small network with $T = 10$

Figure 9 shows the resulting saved proportions over time at $k = 100$. Based on these results, we can say that the graph is converged at $T = 5$. The fair methods perform slightly better than the baseline method, but still the heuristics clearly contribute more to the performance. We see here that some of the methods rise at the start, and go down again after a few timesteps. This is because if the distance between two nodes is shorter, the number of shortest paths between them is lower, which means the removal of edges affects a bigger proportion of them, and is more likely to save nodes at that timestep.

Figure 10 matches the results of the saved proportions and we see that, even though the disparity is increased, the maximin value is also better for the fair selection methods than for the baseline method. This shows that the fair methods do indeed outperform the baseline in this respect.

Figure 11 shows the disparity related to this performance. Here we see that, in all cases, the baseline method has a lower community disparity than either of the fair selection methods.

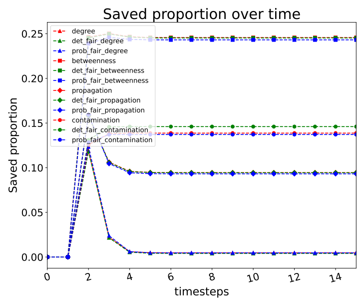

Figure 9: Number of nodes saved in small network with $k = 100$

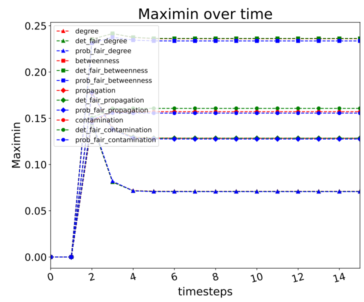

Figure 10: Maximin value in small network with $k = 100$

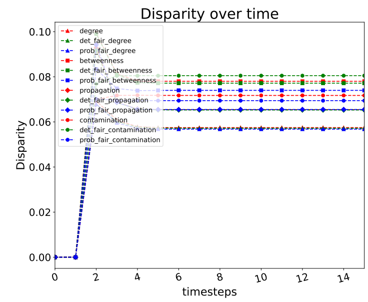

Figure 11: Disparity in small network with $k = 100$

5.3.2 Large network

The following figures are the results of experiments run on the large network, averaged over 250 repetitions. The default values are $T = 15$ and $k = 7500$. The marker shape indicates the used heuristic, and the line colors represent selection methods. For each of the corresponding graphs, the maximum error is below 0.0093 with 95% confidence, and below 0.0122 with a 99% confidence. This shows that our resulting graphs have converged sufficiently to closely represent the actual average values.

The results of Figure 12, Figure 13, and Figure 14 with a varying T are mostly the same as those seen for the small network. The relative performance of the different methods appears to be the same, with the exception that the maximin values of the propagation and degree are closer together. This is also the case for the results of Figure 15, Figure 16, and Figure 17 with a varying k. We also do not see the early convergence of the propagation heuristic to the optimal solution, which indicates that the number of edges with at least one node in M must be greater than 15000.

