

MASTER

Fairness-aware Influence Blocking Maximization for combating Fake News

Gutiérrez Bierbooms, Cristina

Award date: 2022

Link to publication

Disclaimer

This document contains a student thesis (bachelor's or master's), as authored by a student at Eindhoven University of Technology. Student theses are made available in the TU/e repository upon obtaining the required degree. The grade received is not published on the document as presented in the repository. The required complexity or quality of research of student theses may vary by program, and the required minimum study period may vary in duration.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
You may not further distribute the material or use it for any profit-making activity or commercial gain



Department of Mathematics and Computer Science Data Mining Research Group

Fairness-aware Influence Blocking Maximization for combating Fake News

Master's Thesis

Cristina Gutiérrez Bierbooms

Supervisors: prof.dr. Mykola Pechenizkiy dr. Akrati Saxena

Assessment Commitee: prof.dr. Mykola Pechenizkiy dr. Akrati Saxena

dr. Felix Mannhardt

Eindhoven, August 2022

Abstract

Fake news mitigation has been the main research direction for the network science and data mining communities in the past few years. Several methods to solve this problem have been proposed, however, none of these considers the fairness of the approach towards each community. These approaches rely on heuristics that favour certain communities, which are more likely to occupy these beneficial positions they select. This results in the exclusion of some other communities from the benefits of the intervention, which can have important societal repercussions.

In this project, we will focus on developing a fairness-aware truth campaigning technique for blocking the influence propagation of a competing entity, in this case with the use case of mitigating the spread of fake news. Truth campaigning consists in identifying a set of users in a social network to begin the spreading of the fact. The spread of this real information will then counteract the spreading of fake news. Educating users about the real fact will then avert them from believing in the fake news. We then propose a solution that not only maximizes the number of users being protected from the spread of this misinformation but a solution that also maximizes the group- fairness objective to avoid disregard towards certain groups.

This proposed method is evaluated against existing benchmarks, extracted from previous work in the area of information diffusion and some additionally proposed fairness benchmarks, in both real and synthetic data sets. With this evaluation the unfairness of the existing methods and the need for a fairness-aware alternative are presented. The proposed method aims then to fill this void in research as it achieves higher fairness, but we also show that such fairness does not come at a cost in the number of saved nodes. In fact, in most cases it works as a catalyst for achieving better effectiveness, performing better than the state-of-the-art methods. Such work shall motivate the research of fairness-aware methods for solving the Influence Blocking maximization (IBM) problem.

Keywords—Fake news, Fairness, WRR

Preface

First, I would like to thank Akrati Saxena, my daily supervisor, for all her interaction and support throughout the project. She allowed for my total independence in this work, which empowered me to explore my crazy ideas. She did this while remaining active and involved in the process. This perfect balance allowed us to have very enriching discussions, to further improve my work into achieving the good results of this final thesis.

I also want to thank Mykola Pechenizkiy for his supervision in this project, his comments and discussions not only on the technical content of the project but also on the social and psychological one, were very enriching and helped me improve the report.

And lastly, my family, who deserves endless gratitude. The two greatest gifts you could have given me were education and freedom of choice, and for that, I owe you a big part of my success in life. Thank you for making me who I am today. I want to express my gratitude for your unconditional support throughout my time at university, especially in light of the challenges posed by the big distance between us. There are no words to adequately explain how grateful I am to you all. You've given me the motivation I need to keep trying to be my best self every day and you have genuinely been the best source of support I could have imagined. Throughout this arduous journey, I sincerely appreciated your love, support, and care.

Contents

Li	st of	Figures	\mathbf{v}							
\mathbf{Li}	st of	Tables	\mathbf{vi}							
N	otati	on	x							
1	Intr 1.1 1.2 1.3 1.4	coduction Context	1 1 3 4 5							
2	Pre 2.1 2.2	eliminaries Independent cascade model								
3	Lite 3.1 3.2	erature Review Influence Blocking Maximization Fair Influence Blocking Maximization	9 9 12							
4	Pro 4.1 4.2 4.3	oposed methods FWRRS method Parity seeding for FIBM problem Fairness-aware CMIA-O								
5	Exp 5.1 5.2	Periments Real datasets experiments 5.1.1 Datasets 5.1.2 Experiment settings 5.1.3 Experiments Evaluation 5.1.3.1 Evaluation under the WC model with random negative seed 5.1.3.2 Evaluation under the WC model with highest degree negative seed 5.1.3.3 Evaluation under alternative cascade models 5.1.3.4 Evaluation under different community detection methods Synthetic datasets experiments	30 30 33 34 34 34 37 41 42 44 44 47							
6	Con 6.1 6.2	Main contributions	53 53 54							
Bi	bliog	graphy	57							

Appendix			
\mathbf{A}	Uniform model evaluation	63	
В	Synthetic network generator model	65	

List of Figures

Figure 1.	Findings visualization of the Ohio State University research, values were extracted from [49]	2
Figure 2.	Example of independent cascade model propagation	7
Figure 4. Figure 4. Figure 4. Figure 4.	 Example of a WRR (right) of a graph (left) Flow chart of the FWRRS method Differentiated thresholds for strategic seeding, extracted from [72] Differentiated thresholds for strategic seeding 	$13 \\ 15 \\ 25 \\ 27$
Figure 5.	Degree distribution of the datasets	32
Figure 5.2	Experimental analysis in the Facebook dataset under the WC model with Random S_N	34
Figure 5.	Experimental analysis in the Twitter dataset under the WC model with	01
Figure 5.	Random S_N Experimental analysis in the NetPHY dataset under the WC model with	35
Timuna F	Random S_N	35
Figure 5.	Experimental analysis in the Nether 1 dataset under the WC model with Random S_N	35
Figure 5.	Experimental analysis in the NetASTRO dataset under the WC model with	36
Figure 5.	' Experimental analysis in the Enron-Email dataset under the WC model with	50
Figure 5	Random S_N	36
i iguie o.	degree S_N	38
Figure 5.	Experimental analysis in the Twitter dataset under WC model with the highest degree S_N	38
Figure 5.	0 Experimental analysis in the NetPHY dataset under WC model with the highest	20
Figure 5.	degree S_N 1 Experimental analysis in the NetHEPT dataset under WC model with the highest	38
Figure 5	degree S_N	39
rigure 5.	the highest degree S_N	39
Figure 5.	.3 Experimental analysis in the Enron-Email dataset under the WC model with the highest degree $S_{\rm W}$	39
Figure 5.	4 Experimental analysis in the NetPHY dataset under the Uniform model	41
Figure 5.	5 Percentage of saved nodes of the methods under different community detec- tion algorithms in NetPHY under the WC model, with k=100	43
Figure 5.	6 Maximin value of the methods under different community detection algorithms	
Figure 5.	IN NetPHY under the WC model, with k=100	44 46
Figure 5. Figure 5.	.8 Degree distribution of the synthetic datasets $\dots \dots \dots \dots \dots \dots \dots \dots \dots \dots \dots$ 9 Experimental analysis in the SG1 dataset under WC model with random $S_N \dots$	48 49

Figure	5.20	Experimental	analysis i	n the	SG2	dataset	under	WC	model	with	random	S_N .		•	50
Figure	5.21	Experimental	analysis i	n the	SG3	dataset	under	WC	model	with	random	S_N .			50
Figure	5.22	Experimental	analysis i	n the	SG4	dataset	under	WC	model	with	random	S_N .			51
Figure	5.23	Experimental	analysis i	n the	SG5	dataset	under	WC	model	with	random	S_N .			52
Figure	A.1	Experimenta	l analysis	$\sin t$	the I	Faceboo	k data	aset	under	the	Uniform	n mo	del		
		(p=0.1) with	Random	nega	ative	seed S_1	N • •								63

List of Tables

Table 1	Notation Table	х
Table 3.1	Literature review on truth campaigning table	11
Table 5.1	Datasets characteristics	33
Table 5.2 Table 5.3	Synthetic datasets characteristics	$43 \\ 47$

Notation

All utilized notation in this thesis, is presented in the following table.

Abbreviations

IBM	Influence Blocking Maximization
FIBM	Fairness-aware Influence Blocking Maximization
IM	Influence Maximization
ICM	Independent Cascade Model
OSN	Online Social Networks
WRR	Weighted Reverse Reachable
FWRRS	Fairness-aware Weighted Reverse Reachable system
HICH-BA	HIgh Clustering Homophily barabási-Albert
WC	Weighted Cascade model
EVC	Eigenvector centrality
CLTM	Competitive linear threshold model
MCICM	Multi-Campaign Independent Cascade Model
COICM	Campaign Oblivious Independent Cascade Model
MIIA	Maximum influence In-Arborescence
MIOA	Maximum influence Out-Arborescence
DAG	Directed Acyclic Graph
MC	Monte Carlo

Variables

SN	set of nodes	in G	that	will star	spreading	the	fake	news
\mathcal{O}_N	set of nodes	шG	unau	will Star	spreading	une	lake	news

 S_P set of debunker nodes in G that will start spreading the counter true information

$\mathbb{B}(S,G,S_N,S_P)$	blocking power of a set of nodes S , consists of the number of additional nodes S_P saves from being infected by S_N if S is added to S_P .
$s^-(G, S_N, S_P)$	number of nodes in G that will get negatively infected by S_N if S_P spreads the counter information
$\mathbb{M}(G, S_N, S_P)$	maximin value achieved in the communities of G given S_N and S_P are spreading competing information.
WRRS	Weighted Reverse Reachable System, consists of a pool of WRR trees.
$p_{u \rightarrow v}$	probability of node v believing the information being shared by node \boldsymbol{u}
k	budget for the truth campaign consists of the maximum number of nodes the set S_P can have
$comm_v$	community to which v belongs to
d_v	out-degree of node v
d_in_v	in degree of node v
$Neigh_out_v$	out neighbours of node v
$Neigh_in_v$	in neighbours of node v
C	set of non-overlapping communities that constitute the population of ${\cal G}$
p_v	defined within a WRR, is the probability of node v reaching the root of the tree, with the information it is sharing
$probs_v$	approximated probability of node v getting infected by S_N if no intervention is made
$paths_v$	set of paths throughout node v could be reached by S_N
C_inf	communities that get infected in the $WRRS$
$prob_inf_v$	according to the $WRRS$, the probability of node v getting infected by S_N , given S_P
$init_inf_c$	number of infections expected in community \boldsymbol{c} if no intervention is made
exp_inf_c	number of infections currently expected in community c with the constructed ${\cal S}_P$
$B_v[c]$	approximated blocking power of $S_P \cup \{v\}$ in community c .
pct_saved_c	expected percentage of avoided infections in community \boldsymbol{c}
$maximin_v$	expected maximin value if node v were to join S_P
$total_sav_v$	approximated blocking power of node v according to the $WRRS$
$sav_no_help_v$	blocking power of node v within those communities which are currently being neglected by the constructed S_P , according to the WRRS

$wgt_sav_below_avg_v$	weighted blocking power of node v within those communities which currently have a pct_saved_c value below average, the weight of a node w is set to $(1-pct_saved_{comm_w})$ according to the $WRRS$
$saved_min_v$	blocking power of node v within those communities which are currently establishing the maximin value , according to the $WRRS$
S	set of nodes expected to get infected within the $FWRRS$ method
γ	minimum probability of infection a node must have to be part of ${\cal S}$
ϵ	approximation guarantee of the probabilities in the WRRs re- turned by D-SSA, and thus, the allowed error along the FWRRS method
$descts_v$	descendant nodes of node v within a WRR
$original_v$	original node in G that a node v in a WRR , refers to
s_v	distance from a node v to the root in a WRR
t_{max}	upper-bound on number of iterations performed by $D\text{-}SSA$
$\hat{S}_{m k}$	optimal solution for the $Max - Coverage_W RRS$ problem
Λ_1	minimum coverage of WRRs that both S_N and \hat{S}_k should achieve
$\hat{\mathbb{I}}_t(\hat{S_k})$	expected number of nodes in S that \hat{S}_k reaches with its information
$community_S_N[c]$	proportion of S_N nodes that belong to community c
$take_c$	number of nodes the Stoica method will select from community \boldsymbol{c}
$DecInf_c(v)$	number of infections in community c if node v was to join S_P
$ExpInf_{c}$	number of currently expected infections in community c with the constructed S_P .
$ap^N(u, S_N, S_P, MIIA)$	probability of infection of node u within the $MIIA$, given S_N and S_P .
$ap^N(u,t)$	probability node u being negatively infected by time step t within a $MIIA$.
$ap^P(u,t)$	probability node u being infected with the information S_P is spreading, by time step t within a <i>MIIA</i> .
$p^N(u,t)$	probability node u being negatively infected precisely at time step t within a $MIIA$.
$p^P(u,t)$	probability node u being infected with the information S_P is spreading, precisely at time step t within a <i>MIIA</i> .
heta	influence threshold used in the CMIA-O and $\mathrm{Maximin}_CMIA-Oalgorithms$
NegS	set of nodes that have a path to a node in S_N that reaches them with probability at least θ .
$MIIA(u, \theta)$	MIIA rooted at node u containing all paths to such node, that have a probability of at least θ .

 $MIOA(u, \theta) \qquad \text{MIIA rooted at node } u \text{ containing all paths from such node, that} \\ \text{have a probability of at least } \theta.$

Table 1: Notation Table

Chapter 1

Introduction

This first chapter will introduce the problem covered in this project. A introduction of the context of the problem is presented in Section 1.1, followed by the problem description, presented in Section 1.2. The contribution of the research performed for this thesis is presented in Section 1.3 and last, in Section 1.4 an outline of the structure of the remaining thesis is given.

1.1 Context

Long gone are the days of people getting their news information from the morning news or Sunday's newspaper. Consumers continue to shift away from these traditional media sources and are moving towards social media for their news consumption. This source of information has attracted over 2.4 billion internet users [1] by easing access to not only textual content but also multimedia and external websites with a potentially broad audience. In a perfect world, everything reported would be based entirely on facts such that one could trust that the media consumed is reliable. However, this is not the case. Misinformation and rumours are becoming highly prevalent and such ease of access and fast dissemination of information on social media has unfortunately been exploited for the spread of fake news.