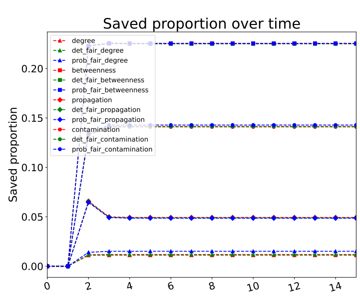

Figure 12: Number of nodes saved in large network with $k = 7500$

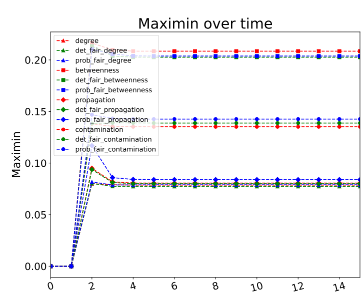

Figure 13: Maximin value in large network with $k = 7500$

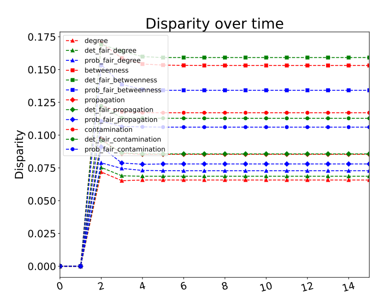

Figure 14: Disparity in large network with $k = 7500$

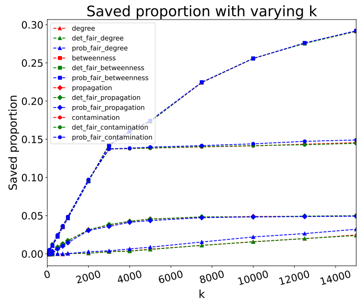

Figure 15: Number of nodes saved in large network with $T = 15$

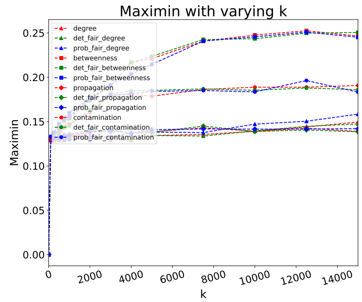

Figure 16: Maximin value in large network with $T = 15$

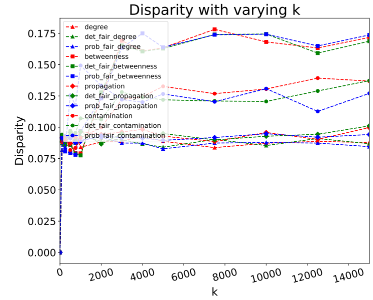

Figure 17: Disparity in large network with $T = 15$

5.3.3 Small network with four methods

Based on the previous results, we see that the different edge selection methods barely affect the performance or fairness seen in the simulations, whereas the heuristic makes a big difference. The fairness-agnostic edge selection model is the computationally least expensive of the three edge selection models. Therefore, in further experiments, we will compare the edge weight heuristics using only the fairness-agnostic edge selection model to minimize computing time and generate more intuitive figures. The following figures show the results of experiments run with only four methods on the small network, averaged over 50000 repetitions.

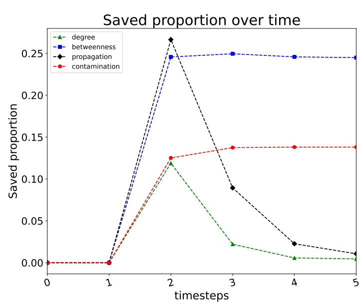

Figure 18: Number of nodes saved in small network with $k = 100$

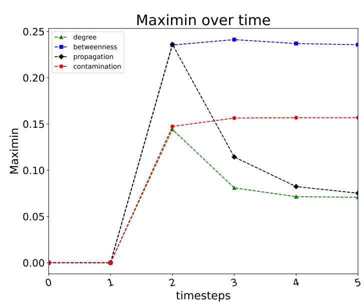

Figure 19: Maximin value in small network with $k = 100$

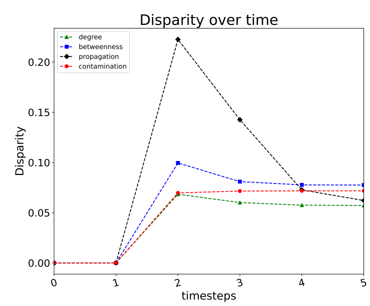

Figure 20: Disparity in small network with $k = 100$

5.4 Real world networks

We also want to test our methods on real world networks. For this purpose we use the Facebook network of McAuley and Leskovec [85] and the Astrophysics collaboration network of Leskovec et al. [86]. The Facebook network consists of 4039 nodes and 88234 edges, and the Astrophysics collaboration network consists of 17903 nodes and 197031 edges. Further details can be found in Table 3. These networks are found on the Stanford Network Analysis Platform [87]. To use them in our experiments, we convert each of these networks from an edgelist to the .graphml-format.

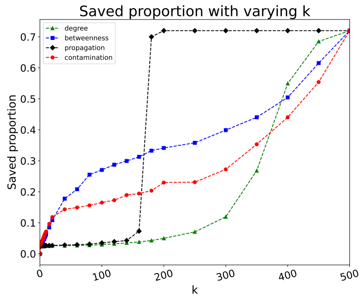

Figure 21: Number of nodes saved in small network with $T = 5$

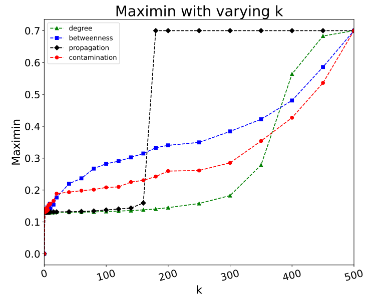

Figure 22: Maximin value in small network with $T = 5$

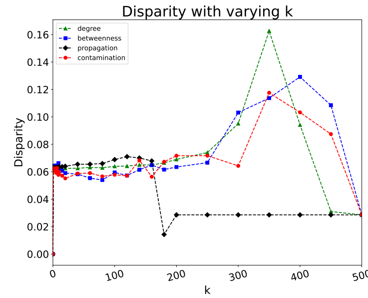

Figure 23: Disparity in small network with $T = 5$

We test our methods on two real world networks: a Facebook network and an astrophysics collaboration network. The network properties are given in Table 3.

Network	Facebook	Astrophysics
Nodes	4 039	17 903
Edges	88 234	197 031
Diameter	8	14
Avg Shortest Path	3.69251	4.19401
Avg Degree	43.69101	22.01095
Avg Clustering Coefficient	0.60555	0.63282
Nr Communities Louvain	16	40
Avg WEI	-0.60934	0.17774
Assortativity Coefficient	0.05784	0.23102

Table 3: Network properties for real world networks

The results of experiments using the Facebook network are presented and discussed in Section 5.4.1. Subsequently, the results of experiments using the Astrophysics collaboration network are presented and discussed in Section 5.4.2.

5.4.1 Facebook network

For the Facebook network, we use the default values $k = 5000$ and $T = 10$. For the experiments with a changing value of T we average over 50000 repetitions. The maximum errors with 95% confidence are 0.0003, 0.00004, and 0.0027 for Figure 24, Figure 25, and Figure 26 respectively. Similarly, the corresponding maximum errors with 99% confidence are 0.0004, 0.00005, and 0.0035. For the experiments with a changing value of k we average over 1000 repetitions. The maximum errors with 95% confidence are 0.0033, 0.0023, and 0.0207 for Figure 27, Figure 28, and Figure 29 respectively. Similarly, the corresponding maximum errors with 99% confidence are 0.0044, 0.0030, and 0.0273.

The results shown in these figures are consistent with the results of the synthetic networks. We see that some methods rise and fall at the start for a changing number of timesteps, and the order of best-performing methods is the same as well. In the figures with a changing value of k, we see that the propagation method still rises to the optimum sooner than all other methods, and here too the order of methods is the same.

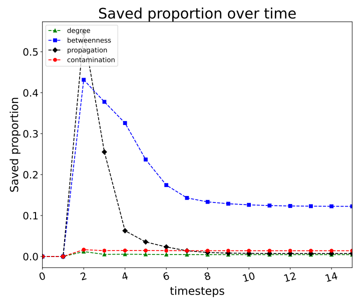

Figure 24: Number of nodes saved in Facebook network with $k = 5000$

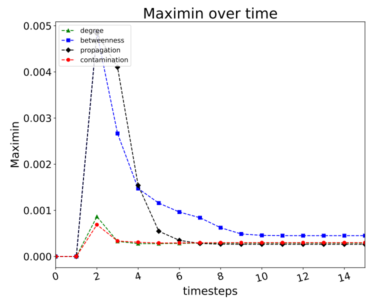

Figure 25: Maximin value in Facebook network with $k = 5000$

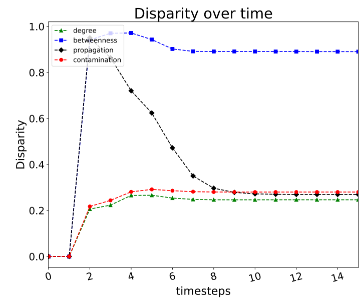

Figure 26: Disparity in Facebook network with $k = 5000$

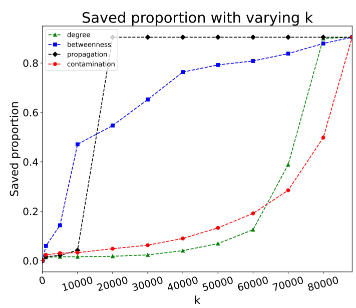

Figure 27: Number of nodes saved in Facebook network with $T = 10$

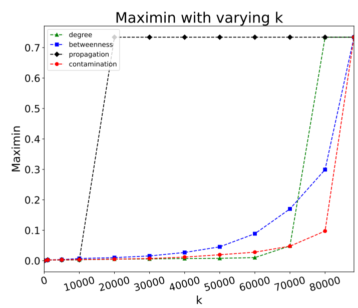

Figure 28: Maximin value in Facebook network with $T = 10$

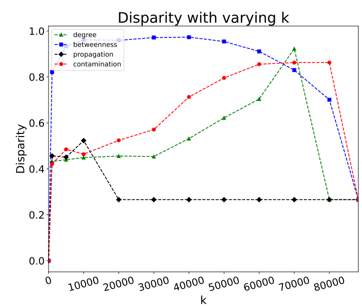

Figure 29: Disparity in Facebook network with $T = 10$

5.4.2 Astrophysics collaboration network

For the Astrophysics network, we use the default values $k = 10000$ and $T = 8$. For the experiments with a changing value of T we average over 50000 repetitions. The maximum errors with 95% confidence are 0.00004, 0.0003, and 0.0025 for Figure 30, Figure 31, and Figure 32 respectively. Similarly, the corresponding maximum errors with 99% confidence are 0.00005, 0.0004, and 0.0033. For the experiments with a changing value of k we average over 1000 repetitions. The maximum errors with 95% confidence are 0.0009, 0.0029, and 0.0127 for Figure 33, Figure 34, and Figure 35 respectively. Similarly, the corresponding maximum errors with 99% confidence are 0.0012, 0.0038, and 0.0167.

The results shown in these figures are consistent with the previous results. What is interesting in these figures, is that the converged maximin value in Figure 34 is fairly low, and the corresponding disparity in Figure 35 is high. Since saved community proportion is computed as the number of saved nodes in a community divided by its community size, a low maximin value can be explained by the presence of a community with few activated nodes in the baseline simulation. Similarly, a high disparity is the result of a community in which few nodes are activated in the baseline simulation and another community in which many nodes are activated in the baseline simulation. In a big network like the Astrophysics network with 40 communities, this is most likely the case.