An example of the impact of fake news in society is the US elections of 2016. Mark Zuckerberg [2] admitted that 126 million Americans were exposed to Russian-backed, politically-oriented fake news via Facebook during the presidential election campaign of 2016. In the aftermath of these elections as the extent of fake news could be observed, three scientists from Ohio State University [49] -Gunther, Beck, and Nisbet- explored whether people might have changed their vote due to fake news. They interviewed a group of people who voted for Obama in 2012 and looked at their votes in the 2016 election. The voters were asked how much they believed in three statements that, according to independent analysis, had been promoted by fake news while being false. These were: that Hilary was in poor health due to serious illness, that Pope Francis endorsed Trump and that during Hilary's time as Secretary of State, she approved weapon sales to Islamic Jihadists like ISIS. Although the majority of people did not believe these assertions to be true, there was a very substantial correlation between their beliefs and voting behaviour. The findings of this study were summarized in Figure 1.1, which was extracted from the work of Gunther, Beck, and Nisbet.



Figure 1.1: Findings visualization of the Ohio State University research, values were extracted from [49]

Among those voters who did not believe in any of the three statements, 89% did cast their vote for Hilary in 2016; those who believed in just one of the items 61% voted for her, but from those who believed in at least 2 of these fakes news only 17% supported Clinton. These results must be taken with consideration, as this correlation does not imply the causality of Hilary's defeat. However, this shows how big the impact of fake news was on the election, and how misinformed the voters who changed their vote were.

Fake news detection mechanisms have been extensively proposed in past years [67]. These methods have been proven to achieve very high accuracy in their predictions. For example, the work of Antoun [7], who proposed a method with a precision of 98%. Our goal now is to determine how to proceed once the fake news has been identified, namely focusing on the mitigation of such misinformation.

Many researchers have studied the propagation of fake news over social media, proposing various techniques to combat them, most are collected in the work of Saxena [68]. However, the proposed methods do not consider fairness. In networks, different groups or communities are unequally represented, and typically, minority groups are disproportionately absent from the advantageous positions that these approaches cover, creating a diversity gap. This could result in the exclusion of these communities from the benefits of the intervention, which can have important societal repercussions. This has motivated the research on fairness-aware fake news mitigation for this paper.

Given a set of nodes identified to be the source of the fake news propagation in the online social network, as well as the network in which the information will be spread, we focus on the Influence Blocking Maximization (IBM) problem. This consists of finding a set of users from whom counter information will begin to propagate to block the negative spread as much as possible, this technique is known as truth campaigning.

It has long been pointed out that actively broadcasting the true information is much more effective than merely immunizing some nodes or edges [70], also known as user blocking. Immunizing these users by requesting them not to spread the fake news when encountered might have feasibility problems, as the effectiveness of the method will rely on the willingness of the selected users to follow our recommendation. The truth campaigning approach, on the other hand, might offer higher feasibility than influence blocking, as users are more likely to accept and follow the recommendation as a result of the provided education about the real fact. Additionally, the goal of this work is to overcome the unfairness current methods have towards certain communities. Since allowing the impact of the fake news to be prominent within certain communities can have significant societal repercussions, it must then be the goal of the campaign to avoid this scenario by enforcing the fairness of the intervention. Thus the problem that is being solved will be a variation of the IBM, where the second objective in terms of fairness among the communities will be included. The main research question of this study is then:

How do we achieve the trade-off between fairness and effectiveness for IBM problem?

To answer this question the following sub-questions will have to be studied

- What is an appropriate evaluation of fairness in the IBM problem?
- How fair are existing methods?
- How can the fairness of the truth campaign be guaranteed? For this, a novel fairness-aware solution for IBM will be proposed.
- Does fairness come at a cost? If so, how could we overcome such a cost?

1.2 Problem Description

The problem being covered is the Influence Blocking Maximization (IBM) one, which is an optimization problem. It has as input a graph, G = (V, E), which may be directed or not, the negative seed $S_N \subseteq V$ which are those nodes spreading the fake news and a positive integer krepresenting the available budget for the mitigation campaign. We then aim to find the positive seed set $S_P \subseteq V \setminus S_N$ of size at most k, such that the expected number of negatively activated nodes is minimized.

Equivalently we want to choose S_P such that the number of nodes that are saved is maximized, where saved nodes are those that would have been "infected" by the misinformation if it was not for the truth campaign. We define then the blocking power of a node u, $\mathbb{B}(u, G, S_N, S_P)$, to be the number of nodes that would additionally be saved if this node was to be included in the positive seed S_P . The blocking power of a node u in a network G, given a negative seed S_N and a positive seed S_P , is computed as follows:

$$\mathbb{B}(\{u\}, G, S_N, S_P) = s^-(G, S_N, S_P) - s^-(G, S_N, S_P \cup \{u\})$$

where $s^{-}(G, S_N, S_P)$ represents the number of nodes that get infected in G if S_N starts spreading misinformation and S_P spreading the truth. Thus, the goal is to choose those nodes with the highest blocking power. In summary, the IBM problem is defined as:

$$\begin{array}{ll} \underset{S_P}{\operatorname{maximize}} & \mathbb{B}(S_P, G, S_N, \emptyset) & (1.1a) \\ \text{subject to} & \\ & |S_P| <= k, \ S_P \subseteq V \setminus S_N & (1.1b) \end{array}$$

But in this case, we will be solving a fairness-aware alternative of the problem, which we define to be the **Fairness-aware Influence Blocking Maximization** (FIBM) problem. The goal of the problem is not only to maximize the blocking power of the positive seed but also to ensure that the outcome of the truth campaign is fair.

In this work, the fairness of the campaign will be measured in terms of maximin, a choice that will be introduced and motivated in Section 2.2. Maximin aims to avoid the scenario in which some communities are disproportionately neglected from help, compared to the remaining population. Thus, trying to maximize the minimum ratio of saved nodes that all communities have. Let $\mathbb{M}(G, S_N, S_P)$ define the maximin value achieved by S_P in the graph G which contains communities C, that is required to mitigate the information being spread by S_N . The maximin value is computed as follows:

$$\mathbb{M}(G, S_N, S_P) = \min_{c \in C} \; \frac{s_c^-(G, S_N, \emptyset) - s_c^-(G, S_N, S_P)}{s_c^-(G, S_N, \emptyset)}$$

The optimization definition for this version of the problem is defined as follows:

$$\begin{array}{l} \underset{S_P}{\operatorname{maximize}} \quad \mathbb{B}(S_P, G, S_N, \emptyset) \tag{1.2a} \end{array}$$

subject to

$$S_P \in \underset{S'_P}{\operatorname{arg\,max}} \mathbb{M}(G, S_N, S'_P), \tag{1.2b}$$

$$|S'_P| \le k, \ S'_P \subseteq V \setminus S_N \tag{1.2c}$$

In this version of the problem an additional constraint is included, namely 1.2b, this will guarantee the fairness of the approach. The solution set S_P shall then not only guarantee that the truth camping is as effective as possible, by saving as many nodes as possible from believing the fake news. But it shall also guarantee that the maximin value has also been maximized. Namely by reducing the set of possible solutions to those that meet condition 1.2b, those that achieve the maximum fairness value, and from that solution space choosing the set S_P , that achieves such desired maximum effectiveness.

1.3 Contribution

To solve the presented problem, this thesis proposes an algorithmic solution. The Fairness-aware Weighted Reversible reachable system (FWRRS) method, relies on the structure of weighted reversible reachable trees (WRR) to choose those nodes that will save, from the fake news, the largest number of nodes while maximizing the least help that every community receives. Later this method will be evaluated against numerous existing methods for solving the IBM problem. Such will allow for evaluating not only the fairness of existing methods but also the cost of the fairness of the proposed method. To make such evaluation more complete, as no existing benchmark considers fairness, we propose two extensions to existing methods, that aim at achieving a fairer outcome.

The evaluation of the methods is performed in both real and synthetic data. The real data consists of real social networks which are publicly available and have been extensively utilized in information diffusion research. Additionally, they cover numerous types of social interactions. To be able to control the characteristics of the networks and observe their impact on the performance of the methods, a novel model to generate synthetic networks is presented. The HIgh Clustering Homophily Barabási-Albert (HICH-BA) model generates networks that resemble the characteristics of real-life social networks better than other existing models [23]. It guarantees a higher clustering coefficient and a better-defined community structure than any other synthetic data model. The methods are then evaluated in the constructed synthetic datasets.

To sum up, the contribution of this thesis is the following. A novel problem in the area of information diffusion is introduced, a fairness-oriented version of the IBM problem. For this new problem, the FWRRS method is proposed, which is evaluated with in-depth testing against numerous benchmarks. In this evaluation, we show the need for fairness-aware methods to guarantee fair interventions in society, as in most scenarios all benchmarks are extremely unfair. This method additionally is shown to perform as the state-of-the-art for the IBM problem under the Independent cascade model, utilized to simulate the spread of the information. Thus not only is the need for fairness presented but its value beyond morality is also shown, as it offers higher effectiveness.

The implementation of the proposed methods as well as all mentioned benchmarks, network generator and propagation models can be found in the following repository: https://github.com/ cristinagub/master_project.git

1.4 Outline

The remaining of this document is organised as follows. In Chapter 2, the background knowledge upon which this research is built is reported. This is followed by related work from literature in Chapter 3. Next, in Chapter 4, the proposed methods are introduced. In this chapter, the method FWRRS is explained, as well as some additional fairness-aware benchmarks to make the later evaluation more complete. This evaluation is presented in Chapter 5, which is organized into two subsections corresponding to the real datasets and synthetic datasets experiments. Last, the conclusions, as well as limitations and future research, are then summarized in Chapter 6.

Chapter 2 Preliminaries

In this section, we will introduce all additional theories upon which this work was built. The notions presented in this chapter will be referenced in the remaining of the report. The chapter is structured as follows, in Section 2.1 the notion of propagation models and the utilized independent cascade model are introduced and in Section 2.2 the notion of Fairness in the area of information diffusion, both crucial for achieving fair influence blocking maximization.

2.1 Independent cascade model

Understanding fake news propagation, is a crucial first step, to make an educated choice on how to mitigate them, as well as to evaluate the performance of the chosen methods. It is then necessary to model their spread within an online social network (OSN) such as Facebook or Twitter with a propagation model. Diverse propagation models have been proposed to simulate the spread of information, most of these were collected by Saxena et al. on the following survey [68].

The most commonly used model to simulate the spread of misinformation is the **Independent** cascade model, presented by Herrmann [31]. The main idea behind the Independent Cascade Model(ICM)[36] is that after a node is infected, meaning that it believes in the fake news, it tries to infect all its neighbouring nodes in the next time step. These neighbours believe the information with a certain probability, which has been allocated to the edge connecting them to the infected node. The infected node will only try to infect its neighbours once, thus if in an iteration no new nodes are infected, the spread is ceased, like in Figure 2.1. The author compares the diffusion simulated by the model to various diffusion datasets extracted from Twitter and experimentally shows how the proposed method and the real diffusion behave similarly.

This type of model has been extended for two competitive information propagates by Budak et al. [14]. Budak proposed the Multi-Campaign Independent Cascade Model(MCICM) that models the diffusion of two cascades evolving simultaneously in a network, each of which has its propagation probabilities, which in this case differ. The Campaign-Oblivious independent cascade model (COICM) is a type of MCICM, proposed by Wu [75] which assumes that both competitive information spread simultaneously using the same cascade, thus sharing propagation probability values. For the COICM in the case of a node being reached by both campaigns at the same step, the user will be biased towards one of them. It is common to set all nodes to have a bias towards the true information.

To be able to evaluate existing methods in terms of fairness, it was decided to work with the most commonly used propagation model, the Independent Cascade Model (ICM). In this paper, the IBM problem will be studied under the Campaign-Oblivious Independent Cascade Model (COICM).



Figure 2.1: Example of independent cascade model propagation

2.2 Fainess in information diffusion

The notion of fairness is notoriously difficult to define. To try and define fairness we will refer to two different ethical frameworks. On the one hand is the **consequentialist ethical framework**, which defines good actions as those that have good consequences, suggesting that fairness might be viewed as equal impact. Under this category are the following definitions of fairness:

- Equality: focuses on the fair allocation of resources to the groups proportional to the size of the group within the population. Measuring fairness on the allocations. This would be achieved for the IBM problem by guaranteeing the population of the positive seed S_P resembles the one of the network population. However, such fairness might be difficult to achieve, and may lead to wastage of resources [59].
- Equity: focuses on fair treatment among the groups proportional to their size in the population. Measuring fairness in the outcome, like in the work of Stoica [72] for the IM problem. Under this definition, for the IBM problem, it shall be guaranteed that the population of the saved nodes resembles the real population distribution.
- Maximin: is closely related to equity, as it also measures fairness in the outcome. This definition is based on the Rawlsian theory [60]. The difference is that maximin wants to maximize the minimum relative influence received by any group. The objective is then to maximize the percentage of nodes we save from the community that was saved the least.
- Diversity constraints: the key idea is that a community should not be better off on its own. The campaign should guarantee a better outcome for all communities than the result of assigning each community its proportional share of the resources and allocating them internally. This constraint was introduced by [59] for the IM problem, but can also be extended for the IBM one.

On the other hand, the **deontological ethical framework**, defines good actions as those that accord with good norms, suggesting that fairness might be viewed as equal treatment. This view suggests that fairness can only be achieved by algorithms that are agnostic to community affiliation or sensitive attributes when maximizing blocked influence. However as shown by Stoica [71],

when using feature-blind mechanisms, the inequality based on sensitive features that exist in the network propagates through the algorithmic design and affects the outcome.

In this work, we will focus on the definition of fairness as equal impact within the communities, as we strive to overcome the inequalities of the network and avoid its effect on the truth campaign's outcome. Further, since fairness should not come at a large cost, the maximin definition will be used. According to Farnad's [28], the maximin approach not only has the highest coverage for the most disadvantaged group in the IM problem, but it also has the lowest price of fairness. Since the IBM problem has a goal to take advantage of the positive information coverage to block the negative spread, we observe the close relationship between the IM and IBM problem. Thus, we expect that maximin will help us overcome the cost of fairness in this problem too, since other definitions that come at a cost in the IM problem will cause this cost also in the IBM problem.