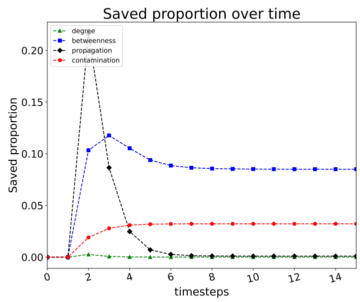

Figure 30: Number of nodes saved in Astrophysics network with $k = 10000$

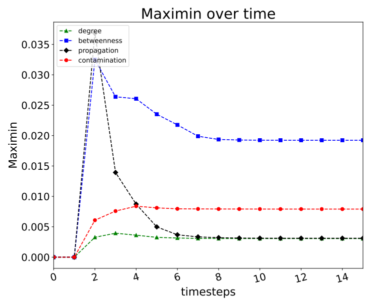

Figure 31: Maximin value in Astrophysics network with $k = 10000$

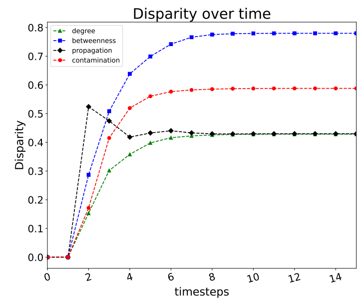

Figure 32: Disparity in Astrophysics network with $k = 10000$

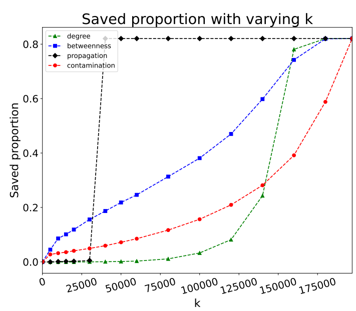

Figure 33: Number of nodes saved in Astrophysics network with $T = 8$

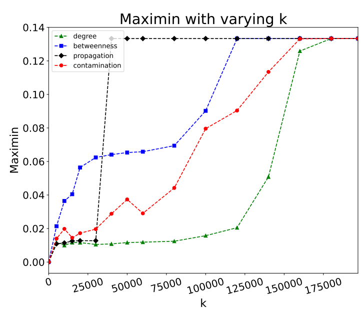

Figure 34: Maximin value in Astrophysics network with $T = 8$

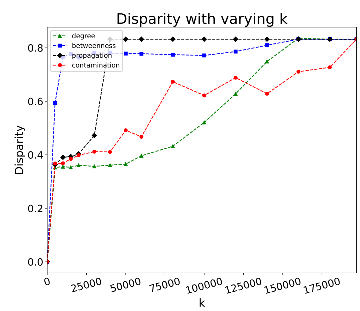

Figure 35: Disparity in Astrophysics network with $T = 8$

6 Conclusions and future work

In Section 6.1, we give our conclusions based on the results of our experiments. We present the best methods for influence blocking performance and fairness, and the findings regarding the experimental setup. This is followed in Section 6.2 by proposed future work relating to our research.

6.1 Conclusions

The objective of this thesis is to minimize the spread of influence by removing a set of edges. The resulting protection of nodes from this influence should be fair, meaning that the number of protected nodes for each community should be proportional to the community size. Since this is the first research into fairness-aware edge blocking for influence minimization, we also compare different methods for the experimental setup.

During the comparison of diffusion models, we have found that edge removal for influence minimization combined with propagation models such as the Linear Threshold model can cause a larger spread of influence. This happens when removing edges causes those edges that are important for spreading influence to become stronger and more likely to spread the influence. With such propagation models, the weight of an edge is inversely correlated to the number of edges around it. In this thesis, we use the Independent Cascade model for which this is not the case. However, this finding shows that it is important to distinguish between edge blocking methods and edge removal methods for influence minimization.

Our objective is to find a fairness-aware edge blocking method. To determine the fairness of a given method, we need fairness metrics. The metrics we use for this are the disparity and maximin metrics. We find that, when specifying these evaluation metrics regarding community fairness, it should be taken into consideration that some communities are simply too isolated to be affected. If such communities always cause a maximin value of zero because there are no infected nodes to save, this does not properly show the effect of the methods regarding community fairness.

Regarding our main set of experiments, we see that the heuristic used to assign edge weights plays a more important role than the method used to select edges based on these weights. Based on the fact that a higher proportion of nodes may be saved at the start of the diffusion than at a later timestep, we can also say that some nodes are not saved completely from influence, but it will simply take more timesteps for them to become activated. This is most obvious for the black line representing the propagation method, which looks only at the edges connecting the initially active node subset to the rest of the network. Given the influence blocking performance metric, we can rank the proposed influence blocking methods for a given number of timesteps and a given blocked edge set size. This ranking tells us that, for a larger number of timesteps, betweenness is the best heuristic to calculate the edge weights when focussing on the general performance. This method is followed by contamination, propagation, and finally degree. This ranking is reversed when the objective is to maximize group fairness based on the disparity metric, which implies that there is a trade-off to be made between fairness and performance. The rise of the propagation method to its optimum before the other heuristics, as seen in each of the figures where the number of removed edges is varied, shows that isolating the initially active node subset would be best, but as long as connections to the rest of the network remain its performance will remain relatively low. Finally, we see that as the number of removed edges increases, the performance also increases. This confirms our expectation of a monotonic relationship between the number of removed edges and the number of saved nodes.

6.2 Future work

In this thesis, we have compared a number of methods to find a fairness-aware approach for selecting a blocked edge set that minimizes the spread of influence. Based on experimental results, we propose six avenues for future work.