In this project we will then measure fairness in terms of maximin, choosing S_P to be the positive seed that maximizes the minimum proportion of nodes saved from each community. This relates to the legal notion of disparate impact, which states that a community has been unfairly treated if its "success rate" under a policy is substantially lower than that of the other communities [9]. Maximizing fairness, as defined by maximin, is therefore important to governmental or community organizations which are under pressure to prevent this kind of disparity.

Chapter 3 Literature Review

In this chapter, related work to this project will be presented. This intends to cover the vast majority of the publications in the area of our research. The chapter is structured as follows. In Section 3.1 the previous work in the area of Influence Blocking Maximization is presented, and in Section 3.2 fairness-aware solutions in the area of information diffusion are discussed.

3.1 Influence Blocking Maximization

The IBM problem is NP-hard and its objective function has been proven under most propagation models to be monotone sub-modular. In the work of Lv et al. [43], this was proven for the COCIM, which will be used in this thesis. This implies that there exists a **greedy algorithm** that offers a solution with an approximation ratio of $(1 - 1/e - \epsilon)$ to the optimal solution, where ϵ depends on the accuracy of the influence range of each node, estimated using Monte Carlo (MC) simulations. However, this greedy method is very slow as a consequence of the numerous and time-consuming MC simulations that are required. This makes its use too expensive, especially for large networks like those encountered in social media.

More scalable approaches for solving the IBM problem have been proposed in earlier work. These can be grouped into two categories: heuristic methods and simulation-based methods.

On the one hand are the **heuristic approaches**. These rely on the characteristics of each node to make the selection of the k most suitable ones for the truth campaign. Under this category, we find Yao's et al. work [79], which proposed to choose the seed of truth campaigners based on *eigenvector centrality (EVC)* [82]. A high EVC value for a node means it is connected to many nodes who themselves also have high EVC values. They show that under the Competitive Linear Threshold Model (CLTM), it performs better than the state-of-the-art algorithm for such propagation model, the CLDAG while also being faster. We also find other heuristics such as degree, clustering coefficient [26], betweenness centrality [13] or percolation centrality [56]. The performance of these metrics and some others were evaluated by Erd et al. [25] under two cost functions. One of these considered a degree cost penalty making nodes with many connections more expensive, and the other one considered equal cost across the network's nodes. It was shown that under the Multi-Campaign Independent Cascade Model (MCICM), degree, betweenness and percolation offered the best performance. However, they required expensive nodes for their success, thus under the degree penalty cost function, these methods did not perform as well.

Under the heuristic category, methods exploiting the structure of the network have also been proposed. These rely on the community structure of the graph and will be considered as **community-based heuristic methods**. These proposed methods include the work of Lv et al. [43] who al-

located resources to each community proportional to the proportion of negative seed nodes within that community and their power of infection. Similarly, the work of Arazkhani et.al [8] proposes to select the nodes with the largest degree, betweenness and closeness centrality measures from the largest k communities as truth campaigners. The community-based heuristic methods aim to allocate resources to communities based on their risk of infection, this could lead to fairer outcomes as resources are fairly divided, however, the approach might be unfair toward indirectly infected communities.

Regarding simulation-based algorithms we find the state-of-the-art method under the CLTM, the CLDAG algorithm [30]. The algorithm is characterized by restricting the influence computation of a node v to its local area to reduce the computation cost; and by carefully selecting a local graph structure for v to allow efficient and accurate influence computation for v under this structure. Such a method utilizes the properties of the CLTM propagation model, to solve the problem. Along this line we also find the work of Wu et al. [75], who proposes two very similar methods, CMIA-H and CMIA-O which instead exploit the characteristics of the MCICM with high-effectiveness property (assuming the probability of positive transmission is 1 always) and Campaign-oblivious independent cascade model (COICM) respectively. Their research shows that these methods outperform the considered heuristic approaches, such as degree or proximity while remaining fast. Despite the good efficiency of the CMIA-O, Lin et al. [42] shows that its performance is far from optimal. The method relies on the construction of numerous subgraphs and the consideration of a vast amount of nodes as candidates for the positive seed, which is time-consuming. To solve this, they propose the BIOG approach, which claims to be significantly faster than Wu's one while performing similarly. Last, under this category, we find the TIB-Solver proposed by Song et al. [70]. In this approach, they first find the set of nodes that might be infected by the rumour and then calculate the threat of each of these nodes (threat of v is the expected number of nodes that can be infected by v). They then rely on weighted Reverse Reachable (WRR) trees to greedily select k nodes that save the most nodes in a given deadline.

All previously mentioned work aims at solving the version of the IBM problem that we will be considering. However, many researchers have extended this problem to consider more complex scenarios. For example, the work of Nguyen et al. [48], who proposes to find the least cost set S_P , that can protect at least a predefined percentage of nodes, where the spread of fake news was detected with some delay T. Under this category of IBM extensions, is the work of Hosni et al. [32], who proposed to reduce the area in which the impact is measured to a predefined sub-sample of the population. Here we also find the work of Song et al. [70], as they also evaluated their method in an extended version of the IBM problem where a deadline for the campaign is defined. [64, 66] proposed methods to identify truth-campaigners when the users have strong biases towards competitive information propagation. Further is the work of Budak et al. [14], who also includes a delay in detecting the rumour spread before taking action.

In Table 3.1, we summarize works in the area of the IBM problem that have been mentioned in this section. This can be found in the next table. In this table, the following is included. For each reference, we first mention the type of method that we proposed in this work and the method's name. Next, under *complexity*, we mention the proven complexity of the problem being solved. We include which extensions of the problem were considered in each work. If a reference considered *budget*, means that they included a parameter k in the optimization setting the maximum cost a campaign could have. *Prospective debunkers* mean that the nodes for S_P could only be selected from a subset of nodes. *Target nodes*, on the contrary, mean that the goal is to maximize the number of saved nodes from a subset of nodes of the graph. The *decontamination ratio* shows if the authors aim at ensuring that θ % of the nodes do not get infected. *Delay* means that the fake news has already started spreading. *Deadline*, if the authors aim at achieving the maximum number of saved nodes, before a certain time step. Last, the *propagation model* that was utilized as well as the benchmarks considered in the evaluation are also mentioned.

Baseline heuristics	Random, CLDAG and Degree	Degree, Percolation Betweenness, Random Clustering and PageRank	Random, degree and proximity	Random, degree and proximity	Random, degree and proximity	Random, degree and proximity	CMIA-O, Random , degree and proximity	PageRank, LSMI , largest infectees	Degree, early and largest infectees	Degree, PageRank, random and DiscountIC	Degree and random
Propagation model	CLTM	MCICM	COICM	MCICM	CLTM	COICM and MCICM	COICM	COICM	MCICM	COICM and CLTM	COICM
Deadline								>			
Delay									>	`	
Decontami- nation ratio										<	
Target nodes											>
Prospective debunkers									>		
Budget	>	>	>	~	~	>	~	>	>		>
Complexity			NP-Hard					NP-Hard	NP-Hard		NP-Hard
Method	EVC centrality		CB_IBM	Centrality_IBM	CLDAG	CMIA-O and CMIA-H	BIOG	TIB-Solver	TIB-Solver	GVS	
Type	Heuristic	Heuristic	Community-based heuristic	Community-based heuristic	Simulation-based	Simulation-based	Simulation-based	Simulation-based	Simulation-based	Simulation-based	Simulation-based
References	Yao et al. [79]	Brd et al. [25]	Lv et al. [43]	Arazkhani et.al [8]	He et.al[30]	Wu et.al[75]	Lin et. al[42]	Song et.al[70]	Budak et.al[14]	Nguyen et.al[48]	Hosni et.al[32]

Table 3.1: Literature review on truth campaigning table

3.2 Fair Influence Blocking Maximization

To the best of our knowledge, fairness has not been considered for solving the problem of Influence Blocking Maximization. In fact in the extensive surveys on fairness in graph mining [22, 63], no existing work in this area was included. However, fairness has been considered in other related areas of information diffusion, such as in solving the Influence Maximization (IM) problem.

The IM problem was first formulated by Kempe et al. [35] and focuses on identifying a set of k initial adopters for maximizing the spread of certain information within a network. In the area of **fairness in IM** is the work of Ali et al. [5], who proposed to greedily choose the node that achieves the highest value for an objective function which balanced influence and fairness. In this work, fairness was defined as achieving an equal fraction of influenced nodes across the communities, also known as equity. Thus the objective function would punish nodes that would cause the distance between the most and least influenced communities to be large, showing that guaranteeing such fairness comes at the cost of a reduction in performance. Following this work, Stoica et al. [71, 72] aims to examine the conditions under which diversity would not come at a cost, but as a catalyst for maximizing outreach. This shows that when the seed set size is large enough, promoting parity in the seed results in better parity in the outreach. Additionally, this diverse seed set taps into inactivated communities that are hard to reach from central nodes, and thus leads to a better outreach, proving the benefits of fairness beyond a balance in the output. ElHalabi et al. [24] demonstrated empirically that fair solutions are often nearly optimal, meaning fairness does not come at a significant cost (less than 15%). Additionally, they proved that the algorithms that do not impose fairness constraints introduce significant bias.

Inspired by the necessity for fairness constraints in the area of IM for fair outcomes and their positive impact, we will bring fairness into the IBM problem. In this report, we will evaluate the fairness of multiple of the currently used IBM methods, and present a new approach that aims to offer a fairer impact while offering a close to the optimal outcome.

Chapter 4

Proposed methods

In this chapter, we cover the methods that were developed to solve the influence blocking maximization problem while remaining fair, in terms of maximin. The proposed method of this work namely the Fairness-aware Weighted Reversible reachable system (FWRRS) method is introduced in Section 4.1. Additionally, benchmarks utilized in the evaluation are presented in Sections 4.2 and 4.3, both consisting of adaptations of previous work in the area of Fairness in influence diffusion and IBM respectively, to solve the FIBM problem. These will work as fairness benchmarks in our evaluation in Chapter 5.

4.1 FWRRS method

The main possible drawback of the existing methods is that to some extent they simplify the network for the corresponding estimation of blocking power, which can affect the effectiveness in reality. The heuristic methods, in the extreme, reduce it to the characteristics of a node, and the simulation-based ones, to some extent, to their local area. For example, CMIA-O and BIOG, assume that information only spreads through maximum probability paths with a probability of at least θ , which is typically set to 0.01 and *TIB-Solver* reduces the network structure to a DAG to compute the blocking power of each node. To overcome such a drawback, the proposed method will not oversimplify the network. We proposed to exploit the structure of weighted reversible reachable (WRR) trees, an example of such structure is represented in Figure 4.1. In this figure, a toy example network is represented on the left and an example WRR of such network is represented on the right. Each node in the WRR has a parameter p, which denotes the probability of it reaching the root.



Figure 4.1: Example of a WRR (right) of a graph (left)

The construction process for a WRR can be found in Algorithm 6. For the WRR in Figure 4.1, this consisted of the following steps. First, a random node in the graph is chosen, in this case, this was N_5 , and is added to the WRR as the root. From this node, we iterate over the graph, by visiting nodes in a breadth-first search manner. After adding the root, we consider the neighbours of N_5 , namely N_4, N_2, N_3 and N_1 , which are added to the WRR with a probability equal to the probability allocated to the edge connecting them to the node being considered. In this case, this is 0.5 for all neighbours. Based on these probabilities only N_3 and N_1 were added, and their p-values were set to 0.5. Next, the neighbours of N_3 were considered similarly, resulting in the inclusion of N_2 whose p-value is now equal to the probability of the path, in the WRR, connecting it to the root. The same process is repeated also for N_1 where N_2 and N_3 were added to the WRR, however since they were already present in this graph, a copy of them, N'_2 and N'_3 respectively were added. This process is then repeated from every newly added node in the WRRR. This is then done from N_2 , where no new nodes were added. Following we considered N'_2 and from this node, N_4 which belongs to S_N was added. In this case, when a rumour spreader node s is added to a WRR no nodes with a distance from the root larger than the one from this node s are added. Thus, we consider the node N'_3 and add N_6 in this case, but the process terminates here. Additionally, we observe that all nodes in the WRR have a p-value larger or equal to the one of N_4 the rumour spreader node, so no further pruning is performed since those nodes with a lower probability would normally be removed.

This structure aims at considering for each node all those other nodes which can reach it with their information. Thus, if a path of diffusion is to be considered, a WRR is expected to cover it, avoiding the oversimplification problem of all other benchmarks. Such structure can be generated very fast, allowing the methods to generate a large pool of WRRs trees that resemble how information could diffuse in the network. The proposed method will also incorporate maximin as its objective to solve the IBM problem while being fairness-aware.

The proposed method is presented in Algorithm 1. This algorithm takes 6 parameters as input. The first one is G, which consists of the network in which the campaign will take place. This network contains also the division of the population into communities C. Further, are the negative set S_N , and the budget k. The parameter γ represents the probability threshold to add a node to S, which is the set of nodes we expect will get infected. Next, is parameter ϵ , which is one of the input parameters of the D-SSA algorithm, set to a value of 0.1. This parameter will be the approximation guarantee of the probabilities in the WRRs returned by D-SSA, and thus, the error that will be allowed throughout the proposed method. Last, is *iterations* which consists of the number of simulations that will be utilized to approximate the infection probabilities. Using these parameters a pool of WRRs is constructed, with which the blocking power of each node per community is approximated. Iteratively a node is selected and the blocking power approximations are updated until the size of S_P is equal to k. The selection of the node consists of the following 6 selection steps:

- 1. Select those nodes that maximize maximin, these are stored in a list called options_1
- 2. Remove from *options*_1 those nodes without blocking power in those communities that have not been saved yet unless these communities are connected to the current S_P and have a very small expected number of infections.
- 3. Further, select those that maximize the weighted blocking power in those communities with a saving ratio below average. The resulting subset of *options_1* is stored in *options_2*
- 4. Then those that maximize the blocking power in those communities with a minimum saving ratio are stored in a list called *options_3*
- 5. Those with a maximum blocking power are selected, if the previous sub-sampling 2-4 steps caused a cost for effectiveness.
- 6. Last, select the node with maximum weighted out-degree.