First, given the performance of the propagation degree heuristic and the betweenness heuristic, a combined heuristic could be developed to maximize the influence blocking performance. In this combined heuristic, we would use the betweenness centrality, but limit the set of blocked edges to only the out-edge set as is done for the propagation degree heuristic. This guarantees an optimal solution if the entire out-edge set can be selected. However, in cases where not the entire out-edge can be blocked, examples can be given for which it would be better to select an edge that is not part of the out-edge set. One example would be a network in which all initially active nodes are connected to the same node. This first inactive node is connected to the rest of the network by one edge. In this situation, removing the edge connecting the first inactive node to the rest of the network would disconnect the initial misinformation nodes from all other nodes, with the exception of the inactive node that they are all directly connected to. This removed edge would be the edge with the highest betweenness centrality, while not being part of the out-edge set.

Second, a different method for selecting the initially active node subset could be used. In our experiments, we select this set of initially active nodes as the set of nodes with the highest degree. It would be interesting to see the effect of using a different selection method on the performance of heuristics such as the degree heuristic. An example would be random selection.

Third, using different diffusion models than the Independent Cascade model may give an interesting result. Since diffusion models such as the Linear Threshold model or the Weighted Cascade model use the node degree to compute the propagation probability, a distinction has to be made between edge blocking and edge removal. For edge blocking, the propagation probability of a blocked edge could be set to zero, instead of removing the edge entirely.

Fourth, the objective could be changed to protect a target set. The research problem in this thesis is to find a blocked edge set to minimize the spread of influence through the entire network. However, the problem of influence minimization by blocking a set of edges has not yet been expanded to include a target set to protect from misinformation.

Fifth, a greedy edge blocking approach could be introduced for the fairness-aware influence minimization problem. In this thesis, we use edge weights calculated using heuristics due to limited computing power, but given enough time and resources, a greedy approach could be created. Such a greedy approach would have to deterministically calculate the expected spread of influence for each edge that may be blocked, similar to the expected lift in profit from earlier works.

Finally, an interesting approach would be to consider combined importance of edges. The heuristics used by our methods only look at the individual importance of edges. However, cases can be imagined in which removing a group of a few edges would completely disconnect (and thereby save) a large subset of nodes from the subset containing the initially active nodes. It would be interesting to see a method in which the combined effect of blocking or removing a set of edges is removed.

References

[1] Christopher L Barrett, Richard J Beckman, Maleq Khan, VS Anil Kumar, Madhav V Marathe, Paula E Stretz, Tridib Dutta, and Bryan Lewis. Generation and analysis of large synthetic social contact networks. In *Proceedings of the 2009 Winter Simulation Conference (WSC)*, pages 1003–1014. IEEE, 2009.

[2] Marco Conti, Andrea Passarella, and Fabio Pezzoni. A model for the generation of social network graphs. In *2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–6. IEEE, 2011.

[3] Robert De Caux, Christopher Smith, Dominic Kniveton, Richard Black, and Andrew Philippides. Dynamic, small-world social network generation through local agent interactions. *Complexity*, 19(6):44–53, 2014.

[4] Alan Mislove, Hema Swetha Koppula, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Growth of the flickr social network. In *Proceedings of the first Workshop on Online Social Networks*, pages 25–30, 2008.

[5] Chengxi Zang, Peng Cui, and Christos Faloutsos. Beyond sigmoids: The nettide model for social network growth, and its applications. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2015–2024, 2016.

[6] Akrati Saxena. Evolving models for dynamic weighted complex networks. *Principles of Social Networking: The New Horizon and Emerging Challenges*, pages 177–208, 2022.

[7] Smriti Bhagat, Graham Cormode, Balachander Krishnamurthy, and Divesh Srivastava. Class-based graph anonymization for social network data. *Proceedings of the VLDB Endowment*, 2(1):766–777, 2009.

[8] Sean Chester, Bruce M Kapron, Gautam Srivastava, and Srinivasan Venkatesh. Complexity of social network anonymization. *Social Network Analysis and Mining*, 3(2):151–166, 2013.

[9] Bin Zhou, Jian Pei, and WoShun Luk. A brief survey on anonymization techniques for privacy preserving publishing of social network data. *ACM SIGKDD Explorations Newsletter*, 10(2):12–22, 2008.

[10] Eytan Bakshy, Itamar Rosenn, Cameron Marlow, and Lada Adamic. The role of social networks in information diffusion. In *Proceedings of the 21st International Conference on World Wide Web*, pages 519–528, 2012.

[11] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A Zighed. Information diffusion in online social networks: A survey. *ACM SIGMOD Record*, 42(2):17–28, 2013.

[12] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *Automata, Languages and Programming: 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005. Proceedings 32*, pages 1127–1138. Springer, 2005.

[13] Dunia López-Pintado. Diffusion in complex social networks. *Games and Economic Behavior*, 62(2):573–590, 2008.

[14] Nicholas A Christakis and James H Fowler. Social network visualization in epidemiology. *Norsk Epidemiologi (The Norwegian Journal of Epidemiology)*, 19(1):5, 2009.