Following, the pseudocode of the algorithm in Algorithm 1 and a flow chart of the method in Figure 4.2 are presented. In this figure, input parameters are represented in yellow, variables in blue, decisions are in green and terminating states are in red.



Figure 4.2: Flow chart of the FWRRS method

Algorithm 1 FWRRS (G, S_N , k, γ , ϵ , iterations)

1: $S, probs, paths \leftarrow compute_S(G(V, E, C), S_N, \gamma, iterations)$ 2: $WRRS \leftarrow D$ -SSA_WRRS $(G, S, probs, paths, S_N, k, \epsilon)$ 3: $WRRS_{inf} \leftarrow WRR \in WRRs$ if $WRR.has_S_N = True$ 4: $init_inf \leftarrow \text{list of length } |C| \text{ consisting of } 0s$ 5: for $v \in S$ do $count_v \leftarrow \#WRRs$ in WRRS with root v6: $prob_inf_v \leftarrow \frac{count_inf_v}{\#WRRs \in WRRS_{inf} \text{ with root } v}$ 7: $init_inf[comm_v] + = prob_inf_v$ 8: 9: $C_inf \leftarrow [c \in C \text{ if } init_inf[c] > 0]$ 10: $pct_saved_c \leftarrow 0$ for $c \in C_inf$, $exp_inf \leftarrow init_inf$ 11: $B_v \leftarrow \text{list of length } |C|$ consisting of 0s, for all $v \in V$, candidates $\leftarrow \emptyset$, $S_P \leftarrow \emptyset$, maximin_value $\leftarrow 0$ 12: for $WRR \in WRRS_inf$ do $r \gets WRR.root$ 13:14:for $v \in WRR \setminus S_N$ do $B_v[comm_r] = +1/count_r$ and add v to candidates 15: while $|S_P| < k$ do $maximin_v \leftarrow \min_{c \in C_inf} \quad \frac{B_v[c]}{init_inf[c]} \text{ for all } v \in candidates$ Selection step 1 16:options_1 \leftarrow [$v \in candidates$ if $maximin_v \in (1 \pm \epsilon * \Phi_1) * \max_{w \in V} maximin_w$] 17: $total_sav_v \leftarrow \sum_{c \in C_inf} B_v[c]$ for $v \in options_1$, $list_total_sav \leftarrow [total_sav_v \text{ for } v \in options_1]$ 18:Selection step 2 $C_no_help \leftarrow [c \in C_inf \text{ if } pct_saved_c = 0]$ 19:20: if $|C_no_help| > 0$ then $sav_no_help_v \leftarrow \sum_{c \in C_no_help} B_v[c] \text{ for } v \in options_1$ 21:if $\frac{\sum\limits_{c \in C_no_help} exp_inf[c]}{\sum\limits_{c \in C_inf} exp_inf[c]} \ge \epsilon * \frac{|C_no_help|}{|C_inf|}$ or $\exists_{c \in C_no_help}$ disconnected from S_P then 22: $options_1 \leftarrow [v \in options_1 \text{ if } sav_no_help_v > 0]$ 23: $C_below_avg \leftarrow [c \in C_inf \text{ if } pct_saved_c \leq \sum_{c' \in C_inf} pct_saved_{c'} / |C_inf|]$ Selection step 3 24: $wgt_sav_below_avg_v \leftarrow \sum_{c \in C_below_avg} B_v[c] * (1 - pct_saved_c) \text{ for } v \in options_1$ 25:26:options_2 \leftarrow [$v \in options_1$ if $wgt_sav_below_avg_v \in (1 \pm \epsilon) * \max_{w \in options_1} wgt_sav_below_avg_w + \Phi_2$] Selection step 4 $C_min_help \leftarrow [c \in C_inf \text{ if } pct_saved_c \in (1 \pm \epsilon) * maximin_value]$ 27: $saved_min_v \leftarrow \sum_{c \in C_min_help} B_v[c] \text{ for } v \in options_2$ 28:options_3 \leftarrow [$v \in options_2$ if saved_min_v $\in (1 \pm \epsilon) * \max_{w \in options_2} saved_min_w + \Phi_3$] 29: $\max_{v \in options_3} total_sav_v \le \max list_total_sav * (1 - \epsilon) \text{ then}$ Selection step 5 30: remove nodes in options_3 if $total_sav_v \notin (1 \pm \epsilon) * \max_{v \in options_3} total_sav_v$ 31: Selection step 6 $\sum_{w \in Neigh_out_v \cup \{v\}}$ $prob_inf_w$ for $v \in options_3$ 32: $u \leftarrow \arg \min$ $v \in options_3$ 33: add u to S_P 34:remove u from candidates $exp_inf \leftarrow init_inf - B_u, maximin_value \leftarrow maximin_u, pct_saved_c \leftarrow \frac{B_u[c]}{init_inf[c]}$ 35: for $WRR \in WRRS_inf$ with $u \in WRR$ do 36: $r \leftarrow WRR.root$ 37: $prob_inf_r - = 1/count_r$ 38: for $v \in candidates \setminus WRR$ do $B_v[comm_r] + = 1/count_r$ 39: remove WRR from WRRS_inf 40: 41: return S_P

Algorithm 1 is explained below in more detail.

1. Identify the subset of nodes with expected infection (Line 1). This subset consists of those nodes that are reached by the information shared by a node in S_N with a probability of at least γ in the *iterations* simulations performed. In this work, the value of *iterations* was set to 1000, as larger values did not show significant performance improvement, and caused an increase in running time. These simulations are performed efficiently through the construction of WFR (Weighted Forward Reachable) trees. The subset of nodes S and the corresponding probabilities and paths of infection are obtained by calling *compute_S* on Gand S_N . The code for this computation is presented in the Algorithm 2.

The process of a call of compute_S begins by generating a graph $G_{extended}$. This consists of a copy of G containing an extra node source. This node is connected to all infectees with edges that have a probability of 1 (Lines 1:3). Utilizing $G_{extended}$, we generate 1000 WFR trees rooted at source (Lines 6:11). These WFR trees are generated according to the pseudo-code presented in Algorithm 3. Each WFR represents how the information, being spread by the S_N nodes, propagates and reaches other nodes in the network. For each node vwe keep track of the number of occurrences within the WFR trees. The proportion of WFR trees in which a node v is present is then an estimate of the probability of v being reached by S_N , defined as $probs_v$. We also keep track of the paths through which each node got reached by S_N , this list paths will then be used for pruning in the construction of the WRR trees in Algorithm 6. This variable will allow us to avoid the construction of unnecessarily large WRR trees. When all 1000 WFR trees are constructed, S is set to be those nodes with a probability of infection of at least γ , such selection will help speed up the process of generating WRRs in which nodes get infected. Then, probs, paths and S are returned.

Algorithm 2 compute S (G, S_N , γ , iterations)

1: $G_extended \leftarrow G$ 2: add node *source* to G_extended 3: add edges (source,s) for s in S_N to G_extended, where $prob_{source \rightarrow s} = 1$ 4: $S \leftarrow \emptyset$ 5: $probs_v \leftarrow 0$ for $v \in V$ 6: for i=1:*iterations* do 7: $S_i, inf_path \leftarrow generate_WFR(G_extended)$ for $v \in S_i$ do 8: $probs_v + = 1/1000$ 9: $paths_v$ add inf_path_v 10: add s to S11: 12: $S \leftarrow \text{list of nodes } s \in S \text{ with } probs_s >= \gamma$ 13: return S, probs, paths

Each of these WFR trees is generated by calling generate_ $WFR(G_extended)$, a method whose pseudocode is presented in Algorithm 3. The process of constructing a WFR tree begins by initializing the set S_i which will be containing the set of nodes that get infected by S_N in this WFR, to be an empty set (Line 1). Next, the node source is selected as the root (Line 2). Then $G_extended$ is iterated over by adding to S_i a neighbour v of the node u currently being considered with probability $p_{u\to v}$ (Lines 4:13). This is performed with the use of a list A which includes those nodes to be considered next. In each iteration a node u is extracted out of A and its neighbours are selected for being added to S_i with the corresponding probabilities (Lines 6:8). In the case of a node v being selected to be added to S_i , it is added to A also, for it to be considered in next iterations (Lines 9:13). Only nodes that are not in S_N are added to S_i since nodes in S_N can not get infected (Line 12). However, these nodes will be added to A to iterate through the graph. Additionally, nodes are only considered once, since, under the ICM, nodes can only be negatively activated in one time step. Thus, if a node v is to be added to S_i but it is already in this set, then it will not be added again into A. We also keep track of the nodes that needed to be negatively activated to reach each node v in the WFR (Line 11), this is stored in the set inf_path_v . Once no more nodes are present in A, the WFR construction is completed and we return S_i and inf_path . This process resembles taking a random sub-graph of $G_extended$, where an edge (u, v) is included with probability $p_{u \to v}$ and including all nodes reachable from source into S_i . Under this analogy, inf_path_v will then consist of the nodes encountered in the shortest path from source to node v.

Algorithm	3	$generate_WFR(G_extended$	(V, I)	E))
-----------	---	---------------------------	--------	----	---

1: $S_i \leftarrow \emptyset$ 2: $r \leftarrow source$ 3: $inf_path_r \leftarrow \emptyset$ 4: $A \leftarrow [r]$ 5: while $A \neq \emptyset$ do 6: $u \leftarrow \text{take first element out of } A$ 7: for $v \in V$ such that the edge $(u, v) \in E$ do add_v is True with probability $p_{u\to v}$ else False 8: if add_v and $v \notin S_i$ then 9: add v to A10: $inf_path_v \leftarrow inf_path_u \cup \{v\}$ 11:if $v \notin S_N$ then 12:13:add v to S_i 14: return S_i , inf_path

2. Generate a pool of WRR trees (Line 2). The pool of WRR trees utilized by the *FWRRS* method is constructed using a version of the D-SSA sampling algorithm, which is presented in Algorithm 4. This sampling algorithm was introduced by Nguyen [46], which was evaluated to be 1200 times faster than the next best sampling method. It offers a scalable solution for the class of hard optimization problems over samples/sketches with guarantees in the estimated solution.

Algorithm 4 D-SSA_WRRS (G, S, S_N, k, ϵ)

1: $n \leftarrow |S|, \quad \delta \leftarrow \frac{1}{n}$ 2: $N_{max} \leftarrow 8\frac{1-1/e}{2+2e/3} * \Upsilon(\epsilon, \frac{\delta}{6}/\binom{n}{k}) * \frac{n}{k}$ 3: $t_{max} \leftarrow \lceil log_2(2N_{max}/\Upsilon(\epsilon, \frac{\delta}{3})) \rceil$ 4: $\Lambda \leftarrow \Upsilon(\epsilon, \frac{\delta}{3*t_{max}})$ 5: $\Lambda_1 \leftarrow 1 + (1 + \epsilon) * \Upsilon(\epsilon, \frac{\delta}{3*t_{max}})$ 6: $t \leftarrow 0$ 7: $WRRS_t \leftarrow \{WRR_1, ..., WRR_{\Lambda 2^{t-1}}\}$ generated using generate_ $WRR(G, S, probs, paths, S_N)$ 8: while $|WRRS_t| < N_{max}$ do 9: $t \leftarrow t + 1$ 10: WRR-with_ $S_N \leftarrow$ number of $WRR \in WRRS_t$ with WRR-has_ $S_N = True$ 11: $WRRS_{t}^{c} \leftarrow \{WRR_{\Lambda 2^{t-1}+1}, ..., WRR_{\Lambda 2^{t}}\}$ generated using generate_WRR(G, S, probs, paths, S_N) $\hat{S}_k, \hat{\mathbb{I}}_t(\hat{S}_k) \leftarrow Max\text{-}Coverage(WRRS_t, k, S_N)$ 12: $roots_list_t^c \leftarrow list containing WRR.root for all WRR in WRRS_t^c$ 13:14: $count_t^c(r) \leftarrow number of occurrences of r in roots_list_t^c$ for all r for WRR in $WRRS_t^c$ do 15:if $\hat{S}_k \cap WRR \neq \emptyset$ then 16: $r \gets WRR.root$ 17: $\mathbb{I}_t^c(\hat{S}_k) + = 1/count_t^c(r)$ 18: $Cov_{-}WRRS_{t}^{c}(\hat{S}_{k}) \leftarrow \mathbb{I}_{t}^{c}(\hat{S}_{k})/|S| * |WRRS_{t}^{c}|$ 19:if $Cov_WRRS_t^c(\hat{S}_k) \geq \Lambda_1$ then 20: $\epsilon_1 \leftarrow \hat{\mathbb{I}}_t(\hat{S}_k) / \mathbb{I}_t^c(\hat{S}_k) - 1$ 21: $\epsilon_2 \leftarrow \epsilon \sqrt{\frac{n(1+\epsilon)}{2^{t-1}\mathbb{I}_t^c(\hat{S}_k)}}$ 22: $\epsilon_3 \leftarrow \epsilon \sqrt{\frac{n(1+\epsilon)(1-1/e-\epsilon)}{2^{t-1}\mathbb{I}^c_t(\hat{S}_k)}}$ 23: $\epsilon_t \leftarrow (\epsilon_1 + \epsilon_2 + \epsilon_1 \epsilon_2)(1 - 1/e - \epsilon) + (1 - 1/e)\epsilon_3$ 24:if $\epsilon_t \leq \epsilon$ then 25: $roots_list_t \leftarrow list containing WRR.root for all WRR in WRRS_t^c$ 26: $count_t(r) \leftarrow number of occurrences of r in roots_list_t for all r$ 27:for WRR in WRRS + do 28:if $S_N \cap WRR \neq \emptyset$ then 29: $r \leftarrow WRR.root$ 30: 31: $\mathbb{I}_t(S_N) + = 1/count_t(r)$ if $\mathbb{I}_t(S_N) \geq \Lambda_1$ then return $WRRS_t$ 32: $WRRS_t \leftarrow WRRS_t \cup WRRS_t^c$ 33: 34: return $WRRS_t$