[15] James Moody, Daniel McFarland, and Skye Bender-deMoll. Dynamic network visualization. *American journal of sociology*, 110(4):1206–1241, 2005.

[16] Fernanda B Viégas and Judith Donath. Social network visualization: Can we go beyond the graph. In *Workshop on Social Networks, CSCW*, volume 4, pages 6–10, 2004.

[17] Bo-Lun Chen, Wen-Xin Jiang, Yi-Xin Chen, Ling Chen, Rui-Jie Wang, Shuai Han, Jian-Hong Lin, and Yi-Cheng Zhang. Influence blocking maximization on networks: Models, methods and applications. *Physics Reports*, 976:1–54, 2022.

[18] Akrati Saxena, Wynne Hsu, Mong Li Lee, Hai Leong Chieu, Lynette Ng, and Loo Nin Teow. Mitigating misinformation in online social network with top-k debunkers and evolving user opinions. In *Companion Proceedings of the Web Conference 2020*, pages 363–370, 2020.

[19] Dung V Pham, Giang L Nguyen, Tu N Nguyen, Canh V Pham, and Anh V Nguyen. Multi-topic misinformation blocking with budget constraint on online social networks. *IEEE Access*, 8:78879–78889, 2020.

[20] Biao Wang, Ge Chen, Luoyi Fu, Li Song, and Xinbing Wang. Drimux: Dynamic rumor influence minimization with user experience in social networks. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2168–2181, 2017.

[21] Akrati Saxena, Harsh Saxena, and Ralucca Gera. k-truthscore: Fake news mitigation in the presence of strong user bias. In *Computational Data and Social Networks: 9th International Conference, CSoNet 2020, Dallas, TX, USA, December 11–13, 2020, Proceedings 9*, pages 113–126. Springer, 2020.

[22] Akrati Saxena, George Fletcher, and Mykola Pechenizkiy. Fairsna: Algorithmic fairness in social network analysis. *ACM Computing Surveys*, 56(8):1–45, 2024.

[23] Cristina Gutiérrez Bierbooms. Fairness-aware influence blocking maximization for combating fake news. Master's thesis, Eindhoven University of Technology, Eindhoven, September 2022.

[24] Akrati Saxena, Pratishtha Saxena, and Harita Reddy. Fake news propagation and mitigation techniques: a survey. *Principles of Social Networking: The New Horizon and Emerging Challenges*, pages 355–386, 2022.

[25] Leonhard Euler. Solutio problematis ad geometriam situs pertinentis. *Commentarii Academiae Scientiarum Imperialis Petropolitanae*, 8:128–140, 1736.

[26] 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*, KDD '03, pages 137–146, New York, NY, USA, 2003. Association for Computing Machinery.

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

[28] Thomas C Schelling. *Micromotives and Macrobehavior*. W.W. Norton & Company, New York, NY, 1978.

[29] Richard Durrett. *Lecture notes on particle systems and percolation*. Brooks/Cole Publishing Company, 1988.

[30] Thomas Milton Liggett and Thomas M Liggett. *Interacting particle systems*, volume 2. Springer, 1985.

[31] Jacob Goldenberg, Barak Libai, and Eitan Muller. Talk of the network: A complex systems look at the underlying process of word-of-mouth. *Marketing Letters*, 12:211–223, 2001.

[32] Jacob Goldenberg, Barak Libai, and Eitan 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*, 9(3):1–18, 2001.

[33] Tao Hong and Qipeng Liu. Seeds selection for spreading in a weighted cascade model. *Physica A: Statistical Mechanics and its Applications*, 526:120943, 2019.

[34] Masahiro Kimura and Kazumi Saito. Tractable models for information diffusion in social networks. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Knowledge Discovery in Databases: PKDD 2006: 10th European Conference on Principles and Practice of Knowledge Discovery in Databases Berlin, Germany, September 18-22, 2006 Proceedings 10*, pages 259–271, Berlin, Heidelberg, 2006. Springer.

[35] Akrati Saxena, SRS Iyengar, and Yayati Gupta. Understanding spreading patterns on social networks based on network topology. In *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, pages 1616–1617, 2015.

[36] Yayati Gupta, Akrati Saxena, Debarati Das, and SRS Iyengar. Modeling memetics using edge diversity. In *Complex Networks VII*, pages 187–198. Springer, 2016.

[37] Rezvan Mohamadi-Baghmolaei, Niloofar Mozafari, and Ali Hamzeh. Trust based latency aware influence maximization in social networks. *Engineering Applications of Artificial Intelligence*, 41:195–206, 2015.

[38] Hui Li, Sourav S Bhowmick, Aixin Sun, and Jiangtao Cui. Conformity-aware influence maximization in online social networks. *The VLDB Journal*, 24:117–141, 2015.

[39] Tian Zhu, Bai Wang, Bin Wu, and Chuanxi Zhu. Maximizing the spread of influence ranking in social networks. In *Information Sciences*, volume 278, pages 535–544. Springer, 2014.

[40] Guangmo Tong, Weili Wu, Shaojie Tang, and Ding-Zhu Du. Adaptive influence maximization in dynamic social networks. *IEEE/ACM Transactions on Networking*, 25(1):112–125, 2016.

[41] Chris J Kuhlman, Gaurav Tuli, Samarth Swarup, Madhav V Marathe, and SS Ravi. Blocking simple and complex contagion by edge removal. In *2013 IEEE 13th International Conference on Data Mining*, pages 399–408. IEEE, 2013.

[42] Paul A Dreyer Jr and Fred S Roberts. Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion. *Discrete Applied Mathematics*, 157(7):1615–1627, 2009.

[43] Pedro Domingos and Matt Richardson. Mining the network value of customers. In *Proceedings of the seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '01, pages 57–66, New York, NY, USA, 2001. Association for Computing Machinery.

[44] 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, 2002.

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

[46] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *Internet and Network Economics: Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007. Proceedings 3*, pages 306–311. Springer, 2007.

[47] Jan Kostka, Yvonne Anne Oswald, and Roger Wattenhofer. Word of mouth: Rumor dissemination in social networks. In Alexander A. hvartsman and Pascal Felber, editors, *Structural Information and Communication Complexity: 15th International Colloquium, SIROCCO 2008 Villars-sur-Ollon, Switzerland, June 17-20, 2008 Proceedings 15*, pages 185–196, Berlin, Heidelberg, 2008. Springer.

[48] Akrati Saxena, Pratishtha Saxena, and Harita Reddy. Fake news detection techniques for social media. *Principles of Social Networking: The New Horizon and Emerging Challenges*, pages 325–354, 2022.

[49] Paritosh Kapadia, Akrati Saxena, Bhaskarjyoti Das, Yulong Pei, and Mykola Pechenizkiy. Co-attention based multi-contextual fake news detection. In *Complex Networks XIII: Proceedings of the 13th Conference on Complex Networks, CompleNet 2022*, pages 83–95. Springer, 2023.

[50] Akrati Saxena and Sudarshan Iyengar. Centrality measures in complex networks: A survey. *arXiv preprint arXiv:2011.07190*, 2020.

[51] Akrati Saxena, Ralucca Gera, and SRS Iyengar. Estimating degree rank in complex networks. *Social Network Analysis and Mining*, 8(1):42, 2018.

[52] Akrati Saxena, Ralucca Gera, and SRS Iyengar. A heuristic approach to estimate nodes' closeness rank using the properties of real world networks. *Social Network Analysis and Mining*, 9(1):3, 2019.

[53] Akrati Saxena and SRS Iyengar. K-shell rank analysis using local information. In *International Conference on Computational Social Networks*, pages 198–210. Springer, 2018.

[54] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. *Computer networks and ISDN systems*, 30(1-7):107–117, 1998.

[55] James Aspnes, Kevin Chang, and Aleksandr Yampolskiy. Inoculation strategies for victims of viruses and the sum-of-squares partition problem. *Journal of Computer and System Sciences*, 72(6):1077–1093, 2006.

[56] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. Graph structure in the web. *Computer networks*, 33(1-6):309–320, 2000.

[57] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Error and attack tolerance of complex networks. *Nature*, 406(6794):378–382, 2000.

[58] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[59] Yintao Yu, Tanya Y Berger-Wolf, and Jared Saia. Finding spread blockers in dynamic networks. In Lee Giles, Marc Smith, John Yen, and Haizheng Zhang, editors, *Advances in Social Network Mining and Analysis: Second International Workshop, SNAKDD 2008, Las Vegas, NV, USA, August 24-27, 2008*, pages 55–76, Berlin, Heidelberg, 2010. Springer.

[60] Chris J Kuhlman, VS Anil Kumar, Madhav V Marathe, SS Ravi, and Daniel J Rosenkrantz. Finding critical nodes for inhibiting diffusion of complex contagions in social networks. In José Luis Balcázar, Francesco Bonchi, Aristides Gionis, and Michèle Sebag, editors, *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part II 21*, pages 111–127, Berlin, Heidelberg, 2010. Springer.

[61] Huiling Zhang, Md Abdul Alim, My T Thai, and Hien T Nguyen. Monitor placement to timely detect misinformation in online social networks. In *2015 IEEE International Conference on Communications (ICC)*, pages 1152–1157. IEEE, 2015.

[62] Marco Amoruso, Daniele Anello, Vincenzo Auletta, Raffaele Cerulli, Diodato Ferraioli, and Andrea Raiconi. Contrasting the spread of misinformation in online social networks. *Journal of Artificial Intelligence Research*, 69:847–879, 2020.

[63] Ara Hayrapetyan, David Kempe, Martin Pál, and Zoya Svitkina. Unbalanced graph cuts. In *Algorithms–ESA 2005: 13th Annual European Symposium, Palma de Mallorca, Spain, October 3-6, 2005. Proceedings 13*, pages 191–202. Springer, 2005.

[64] Huiling Zhang, Md Abdul Alim, Xiang Li, My T Thai, and Hien T Nguyen. Misinformation in online social networks: Detect them all with a limited budget. *ACM Transactions on Information Systems (TOIS)*, 34(3):1–24, 2016.