Algorithm 5 Max-Coverage_WRRS $(WRRS_t, k, S_N)$

1: $\hat{S}_k \leftarrow \emptyset$ 2: $WRRS_to_cover \leftarrow WRRS_t$ 3: roots_list \leftarrow list containing WRR.root for all WRR_t in WRRS 4: $count_r \leftarrow$ number of occurrences of r in roots_list for all elements in the list 5: $coverage_v \leftarrow 0$ for all $v \in V$ for WRR in $WRRS_t$ do 6: for $v \in WRR$ do 7: $r \leftarrow WRR.root$ 8: $coverage_v + = 1/count_r$ 9: 10: $\hat{\mathbb{I}}_t(\hat{S}_k) \leftarrow 0$ 11: for i=1:k do $\hat{v} \leftarrow argmax_{v \in V \setminus S_N} \ coverage_v$ 12:add \hat{v} to \hat{S}_k 13: $\hat{\mathbb{I}}_t(\hat{S}_k) + = coverage_{\hat{v}}$ 14:for WRR in $WRRS_to_cover$ with $\hat{v} \in WRR$ do 15:for $v \in V \setminus WRR$ do 16: $coverage_v - = 1/count_r$ 17: remove WRR from WRRS_to_cover 18: 19: return $\hat{S}_k, \hat{\mathbb{I}}_t(\hat{S}_k)$

The general idea of the algorithm is to recursively double the amount of WRRs until certain

conditions are met. Then the resulting WRRS (Weighted reversible reachable system) is returned to be used by the *F*-WRRS method. In iteration *t*, the algorithm has generated $\Lambda \times 2^t$ WRRs and divides them into two sets. The first one $WRRS_t = \{WRR_1, ..., WRR_{\Lambda 2^{t-1}}\}$ is used to find a candidate solution \hat{S}_k resulting from solving our version of the max-coverage problem, defined in Algorithm 5, *Max-Coverage*(*WRRs*_t, *k*, *S*_N). This consists of choosing a candidate set $\hat{S}_k \subseteq V \setminus S_N$ of size *k*, with the highest coverage of *S* among the *WRRs*. This set \hat{S}_k is aiming at maximizing the number of nodes out of those expected to get infected that are reached by the positive information before the negative one, which is a way in which we can achieve to save the most nodes. This is the logic behind all heuristic methods, choosing the most influential nodes. The second set $WRRS_t = \{WRR_{\Lambda 2^{t-1}}, ..., WRR_{\Lambda 2^t}\}$ is used to verify the quality of the candidate solution \hat{S}_k . If the qualifying conditions are met, $WRRS_t$ is sufficient to obtain a good influence estimate, and thus a good blocking power estimate for the *WRRS* method. Then the *WRRS* is returned. If the conditions are not met in iteration *t* the process is repeated, with $WRRS_{t+1} = WRRS_t \cup WRRS_t^c$.

The quality conditions to be met are the following two. First the number of trees in $WRRS_t^c$ shall be sufficient to guarantee a $(\epsilon, \frac{\delta}{3*t_{max}})$ -approximation of the influence power of \hat{S}_k , where t_{max} is defined as follows:

$$t_{max} = \lceil log_2(\frac{2*N_{max}}{\Upsilon(\epsilon,\delta/3)}) \rceil$$

The computation of the parameters for this condition differs slightly from the original algorithm since the WRRs are not rooted uniformly over the set S. We then keep track of the frequency of the roots to compute a weighted version of the formulas.

If this first condition is met, it proceeds to estimate the best possible precision parameters $\epsilon_1, \epsilon_2, \epsilon_3$. The second condition states that the combination of those precision parameters has to be sufficiently small, i.e.,

$$\epsilon_t = (\epsilon_1 + \epsilon_2 + \epsilon_3)(1 - 1/e - \epsilon) + (1 - 1/e)\epsilon_3 \le \epsilon$$

When these conditions are met the returned set of WRRs, namely $WRRs_t$, is sufficient to achieve a $(1 - 1/e - \epsilon)$ -approximation of the influence power, with high probability.

We extend the D-SSA algorithm to contain a third condition that does not alter the validity of the first two conditions. This third one is met when the number of WRRs that contains a negative seed node is larger than Λ_1 . This ensures that the WRR trees shall be sufficient to guarantee a $(\epsilon, \frac{\delta}{3*t_{max}})$ -approximation of the influence power of S_N . This is necessary to make accurate predictions on how to block such influence.

In the unfortunate event that the algorithm does not satisfy the conditions for any t, a stopping criterion is defined. The algorithm will terminate when the number of WRRs reaches a cap

$$N_{max} = 8\frac{1-1/e}{2+2e/3} * \Upsilon(\epsilon, \frac{\delta}{6} / \binom{n}{k}) * \frac{n}{k}$$

where $\Upsilon(\epsilon, \delta) = (2 + \frac{2}{3}e) * ln(\frac{1}{\delta} * \frac{1}{\epsilon^2})$. This way it is guaranteed a O(log(n)) maximum number of iterations. In this case, n is defined to be the number of nodes with expected infection, thus the size of S.

Each WRR is generated by calling generate_ $WRR(G,S,probs,S_N)$, presented in Algorithm 6. To further clarify this process a toy example was presented in Figure 4.1. First, a root r is chosen at random from the set S using the probabilities in probs (Line 3). Including these probabilities helps boost the performance of the method, as the FWRRS method is interested in those WRRs that include a negative seed node. The probability of choosing node

v out of S is then $\frac{probs_v}{\sum\limits_{w \in S} probs_w}$. From this node, we begin to visit G in a breadth-first search

manner. From the node u currently being considering we check all of its predecessors, and for each $v \in \text{predecessors}(u)$ we add v to WRR with probability $p_{v \to u}$. For every new node v being considered for the WRR the probability of the path towards the root is stored in p_v (Lines 9:13). If v is already in WRR we add a copy of v (Lines 19:26). However, if any copy of v is already present in the path between v and r we do not add it again, as in the ICM a node can not activate twice and this path would not be possible (line 18). Once a node is added to WRR we also add it to A which is a list of nodes to consider next. The process stops when a node in S_N is to be added to the WRR, in this case, we stop the iteration at the level of G, as no nodes at a distance larger than this will be able to save r from getting infected. In this case, the property $.has_S_N$ of the WRR is True and the WRR is then returned, after all, nodes in this last level are considered. In the other case that A is empty, meaning we do not have more nodes to consider, the process also stops. The property $.has_S_N$ of the returned WRR is then False. Last, before returning the final WRR, nodes with a lower probability of reaching the root than any negative seed in the tree are removed (Lines 36: 39).

The list *paths* was introduced to boost the performance of this algorithm under the following scenario. When the probabilities of infection in the edges are large, the size of the WRRtrees can significantly increase, affecting the performance of the method. The number of copies of nodes grows significantly with the number of paths through which information now diffuses in this network. However, not all paths need to be considered. If a node is present in a WRR it is assumed that its information will reach the root node, thus having multiple copies of this node does give us any additional information. However, the inclusion of a copy of this node might be necessary to cover a path through which the root can get infected. We then have the variable $paths_r$ which contains the paths through which the root r was infected in the 1000 simulations previously performed. If a new node v is to be added to a WRR, where already numerous copies of a such node are present, we use $paths_r$ as follows. If the path we are currently iterating over in the WRR, namely descendants_u $\cup \{v\}$, does not lead to the infection of the root, because it is not a subset of any path in $paths_r$ we will not consider it any further. This allows for the pruning of the WRR trees significantly as the consideration of numerous unnecessary paths is avoided (Line 20). The maximum number of copies is set to 10, and the method will then be considering at least 10 different paths through which node v can inform r. Experiments show that a larger maximum number of copies will not have a significant improvement for blocking power on all evaluated data sets, in fact, under the Weighted cascade model (for each edge $(u, v) \in E$ $p_{u \to v} = 1/\text{in-degree}$ of v) it is rarely reached. Only when the probabilities become larger like under the Uniform cascade (for each edge $(u, v) \in E$ $p_{u \to v} = p$) with large values of p, comes the limit into significant use.

Algorithm 6 generate_WRR(G, S, probs, paths, S_N)

1: $WRR \leftarrow \text{empty graph}, max_depth \leftarrow \infty$ 2: $WRR.has_S_N \leftarrow False$ 3: $r \leftarrow \text{random node in } S$ chosen using the probabilities in probs 4: $WRR.root \leftarrow r$ 5: $descts_r \leftarrow \emptyset$ 6: $A \leftarrow [r]$ 7: $original_r \leftarrow r$, $s_r \leftarrow 0$, $copies_r \leftarrow 0$, $prev_r \leftarrow \emptyset$ 8: while A is not empty do 9: $u \leftarrow \text{take first element out of } A$ 10: if $prev_v = \emptyset$ then 11: $p_v \leftarrow 1$ 12:else 13: $w \leftarrow prev_v$ 14: $p_v \leftarrow p_{original_w \to original_v} * p_w$ if $original_u \notin S_N$ then 15:16:if $s_u \leq max_depth$ then for $v \in V$ such that the edge $(v, original_n) \in E$ do 17: add_v is True with probability $p_{v \to original_u}$ else False 18:if add_v and $v \notin descts_u$ then 19:if $v \in WRR$ then 20:if $copies_v \leq 10$ or $(descts_u \cup \{v\} \subseteq p \text{ for any } p \in paths_r)$ then 21:22: $copies_v \leftarrow copies_v + 1, i \leftarrow copies_v$ Add node v_copy_i and edge (v_copy_i, u) to WRR23: $descts_{v_copy_i} \leftarrow descts_u \cup \{u\}$ 24:25: $s_{v_copy_i} \leftarrow s_u + 1$ add v_copy_i to A 26:27: $prev_{v_copy_i} \leftarrow u$ 28:else Add node v and edge (v, u) to WRR29:descts $v \leftarrow descts u \cup \{u\}$ 30: $s_v \leftarrow s_u + 1$ 31: add v to A32: $prev_v \leftarrow u$ 33: 34: else $max_depth \leftarrow s_u$ 35: $WRR.has_S_N \leftarrow True$ 36: 37: $min_prob \leftarrow \max_{v \in WRR \cap S_N} p_v$ for $v \in WRR$ do 38: if $p_v < min_prob$ then 39: remove v from WRR40: 41: return WRR

3. Choose $\mathbf{S}_{\mathbf{P}}$. In Algorithm 1 once the pool of WRRS has been generated, the infection probability for each node $v \in S$ according to the WRRS is computed (Lines 5:8). This is equal to the ratio of WRRs rooted at v that includes a node in S_N . Once the probability of infection for all nodes in S is computed, the expected number of initial infections per community *init_inf* has also been computed (Line 8). The initial expected number of infections in community c is set to be the sum of the infection probabilities of all nodes $v \in V$ with $comm_v = c$, meaning the community of v is c. Further, the variable C_inf is set to contain those communities where infections take place (Line 9). Next, the percentage of saved nodes per community pct_saved is initialised to 0 and the number of infections currently being expected, exp_inf , is initialised to be equal to the initial ones (Line 11).

The blocking power of v per community, B_v , for all $v \in V$ is then initialized to 0 (Line 12), as well as the maximin_value. In Line 12 the WRR trees in WRRS_{inf}, which are those with WRR.has_ $S_N = True$, are iterated over. If any node v in a WRR were to be added to S_P the root r of this WRR would be expected to be saved from being infected within such tree, as v would reach it with its information before the S_N nodes. Thus, the number of WRRs in which r gets infected reduces by 1, and the probability of infection of r reduces by 1/number of WRRs rooted at r. This means that the value of $B_v[comm_r]$ is increased by 1/number of WRRS rooted at r. This logic is then applied to compute the blocking power in each community for any node. Additionally, to avoid iterating over the entire set of nodes of the network, a set *candidates* is created to contain those nodes with blocking power.

Next, nodes for S_P begin to be selected, which was initialized in line 11 to an empty set. In each iteration, the node that achieves the highest maximin value, meaning that it saves the highest percentage of all communities, is chosen, defined as selection step 1. But since the infections probabilities are (ϵ) -approximations, the set of candidates is established to be those nodes with a maximin value within a range of ϵ from the maximum possible (16:17). In practice it was noticed, that such interval was too large, especially in later iterations, where the maximin value becomes larger. To reduce the size of this interval, and force the sub-sampling of candidates, a parameter Φ_1 was introduced. The subset of candidates options_1 is then set to be those nodes with a maximin value within a range of $\epsilon * \Phi_1$ from the maximum possible. This parameter is set to be the ratio of initial infections that take place in the community with a minimum percentage of saved nodes. The intuition behind this parameter is that the error made is expected to be balanced throughout the network. In this case, the expected error in the number of saved nodes in a community, should be proportional to the expected number of infections that were taking place in such a community. Thus, $\Phi_1 = \frac{exp \cdot inf[c]}{\sum_{c \in C \cdot inf} exp \cdot inf[c]}$, where c' is the community with the smaller proportion of saved nodes.

In the case in which multiple candidates are selected as *options*_1, especially in the initial iterations where no single node can achieve a maximin value higher than 0, we break ties as follows. First, we perform the selection step 2. In the case that some communities have not been saved at all, those nodes that achieve some saving within these communities, are selected. Such requirement shall help the method escape out of the current maximin-value of 0 (Lines 19:23). In practice it was encountered that when the number of infections taking place in these neglected communities was very low and these were already connected in some way to S_P , forcing their help came at a significant cost. The gain from forcing such help was very low, as the number of infections is small, and in practice, the benefits of the intervention would in some way reach these groups. Thus we decide to skip this sampling step in such a scenario. This is the case when the portion of infections in these communities is smaller than $\epsilon * (\frac{|C-no_help|}{|C_inf|})$ and there exists a path from S_P to all the neglected communities (Line 22).

Next, those candidates that save the most nodes from those communities that currently have received the least help are selected, defined as selection step 3 (24:26). These communities are defined to be those who have a saving rate below average. For each node in *options*_1, the number of nodes that are saved from the communities with below-average help is computed. In such computation, nodes have weights, inversely proportional to the ratio of saved nodes in their community. Such weights shall enforce prioritizing saving nodes from those communities with a saving ratio significantly below average over those with a slightly below average saving ratio. Similar to the previous selection those candidates with a number of saved nodes in the range of ϵ from the maximum possible, are sampled, but such selection becomes very restrictive, especially at later iterations when the number of nodes that can be saved reduces. To overcome this, the selection is loosened with the use of parameter $\Phi_2 = \sum_{c \in C.min.help} init_inf[c] - exp_inf[c] * avg_help$, where $avg_help = \sum_{c' \in C.inf} pct_saved_{c'}/|C_inf|$. The intuition, behind this parameter value, is

that as the number of saved nodes increases, the error to be made in the next steps reduces since most of this approximation error has already been made. Thus, this parameter allows for more room for selection in later iterations.

In practice the previous selections still give room for further subsampling, thus we perform
the selection step 4 next. In this step, those nodes with the highest blocking power within those communities with a saving ratio around the current maximin value are selected. Such selection will help the method increase the maximin value since it tries to maximize the help around those communities that might be causing such a low maximin value (Lines 27:29). In this step $\Phi_3 = \sum_{c \in C_min_help} int_inf[c] - exp_inf[c] * \epsilon$, this value follows the same intuition as the previous Φ_2 . Last, in the case that the maximum blocking power of any node in *options_3* is low, we perform the selection step 5. We then guarantee that we choose a node $v \in options_3$ with a blocking power within a range around the maximum possible, by removing from *options_3* those nodes that do not. The maximum blocking power is considered small if this value is outside an ϵ range around the maximum number of nodes a candidate in *options_1* could save, which was computed in line 18. With this, in the scenarios in which the sub-sampling steps 2-4 caused the choice of nodes with limited blocking power,

we at least choose the node with maximum blocking power out of these candidates, limiting

The node u is chosen in the last selection step 6, to be the node out of the sub-sample of candidates with the highest weighted out-degree. The weight of a node is its probability of getting infected with the current S_P . The selected node is then added to the positive seed and removed from *candidates* (Lines 32:33). Once a new node u is added to S_P , the expected infections are then those after u becomes a truth campaigner. Moreover, the maximin value and ratio of saved nodes per community are also modified (Line 34). Next, the blocking power of any other node is updated. This is done in Lines 35:39, where we iterate over those $WRRs \in WRRS_{inf}$ in which the root r is expected to be reached by u. In those WRRs covered by u, the root will remain saved and the expected number of saved nodes of all those nodes that were not present in this WRR is increased. Thus, the expected infections of all those nodes that were not present in this WRR are reduced by 1/number of WRRS rooted at r. Further, this WRR is removed from the set of WRRs that need to be covered, $WRRS_inf$.

This process is repeated iteratively, by choosing a new node to add to S_P in each iteration. The process finalizes when the size of S_P is k, and then the resulting set of truth campaigner is returned.

4.2 Parity seeding for FIBM problem

the cost of fairness as much as possible.

The work of Stoica in the area of Fairness-aware Influence Maximization [71, 72], was introduced in Section 3.2. She shows how the nature of the independent cascade model can be exploited to achieve a better outreach by favouring parity. Including sensitive features in the input of the seed selection substantially improves such parity and it is shown that often the efficiency is not affected or even is proved a small gain. In Stoica's work parity seeding is then defined for the IM problem in a bi-populated network as follows:

Given the population, V of the network, consisting of red(R) and blue(B) nodes, the seed set's S population is based on two differential thresholds $k^{R}(n)$ and $k^{B}(n)$ as

$$\begin{split} S^R_{k^R(n)} \cup S^B_{k^B(n)} &= \{ v \in R | \deg(v) \ge k^R(n) \} \cup \{ v \in B | \deg(v) \ge k^B(n) \} \\ \text{such that } |S_{k(n)}| &= |S^R_{k^R(n)} \cup S^B_{k^B(n)}| \text{ and } \frac{|S^R_{k^R(n)}|}{|S^B_{k^R(n)} \cup S^B_{k^B(n)}|} = \frac{|R|}{|V|}. \end{split}$$

The main idea of this seeding method is that instead of choosing the nodes with the highest degree, or equivalently those with a degree above a certain threshold k(n) (agnostic seeding), we set a selection threshold per group. The thresholds are then defined in such a way that the ratio of each group in the seed set is as in the general population while preserving the seed set budget. In Figure 4.3, extracted from the work of Stoica [72], the different thresholds defined for the seeding mechanisms are displayed. Here another mechanism *diversity seeding*, where the thresholds are defined somewhere between the agnostic and parity seeding, is also plotted.



Figure 4.3: Differentiated thresholds for strategic seeding, extracted from [72]

To adapt this seeding mechanism to the IBM problem, we set the thresholds in such a way that the seed set population resembles that of the negative seed set S_N . This way the positive information will be spread fairly and efficiently over the same communities that S_N can reach. The adaptation of the parity seeding for the IBM problem is defined in Algorithm 7.

Algorithm 7 Stoica $(G(V,E),S_N,k)$

1: $num_com \leftarrow$ number of communities in G, defined as |C|2: community_ $S_N \leftarrow$ list of length num_com consisting of 0s 3: for $s \in S_N$ do $c \leftarrow comm_s$ 4: $community_{-}S_{N}[c] += \frac{1}{|S_{n}|}$ 5:6: $S_P \leftarrow \emptyset$ 7: $take_c \leftarrow \lfloor community_S_N[c] * k \rfloor$ for $c \in C$ 8: while $\sum_{c \in C} take_c < k \operatorname{do}$ $c \leftarrow \arg\max(community_S_N[c] * k - take_c)$ 9: $take_c \stackrel{o}{\leftarrow} = 1$ 10: 11: for $c \in C$ do select $take_c$ number of nodes with the highest degree from community c 12: 13:add them to S_P 14: return S_P

The process begins by computing the ratio of each community in the negative seed S_N (Lines

3:5). Next the equivalent ratio of k is allocated to each of the communities. To make sure we do not go over the budget k the computations are floored (Line 7). Next, if resources are still to be allocated, meaning there is still a budget left, we give these to those communities with the biggest difference between population ratio in S_N and S_P . Once the number of truth campaigners allocated to each community $c \in C$ is defined, we select this amount of nodes with the highest degree in C and add them to S_P (Lines 11:13). The method finishes by returning S_P .

4.3 Fairness-aware CMIA-O

A second fairness-aware benchmark for the IBM problem is introduced in this section. This is the Maximin-CMIA-O, which is an extension of the work of Wu [75], shortly introduced in Section 3.1.

The presented approach CMIA-O was altered to have as an objective function *maximin*, instead of maximizing the number of saved nodes. The remaining computations stayed the same. The structural changes to be able to compute the maximin values efficiently are presented in the pseudo-code of the new algorithm Maximin-CMIA-O in Algorithm 8.

Algorithm 8 Maximin-CMIA-O(G(V,E), S_N , k, θ)

1: $S_P \leftarrow \emptyset$, $NegS \leftarrow \emptyset$ 2: $DecInf_c(v) \leftarrow 0$ for each node $v \in V$ and $c \in C$ 3: $ExpInf_c \leftarrow 0$ for each community $c \in C$ 4: for $u \in S_N$ do 5: construct MIOA(u, θ) $NegS \leftarrow NegS \cup (MIOA(u, \theta) \setminus S_N)$ 6: 7: for $u \in NegS$ do 8: construct MIIA(u, θ) compute $ap^N(u, S_N, S_P, MIIA(u, \theta))$ 9: 10: $ExpInf_c + = p^N(u, S_N, S_P, MIIA(u, \theta))$ for $v \in MIIA(u, \theta)$ do 11:compute $ap^N(u, S_N, S_P \cup \{v\}, MIIA(u, \theta))$ 12:13: $c \leftarrow comm_u$ $DecInf_c(v) + = ap^N(u, S_N, S_P, MIIA(u, \theta)) - ap^N(u, S_N, S_P \cup \{v\}, MIIA(u, \theta))$ 14:15: **for** i = 1 to k **do** $\underset{C \in C}{\operatorname{arg\,max}} \min_{c \in C} \frac{\operatorname{DecInf}_c(v)}{\operatorname{ExpInf}_c}$ 16: $v \in V \setminus (S_P \cup S_N)$ construct MIOA(u, θ) 17:for $v \in MIOA(u,\theta)$ do 18: 19: $c \leftarrow comm_v$ 20: for $w \in MIIA(v, \theta)$ do $DecInf_c(w) = ap^N(v, S_N, S_P, MIIA(v, \theta)) - ap^N(v, S_N, S_P \cup \{w\}, MIIA(v, \theta))$ 21: $S_P \leftarrow S_P \cup \{u\}$ 22: for $v \in MIOA(u,\theta) \setminus \{u\}$ do 23: 24: $c \leftarrow comm_v$ compute $ap^{N}(v, S_{N}, S_{P}, MIIA(v, \theta))$ 25:for $w \in MIIA(v, \theta)$ do 26:compute $ap^N(v, S_N, S_P \cup \{w\}, MIIA(v, \theta))$ 27: $DecInf(w) + = ap^{N}(v, S_{N}, S_{P}, MIIA(v, \theta)) - ap^{N}(v, S_{N}, S_{P} \cup \{w\}, MIIA(v, \theta))$ 28:29: return S_P

The process covered in Algorithm 8 consists of the following. First, the set of nodes expected to get infected is computed, NegS. This consists of those nodes that have a path to any negative

seed with a probability smaller or equal to θ . This set of nodes is obtained by constructing a Maximum influence Out-Arborescence (MIOA) from each of the negative seeds, with the use of an algorithm such as Dijkstra, where the length of an edge (u, v) is set to be $-log(p_{u\to v})$. Then those nodes with a path length smaller than θ are included in NegS (Lines 4:6).

For each node v in NegS we then compute the probability of infection ap^N . This is performed by constructing Maximum influence In-Arborescence (MIIA), from v, constructed similarly to the MIOA but with forwarding edges instead of backward edges. The MIIA of v is then used to compute the probability v will get infected by S_N in this MIIA. This computation remains unchanged from the one proposed by Wu [75], which relies on the cascading effect of the information over the tree. For clarity, we will cover an example, and compute ap^N of u over the MIIA visualized in Figure 4.4, assuming all edges have probability 0.5 and that the negative seed includes nodes a, d, j.



Figure 4.4: Differentiated thresholds for strategic seeding

The probability of infection of any node v in the MIA at any time step t is initialized to be 0, $p^N(v,t) = 0$, except for those nodes in S_N which are initialized to 1 at time step 0. The probability of any node being infected by time step t is then defined to be the sum of probabilities of it being infected at any time step before, $ap^N(v,t) = \sum_{t' \le t} p^N(v,t)$.

The node u has a direct neighbour who could be infected in the previous time step j since $p^{N}(j,0)$ is not 0 as it's a negative seed. Then, at time step 1 u can get infected with probability 0.5, since $p_{j\to u} = 0.5$, $p^{N}(u,1) = 0.5$. Similarly $p^{N}(f,1)$ and $p^{N}(l,1)$ are also set to 0.5.

At time step 2, l can attempt to infect u. The probability of infection at this time step is computed as follows $p^{N}(u,2) = (1 - ap^{N}(u,1)) * 0.5 * p^{N}(l,1) = 0.125$. The probability of ubeing infected during time step 2 is equal to the probability of it not being infected before such time step, multiplied by the probability of any of its neighbouring nodes being infected during the previous step and reaching u. The probability of u not being infected before, during time step 1, is 1 - 0.5 = 0.5. The probability of it being reached by a node infected in the previous step is

$$\sum_{v \in MIIA} p_{v \to u} * p^N(v, 1) = 0.5 * p^N(l, 1) = 0.25$$

Node f can also attempt to infect l and similarly $p^{N}(l,2) = (1 - ap^{N}(l,1)) * 0.5 * p^{N}(f,1) = 0.5 * 0.25 = 0.125.$

Last, node u can also be reached at time step 3 by l again, since this node could be infected at time step 2. Then $p^N(u,3) = (1 - ap^N(u,2)) * 0.5 * p^N(l,2) = 0.0234$. Since no more nodes can get infected the process finishes. The total probability of infection of node u is then set to be the probability of it being infected by time step 3, since this is the longest path from any negative seed, and no further infections of u take place. This is sum of the probabilities of u being infected

at each time step

$$ap^{N}(u, S_{N}, S_{P} = \emptyset, MIIA) = ap^{N}(u, 3) = \sum_{t \ge 0} p^{N}(u, t) = 0.5 + 0.125 + 0.0234 = 0.6484$$

In Algorithm 8, after we have computed $ap^{N}(u, S_n, S_P, MIIA)$ for each node u in NegS, we can compute the expected infections per community by summing them together (Line 10). This is done to be able to compute the maximin values later on. Additionally, we compute the probability of infection of each node u in NegS if any other node v in the MIIA from u was to be added to S_P . The decrease in infection probability is then added to the variable $DecInf_c(v)$, where c is the community u belongs. This way we have computed the expected decrease in infections per community for all our candidate nodes for S_P , namely those that have a path with a probability of at least θ from a node in NegS(Lines 7:14).

Then the node with the highest maximin value is iteratively chosen until the size of S_P is of k. When a new node u is chosen to be added to the positive seed, we then proceed to update the DecInf values of all other nodes which might be affected by this new S_P node. We update the DecInf of all those nodes who were reaching any of the nodes we now expect will be reached by the positive information that u will be spreading, those are the nodes in $MIOA(u, \theta)$. For each of these nodes v, now reached by u, we iterate over the nodes w present in $MIIA(v, \theta)$, the nodes that can also reach v. We remove the decrease in infection w had in node v(Line 21). We then add u to S_P (Line 22) and we recompute the DecInf of those same nodes, except for u, as this does not change (Lines 23:28). Once S_P consists of k nodes, the process stops and we return this variable.

When a new node is added to S_P the values of ap^N are recomputed. But, since now nodes also spread the true information, thus blocking the spread of the negative one in some cases, the computations change. We cover the same scenario as described previously in Figure 4.4, but with the inclusion of a positive seed node, namely f.

The probability of infection by either positive or negative information of any node v in the MIA at any time step t is initialized to be 0, $p^N(v,t) = 0$ and $p^P(v,t) = 0$, except for those nodes $v \in S_N$ for which $p^N(v,0) = 1$ and those $v' \in S_P$ for which $p^P(v',0) = 1$.