[65] Mark EJ Newman, Stephanie Forrest, and Justin Balthrop. Email networks and the spread of computer viruses. *Physical Review E*, 66(3):035101, 2002.

[66] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 665–674, New York, NY, USA, 2011. Association for Computing Machinery.

[67] Nam P Nguyen, Guanhua Yan, My T Thai, and Stephan Eidenbenz. Containment of misinformation spread in online social networks. In *Proceedings of the 4th Annual ACM Web Science Conference*, pages 213–222, 2012.

[68] Songsong Li, Yuqing Zhu, Deying Li, Donghyun Kim, and Hejiao Huang. Rumor restriction in online social networks. In *2013 IEEE 32nd International Performance Computing and Communications Conference (IPCCC)*, pages 1–10. IEEE, 2013.

[69] Lidan Fan, Zaixin Lu, Weili Wu, Bhavani Thuraisingham, Huan Ma, and Yuanjun Bi. Least cost rumor blocking in social networks. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 540–549. IEEE, 2013.

[70] Huiyuan Zhang, Huiling Zhang, Xiang Li, and My T Thai. Limiting the spread of misinformation while effectively raising awareness in social networks. In My T. Thai, Nam P. Nguyen, and Huawei Shen, editors, *Computational Social Networks: 4th International Conference, CSoNet 2015, Beijing, China, August 4-6, 2015, Proceedings 4*, volume 9197, pages 35–47, Cham, 2015. Springer.

[71] Adil Imad Eddine Hosni, Kan Li, Cangfeng Ding, and Sadique Ahmed. Least cost rumor influence minimization in multiplex social networks. In Long Cheng, Andrew Chi Sing Leung, and Seiichi Ozawa, editors, *Neural Information Processing: 25th International Conference, ICONIP 2018, Siem Reap, Cambodia, December 13–16, 2018, Proceedings, Part VI 25*, volume 11306 of *Lecture Notes in Computer Science (LNTCS)*, pages 93–105, Cham, 2018. Springer.

[72] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Minimizing the spread of contamination by blocking links in a network. In *AAAI*, volume 8, pages 1175–1180, 2008.

[73] Zhang Hai-Feng, Li Ke-Zan, Fu Xin-Chu, and Wang Bing-Hong. An efficient control strategy of epidemic spreading on scale-free networks. *Chinese Physics Letters*, 26(6):068901, 2009.

[74] Qipeng Yao, Chuan Zhou, Linbo Xiang, Yanan Cao, and Li Guo. Minimizing the negative influence by blocking links in social networks. In Lu Yueming, Wu Xu, and Zhang Xi, editors, *Trustworthy Computing and Services: International Conference, ISCTCS 2014, Beijing, China, November 28-29, 2014, Revised Selected papers*, pages 65–73, Berlin, Heidelberg, 2015. Springer.

[75] Junaid Ali, Mahmoudreza Babaei, Abhijnan Chakraborty, Baharan Mirzasoleiman, Krishna P Gummadi, and Adish Singla. On the fairness of time-critical influence maximization in social networks. *IEEE Transactions on Knowledge and Data Engineering*, 35(3):2875–2886, 2023.

[76] Ana-Andreea Stoica and Augustin Chaintreau. Fairness in social influence maximization. In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 569–574, 2019.

[77] Ana-Andreea Stoica, Jessy Xinyi Han, and Augustin Chaintreau. Seeding network influence in biased networks and the benefits of diversity. In *Proceedings of the Web Conference 2020*, pages 2089–2098, 2020.

[78] Golnoosh Farnad, Behrouz Babaki, and Michel Gendreau. A unifying framework for fairness-aware influence maximization. In *Companion Proceedings of the Web Conference 2020*, pages 714–722, 2020.

[79] Akrati Saxena, Cristina Gutiérrez Bierbooms, and Mykola Pechenizkiy. Fairness-aware fake news mitigation using counter information propagation. *Applied Intelligence*, 53(22):27483–27504, 2023.

[80] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[81] Fariba Karimi, Mathieu Génois, Claudia Wagner, Philipp Singer, and Markus Strohmaier. Homophily influences ranking of minorities in social networks. *Scientific Reports*, 8(1):11077, 2018.

[82] Eun Lee, Fariba Karimi, Claudia Wagner, Hang-Hyun Jo, Markus Strohmaier, and Mirta Galesic. Homophily and minority-group size explain perception biases in social networks. *Nature Human Behaviour*, 3(10):1078–1087, 2019.

[83] Xindi Wang, Onur Varol, and Tina Eliassi-Rad. Information access equality on generative models of complex networks. *Applied Network Science*, 7(1):1–20, 2022.

[84] James Samuel Coleman et al. Introduction to mathematical sociology. *Introduction to Mathematical Sociology.*, 1964.

[85] Jure Leskovec and Julian Mcauley. Learning to discover social circles in ego networks. *Advances in Neural Information Processing Systems*, 25, 2012.

[86] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *ACM transactions on Knowledge Discovery from Data (TKDD)*, 1(1):2–es, 2007.

[87] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford large network dataset collection. `http://snap.stanford.edu/data`, June 2014.