At time step 1, since now f is part of the positive seed it can not be infected by a. The infection that still takes place in time step 1 is the one of node u, $p^N(u,1) = 0.5$. Additionally, g can negatively infect node l, but node l can also be positively infected in this step by node f, the computation is as follows. Since we assume a node will believe the positive information from the negative one, $p^P(l,1) = 0.5$. On the other hand, $p^N(l,1) = p_{g \to l} * p^N(g,0) * (1-p^P(l,1)) = 0.25$.

In the next time step, l which could have been positively infected in the previous step, will try to reach node u. The probability of succeeding in this time step is then

$$p^{P}(u,2) = (1 - ap^{P}(u,1)) * (1 - ap^{N}(u,1)) * (1 - \prod_{v \in MIIA} (1 - p^{P}(v,1) * p_{v \to u}))$$

= 1 * 0.5 * (1 - (1 - 0.5 * p^{P}(l,1))) = 0.125

l could have also been negatively infected in the previous step, and could then try to reach node u. The probability of node u getting negatively infected in this time step is then computed as follows:

$$p^{N}(u,2) = (1 - ap^{N}(u,1)) * (1 - ap^{P}(u,1)) * (1 - \prod_{v \in MIIA} (1 - p^{N}(v,1) * p_{v \to u}))$$

*
$$\prod_{v \in MIIA} (1 - p^{P}(v,t) * p_{v \to u})$$

= 0.5 * 1 * (1 - (1 - p^{N}(l,1) * 0.5)) * (1 - p^{P}(l,1) * 0.5)
= 0.046875

This computation comes from the fact that node u becomes negatively activated during time step 2 if at least one of its neighbours became negatively activated in the previous time step and it successfully reaches u, while u was neither negatively nor positively activated already. Additionally, no neighbour of u must have been positively activated and successfully reached u, since in this case, u would then believe the positive information. As a result, the probability of u getting infected is now

$$ap^{N}(u, S_{N}, S_{P}, MIIA) = ap^{N}(u, 2) = \sum_{t>0} p^{N}(u, t) = 0.5 + 0.046875 = 0.546875$$

In this case the value of DecInf(f) would be increased by 0.101525, as it reduces the probability of u being infected by this much.

Chapter 5 Experiments

In this chapter, we will experimentally analyze the performance of the FWRRS method. These experiments shall evaluate the methods under real-life scenarios, for this a set of real data sets which are publicly available, was used to observe this. To be able to understand how the characteristics of the network might impact the methods' effectiveness and fairness experiments in a set of synthetic datasets generated with a proposed model, were performed. The chapter is organized as follows, first, in Section 5.1 the experiments performed in real data sets are introduced. This Section first presents the dataset used in Section 5.1.1, later the actual experiment settings in Section 5.1.2 and later the results 5.1.3. Next in Section 5.2 the experiments performed in synthetic data sets are introduced. In section 5.2.1 we introduce the need for a novel model for the generation of networks that resemble social network characteristics, such a model is presented in Appendix B. Next in section 5.2.2, the characteristics of the generated networks for the experiments as well as the evaluation findings on such networks are presented.

5.1 Real datasets experiments

5.1.1 Datasets

In this subsection, each data set that will be utilized in the experiments is introduced, and the characteristics of these social networks are summarised in Table 5.1. The chosen datasets consist of two friendship networks, namely *Facebook* and *Twitter*, three collaboration networks, *NetPH*, *NetHEPT* and *NetASTRO*, and a communication network, *Enron-Email*. Such a variety of networks aims at covering multiple types of social interactions to observe the performance under different social networks.

- Facebook: consists of a snapshot of the online social network Facebook containing 4039 nodes and 88234 undirected edges, which are represented with bidirectional edges, bringing the total number of directed edges to 176468. In the directed version of this network, the average out-degree is 43.69, the clustering coefficient is 0.6055, the average shortest path is 3.69 and the diameter is 8. The degree distribution is plotted in Figure 5.1(a), where we observe heavy tail characteristics of a power-law distribution. However, this dataset's degree distribution has top concavity, which might suggest this network is not purely scale-free. This dataset is commonly used in previous research on the IBM problem [83, 76, 80, 81], was extracted from the work of Mcauley [44] and is available in SNAP datasets [39].
- Twitter: consists of a snapshot of the social network of Twitter with 81306 nodes and 1768149 edges. It was also extracted from the work of Mcauley [44] and is available in SNAP datasets [39]. This network has a diameter of 7, an average degree of 21.74, a clustering coefficient of 0.5653 and an average shortest path of 4.131. It has also been utilized

in previous research on fake news mitigation such as [73, 64]. In this social media platform, friendships are not bidirectional, thus an edge between any pair of nodes (u, v), translates into u is a friend of v, thus the resulting graph is directed.

- NetPHY: is a co-authorship network created using publications in the Physics section of the e-print arXiv [3]. This dataset has been used in research some examples are the following [18, 30, 69] and is publicly available under the research of Wei Chen [16]. In this network, two authors are connected by an edge if they have a publication together. It contains 37149 nodes and undirected 231584 edges, but since in our networks we do not allow for multiple edges, the resulting number of directed edges is 174161. The resulting network consists then of disconnected subgraphs. For the largest component, the average shortest path is 6.26, the average out-degree of 9.376 and the diameter of 19. The clustering of the component is 0.75 and the degree distribution is plotted in Figure 5.1(c).
- NetHEPT: is also a co-authorship network created using publications in the High Energy Physics section of the e-print arXiv [3] and is publicly available under the research of Wei Chen [16]. This network is disconnected, the biggest component has a diameter of 19, an average shortest path of 5.779, an average degree of 4.1205 and a clustering coefficient value of 0.49. Additionally, the degree distribution is plotted in Figure 5.1(d). The dataset contains 15235 author nodes and a total of 58891 undirected connections between them, where two authors are connected with they share a publication. After removing repeated edges the total number of directed edges is 62776. This network has been extensively used in the study of fake news mitigation, some examples are the following publications [30, 18, 69, 27, 53, 54, 47, 17].
- NetASTRO: is a collaboration network extracted from the e-print arXiv [3]. It represents scientific collaborations, where the nodes are authors and an edge between them resembles that they published a paper together in the category of Astro Physics. This network consists of an undirected graph of 18772 nodes and 396160 edges. The diameter of the biggest connected component is 14, the clustering coefficient is large, 0.6305, as well as the average degree, is 21.10. The average shortest path remains small, characteristic of social networks, at 4.19. The degree distribution, which resembles a power-law distribution, is plotted in Figure 5.1(e). This dataset was extracted from the work of Leskovec [38] and is publicly available in SNAP datasets [39].
- Enron-Email: is a communication network covering all email communication within a dataset of approximately half a million emails [37]. In this graph, the nodes represent the email addresses and all edges between u and v represent that u and v established email communication at least once. This way the network consists of 36692 nodes connected via 183831 undirected edges, with an average out-degree of 10.02. The nodes out degree follows a strong power law distribution, characterized by its linear appearance in the log-log scale, as represented in Figure 5.1(f), where we observe a stronger power law distribution than in the previous datasets. The network is not connected, the biggest component has a high clustering coefficient of 0.4969, a diameter of 13 and an average shortest path of 4.02. This dataset was extracted from the work of Leskovec [40] and is publicly available in SNAP datasets [39].



Figure 5.1: Degree distribution of the datasets

Data set	Nodes	Edges	Diameter	Clustering	Average	Average	Number of
				Coefficient	degree	shortest path	communities
Facebook	4039	176468	8	0.6055	43.69	3.69	16
Twitter	81306	1768149	7	0.5653	21.74	4.13	74
NetPHY	37149	231584	19	0.75	9.38	6.26	3959
NetHEPT	15235	62776	19	0.49	4.12	5.78	1821
Enron-Email	36692	367662	13	0.4969	10.02	4.02	1245
NetASTRO	18772	396160	14	0.6305	21.10	4.19	322

Table 5.1: Datasets characteristics

5.1.2 Experiment settings

In this section, the characteristics of the various experiments to be performed are introduced. As stated in previous sections, the propagation model utilized for the diffusion of information is the ICM. The parameters which are then left to define are S_N and the edge probabilities.

For these experiments, the **Weighted Cascade** (WC) model will be used to generate propagation probabilities. This is a model that resembles numerous of the characteristics of real-life diffusion and is extensively used for evaluation in IBM research, some examples are [15, 46, 83]. The impact of other models to define the probabilities will also be evaluated later in this section by evaluating the performance of some data sets under the **Uniform** model, utilized by previous researchers such as [25].

The Uniform model sets all edge probabilities to a constant value, in this case, this is p = 0.05. Larger values of p, cause all methods' effectiveness to converge for very low values of k. Since information diffuses very easily, the choice of S_P does not have a significant impact on the diffusion of the true information, and as result, all methods save the majority of the graph from infection. This phenomenon is shown in Figure A.1, which can be found in Appendix A. To be able to overcome such insensitivity to different methods, we then choose a smaller p-value.

The WC model, on the other hand, introduced by Kempe [35] sets the probability for every edge (u, v) to be $p_{u \to v} = 1/d_{-i}n_v$ where $d_{-i}n_v$ is the in-degree of node v. It defines the probability of a node v believing the information another node u is sharing, to be proportional to the number of sources of information v has. The least informed a node is, the more vulnerable it will be to believing the information from one of its neighbouring nodes. The independent cascade model with uniform probabilities in the edges has the property that nodes with a high degree not only have a chance to influence many other nodes but also to be influenced by them. This is then corrected in the WC model. The WC model then introduces the famous "celebrity effect" which is observed in real life, He [30] showed that on Twitter this effect drives the diffusion of information through retweets. This effect states that the trustworthiness of a node is directly related to its popularity. Such an effect is often exploited through marketing campaigns. Under the WC the ICM resembles the linear threshold model, another extensively used propagation model, in that the expected number of neighbours who would succeed in activating a node v is 1.

For the evaluation, the negative seed S_N is set to consist of 50 nodes. To evaluate the performance under various seeds, and observe the performance of the methods disregarding the choice of rumour spreaders, we evaluate the methods under 2 scenarios. The first one is in which the negative seed is chosen at random. And another scenario in which the choice of S_N is powerful, namely we choose the nodes with the highest degree.

Additionally, since in all experiments the communities detected by the Louvain algorithm are used as ground truth communities, the impact of using other community detection methods is evaluated. For this, the performance change in fairness and percentage of saved nodes of all methods is measured when using various algorithms.

All experiments were executed under the same conditions, as they were performed in a server with the following characteristics:

- CPU: 1x Intel Xeon E5-2698v4 @ 2.2GHz (40 hyperthreads)
- RAM: 256GB
- GPU: 4x Nvidia Tesla V100 (16GB RAM, 2560 tensor cores, 10480 CUDA cores, compute capability 7.0)

5.1.3 Experiments Evaluation

5.1.3.1 Evaluation under the WC model with random negative seed

In this section, we will be introducing the results of the experiments under the WC model with a random negative seed. The results of these experiments are visualized in Figure 5.2, Figure 5.3, Figure 5.4, Figure 5.5, Figure 5.6, and Figure 5.7.



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.2: Experimental analysis in the Facebook dataset under the WC model with Random ${\cal S}_N$



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.3: Experimental analysis in the Twitter dataset under the WC model with Random S_N



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.4: Experimental analysis in the NetPHY dataset under the WC model with Random S_N



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues $% \left({{\left({{{\bf{n}}_{\rm{s}}} \right)}_{\rm{saved}}} \right)$

Figure 5.5: Experimental analysis in the NetHEPT dataset under the WC model with Random ${\cal S}_N$



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.6: Experimental analysis in the NetASTRO dataset under the WC model with Random S_N



(a) Evaluation on Number of saved nodes, with vary- (b) Evaluation on maxi-min value, with varying k valing k values

Figure 5.7: Experimental analysis in the Enron-Email dataset under the WC model with Random ${\cal S}_N$

Under the WC with random set S_N we observe the unfairness of the existing methods. Under these experiments, the only methods that achieve a maximin value above 0, overall datasets, are those with a fairness-aware objective. These are the methods being proposed in this thesis, *Maximin-CMIA-O*, *Stoica* and *FWRRS*. This means that the existing methods tend to neglect communities when solving the IBM problem, showing the need for fairness-aware methods. It is only under the Facebook dataset that other benchmarks, *CMIA-O* and *BIOG*, achieve a maximin value higher than 0. The low number of communities in this dataset 16, followed by the high average degree 43.69, makes spreading the true information within all communities more accessible for any method. Additionally, this network, in particular, is connected, thus the help provided to a community could potentially also reach other communities, whereas in disconnected graphs resources need to be split between infected components. As a result all communities that get infected, are reached by these methods, without the need for a fairness-aware objective. But such a scenario is not repeated in any of the other datasets, where the number of communities is significantly larger. In these other data sets, the number of small communities is larger and so is the number of which get infected. As a result, the existing methods are more likely to neglect such minority communities for the larger good, a larger percentage of saved nodes over the whole network. In these experiments, it can be observed that FWRRS offers the highest fairness value in overall datasets.

Regarding the total percentage of saved nodes, we observe the inconsistency in performance between the methods. In Figures 5.2(a), 5.3(b), 5.6(a) and 5.7(a) we observe the power of selecting S_P based on the degree of the nodes. For these datasets, Facebook, Twitter, NetASTRO and Enron, the best performing method, after the proposed FWRRS, is generally Stoica. The high value of the degree of the most connected nodes results in high coverage of the positive information. Consequently, most paths of diffusion of negative information are blocked. The maximum degree in *Facebook*, *Twitter*, *NetASTRO* and *Enron* is significantly above average, with 1045, 1205, 504 and 1383 respectively. Additionally, in these data sets, the average degree is quite large, as a result, information spreads through numerous paths. Hence, reducing the network to maximum probability paths, like the CMIA-O method, causes sub-optimal effectiveness, as in reality, information diffuses throughout many other paths. However, in the networks, Net-Hept and Net-Phy the best performing methods are CMIA-O and Maximin-CMIA-O. These methods benefit from the low average degree since in this case the number of paths through with information diffuses is reduced significantly. Blocking the maximum probability paths results in the blocking of most propagation paths. Under these networks, degree and Stoica, do not perform as well as in other graphs, due to this characteristic. The low value of the highest degree as well as the low average degree of these two collaboration networks, causes the selected S_P to be less powerful than in other networks. Reducing the number of paths of negative information diffusion that they successfully block.

Despite the clear impact of the network characteristics in the considered benchmarks' effectiveness, the proposed method FWRRS seems to perform best overall networks. The introduction of a fairness objective does not negatively impact the number of saved nodes. In fact for a sufficiently large budget k, it works as a catalyst for effectiveness. Enforcing the true information to reach all communities, which otherwise would have been affected by the spread of the fake news, guarantees that the intervention reaches communities that otherwise would have been neglected. This way we overcome the overlapping influence spheres that the nodes being selected by the other methods have, and reach minority communities that are otherwise hard to reach from these central nodes.

For low budget values, the performance is similar or superior to the next best performing method. It is only in Figure 5.6(a) and Figure 5.7(a) that we observe some cost of fairness, as *degree* outperforms FWRRS for low values of k. However, the cost of such fairness is very low, the percentage of saved nodes is only 2% lower in the worst case. This cost comes from trying to reach all communities in need with a low number of truth spreaders. It can be observed that this initial effort becomes valuable for larger k values, where the FWRRS achieves a better performance than all other benchmarks.

In summary, we can observe the power and need for a fairness-aware method such as the proposed FWRRS. Not only is this method the fairest overall of the evaluated methods, but it also consistently achieves the highest percentage of saved nodes.

5.1.3.2 Evaluation under the WC model with highest degree negative seed

Next the impact of the seed, on the performance of the methods, was measured. In the previous evaluations, the seed was chosen at random so that the results would be extendable to all scenarios. However, by choosing the nodes at random the expected population of S_N over the communities is proportional to their size. To cover other alternative scenarios, we consider the case in which S_N is chosen to be the set of nodes with the highest degree.



(a) Evaluation on Number of saved nodes, with varying k values

with (b) Evaluation on maxi-min value, with varying k values

Figure 5.8: Experimental analysis in the Facebook dataset under WC model with the highest degree S_N



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.9: Experimental analysis in the Twitter dataset under WC model with the highest degree S_N



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.10: Experimental analysis in the NetPHY dataset under WC model with the highest degree S_N



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying varying k values $$\rm k$$ values

Figure 5.11: Experimental analysis in the NetHEPT dataset under WC model with the highest degree ${\cal S}_N$



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying varying k values k values

Figure 5.12: Experimental analysis in the NetASTRO dataset under the WC model with the highest degree S_N



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying varying k values $$\rm k$$ values

Figure 5.13: Experimental analysis in the Enron-Email dataset under the WC model with the highest degree ${\cal S}_N$

In this scenario, in most data sets, the majority of the negative information spreaders belong to one large community, causing the majority of the infections within this group. Additionally, due to the influential power of the high-degree nodes, the number of infections significantly increases as opposed to the previous evaluation, where S_N was chosen at random. This causes more minority communities to be slightly affected, whereas in the previous scenario a smaller number of minority communities would be more moderately affected. It is expected that methods will be less fair in this case, since more minority communities which are likely to be neglected are being infected. The effort to reach all these small communities might come as a cost since a significant part of infections do not take place within these communities. The findings of the evaluations are presented in Figure 5.8, Figure 5.9, Figure 5.10, Figure 5.11, Figure 5.12 and Figure 5.13.

Under this scenario, indeed it can be observed that methods become less fair. More often, the proposed FWRRS is the only method to achieve a maximin value above zero. Due to the characteristics of the seed, the portion of infections taking place in minority communities is significantly reduced. The majority of the negative seeds belong to the biggest communities, causing the majority of the infections in these groups. This leads methods to choose nodes with high saving capacity within these communities, neglecting the smaller groups. Additionally heuristic methods, such as degree, will choose less powerful nodes, as the highest degree nodes already belong to S_N . As a result, these nodes are less likely to reach smaller communities. The FWRRS specifically tries to avoid this scenario, and thus, achieves consistently the highest maximin value in all datasets. Despite this, in some datasets, as can be observed in Figures 5.9(b) and 5.13(b), even the FWRRS struggles to achieve a high value of maximin. The current implementation tries to avoid the scenario in which increasing maximin would cause a large cost. In addition to this, in this scenario in which negative information diffuses strongly over the network due to the nature of S_N , saving nodes is quite expensive. Numerous paths need to be blocked to guarantee to save a node from being reached by the fake news. Thus the reduction in the percentage of saved nodes as posed to the random negative seed experiments. As a result, saving nodes from all the small communities is resource expensive and might not increase significantly the percentage of total saved nodes, as the majority of infections take place elsewhere.

Regarding the percentage of saved nodes, the better performance of the FWRRS is more visible in this scenario. In all datasets evaluated, the proposed method saved a larger percentage of nodes from infection in the network. Since negative information diffuses in the network throughout numerous paths, the WRR structure allows us to identify and block most of them. The number of WRR trees to be considered is larger in this scenario, as the probability of a tree containing a negative seed node is higher, allowing us to cover a significant amount of infection paths. Blocking maximum probability paths like CMIA-O and Maximin-CMIA-O, is then not very effective in most cases, as information can diffuse through many other paths. Further, as previously mentioned heuristic methods like degree or Stoica, tend to choose less powerful nodes, as the ones with the highest degree already believe the fake news.

In summary, we can observe from this experiments, that even in a scenario in which the negative seed is very powerful, as it aims at maximizing the spread of the negative information, the FWRRS performs best in terms of fairness and percentage of saved nodes. In this case, fairness is more difficult to be achieved by other methods, highlighting the need for fairness-aware methods for truth campaigns with a significant impact on society, as the unfairness of these other methods grows with the number of infections. Additionally, the value of the FWRRS methods in terms of saved nodes is demonstrated, as its performance, compared to other methods, improves as the number of infections grows.

5.1.3.3 Evaluation under alternative cascade models

To be able to observe the differences between the methods, a value of p not too large had to be chosen. Further, as it was observed in the evaluation performed in Appendix A, with a large value of p = 0.1 most methods do not scale well to large diffusion cascades. But such a value could not be too small, this would result in the number of infections being low and all methods succeeding in saving a large portion of the graph by only including some infected nodes in the positive seed. The value of p = 0.05 was then established for this evaluation, this way a large number of nodes do get infected, and all benchmarks could be included in the evaluations, as they would terminate within the limited running time of 24 hours.

This evaluation was performed in the NetPHY dataset, which is a network with a relatively large size. The evaluation under this network aims at offering an understanding of the impact the probabilities have on the blocking power of the methods, as well as their fairness. The conclusions drawn in this section shall be extendable to all other datasets. Further, the negative seed was then chosen at random for this evaluation.



(a) Evaluation on Number of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.14: Experimental analysis in the NetPHY dataset under the Uniform model

Under this alternative propagation model, it can be observed the difference in performance between two groups of methods. BIOG, Maximin-CMIA-O and Stoica seem to perform worse than all remaining methods. We observe that those methods that succeed in spreading the truth the most in the network, perform all very similarly good. CMIA-O, spreads the truth through maximum probability paths, *degree* spreads it to a significant number of neighbours and TiB-Solver spreads it to those nodes that could spread the negative information the most, thus, those that can spread the positive one the most too. Last our method FWRRS spreads the truth over those paths that the negative influence might reach, thus also succeeding in spreading the truth powerfully, as the negative influence is expected to reach a vast majority of the graph in this model.

The superior maximin value achieved with our method can also be observed in this scenario. Even though we observe how other methods are fairer under this scenario, due to the ease of access to minority communities, the FWRRS method still achieves the best overall fairness value.

The proportion of the graph that gets infected under this experiment is quite large since it consists of practically the complete population. Thus, increasing the number of nodes directly connected to S_N will not change the considered scenario. Thus, the conclusions drawn in this section shall also be applicable when the negative seed consists of those nodes with the highest degree.

5.1.3.4 Evaluation under different community detection methods

The detection of network communities is not our focus in this work, so far these were computed using the Louvain algorithm. Such an algorithm consists on the state of the art method for community detection, as proven in the evaluation performed by Mothe [45]. This method offers high effectiveness, as it detects the communities with the highest modularity, which measures the density of connections within a community. Groups with a high modularity value have many interconnections within a community and a few pointing outwards to other communities. This approach also offers high efficiency, as it has a running time of $O(n \log(n))$. The scalability and effectiveness of this algorithm make it the go-to choice in community detection for social networks. In this section, the impact of the detection of the communities on the fairness and effectiveness of the methods is evaluated.

The selection of methods aims at covering all different types of approaches utilized for community detection. These are the following:

- Louvain: This algorithm was introduced by Blondel [12]. It initially finds small-sized communities by optimizing modularity locally and then aggregates such nodes into the communities iteratively until the modularity is maximized. In each iteration, for each node $v \in V$, it considers its neighbouring nodes and evaluates the gain in the modularity of removing it from the current community they belong to and placing it in the community of v.
- Greedy modularity: this approach was presented by Clauset [19]. The method begins by setting each node to belong to a different community. It then iteratively chooses the pair of nodes that when set to belong to the same community, increases modularity the most and joins them. This is repeated until modularity can no further increase.
- Walktrap: this other community discovery method, introduced by Pons [57], is based on random walks, in which distance between nodes is measured through random walks in the network. This algorithm has an $O(mn^2)$ running time, where m is the number of edges and n is the number of nodes in the graph. The fundamental premise of this approach is that random walks tend to get trapped in densely connected regions, corresponding in the network to communities. Communities are then aggregated in a bottom-up manner using these random walks.
- Infomap: was first introduces by Rosvall and Bergstom [61]. The procedure followed by this method is very similar to that of Louvain, with the difference that the objective function is the so-called map equation instead of modularity. This map equation goal is to use the community partitions of the graph as a Huffman code [33] that compresses the information about random walks in the graph. The map equation code structure is designed to compress the descriptive length of the random walk when this one takes place for extended periods within a certain region of the network.
- Label propagation: unlike all other community detection mechanisms, label propagation does not optimize any given objective function and requires no prior information about the network structure. Such a method was presented by Raghavan [58]. Initially, each node carries a label that represents the community to which they belong. Throughout the iteration within the network, each node will update its label according to the labels of its neighbour nodes. Choosing for the next iteration the label with the highest frequency within its neighbours. The process repeats until no further label changes are made.

To measure the impact of the use of the different community detection methods, the methods' performance in terms of percentage of saved nodes as well as maximin value with the detected partitioning of the network will be evaluated. The network chosen for this evaluation is NetPHY. It is a network with a relatively large size, as well as having a large number of communities, as detected by Louvain in the previous sections. The large number of communities present in this network gives room for the community detection methods to differ. Achieving then, the desired environment to measure their impact. The difference in the number of communities and modularity are summarized in Table 5.2. For this evaluation, the negative seed consists of 50 nodes at random and the diffusion probabilities are set according to the WC model. The budget value in these experiments is fixed to the value of 100. To be able to observe the impact on the fairness of the methods, a large budget value was chosen. In fact, in Figure 5.14(b), fairness in other methods other than the proposed *FWRRS* is only observed for such large values of k.

Community detection method	Number of communities	Modularity value
Louvain	3957	0.9253
Greedy modularity	4085	0.8717
Label propagation	6159	0.8584
Infomap	5430	0.8490
Walktrap	6491	0.8438

Table 5.2: Performance of the different community detection algorithms

The superiority of the Louvain method is made clear in Table 5.2, as it achieves the largest value of modularity by detecting the least number of communities within the network. The remaining methods, perform well as the modularity value remains high, however, this value differs between the methods. Our findings on the impact of less accurate communities are presented in Figure 5.15 and Figure 5.16



Figure 5.15: Percentage of saved nodes of the methods under different community detection algorithms in NetPHY under the WC model, with k=100



Figure 5.16: Maximin value of the methods under different community detection algorithms in NetPHY under the WC model, with k=100

In Figure 5.15 it can be observed how the percentage of nodes saved from infection of each method is barely altered by the change in the community detection method. Slight variations are observed, for example, the percentage of saved nodes of the Maximin CMIA-O method drops 1%, compared to the Louvain algorithm, under the greedy modularity algorithm and increases by 1% in the label propagation case. These variations do not necessarily come from the change in community detection method, they could be inaccuracies in the measurements, caused by the limit of 1000 simulations being used, especially given the small magnitude of this variation. Such variations are also observed in methods that do not take as input the communities, and thus, shall not be affected by them. For example, TIB-Solver, which has an increase of close to 1% under the Infomap algorithm compared to all other algorithms.

In Figure 5.16 a similar scenario is observed, the methods seem to achieve similar values of maximin across the different community detection methods. The variations, in this case, are a bit larger, with a maximum difference across algorithms of 13%. Some of this variation, as stated earlier can be associated with the inaccuracies caused by the simulations. This variation is then more prominent when looking at smaller portions of the network, namely a community. Additionally, the drop in modularity introduced by other algorithms might cause information within a community to diffuse less powerfully, possibly causing the drop from 42% maximin value in Louvain or greedy modularity for the *FWRRS* method to 29% in the Walktrap one.

Overall we can conclude that the community detection method used does not seem to have a clear impact on the performance of the methods. In most cases, the values of the methods barely differ across algorithms and the FWRRS remains the best performing in terms of saved nodes and fairness under all scenarios, thus it can be applied in combination with any desired algorithm.

5.2 Synthetic datasets experiments

5.2.1 Synthetic network model

The methods will all be evaluated under various synthetic networks with different characteristics. The proposed model to generate such networks will be extending the homophily preferential attachment model. The well-known model of preferential attachment was proposed by Barabási et. al [4], who also proposed a weighted version, the fitness model, with Bianconi [11]. Next Karimi [34] extended the fitness model to include homophily as a weighting parameter in the model. Homophily measures the tendency of the user to connect to others similar to them, creating thus communities. The proposed model in this section further extends this model to ensure the characteristics of real social networks are met.

Social networks are known for having the following three properties as presented by Sallaberry et. al [62]:

• Small world property. As defined by Watts et.al [74] a small world network is one with high clustering coefficient and an average distance between nodes that scales proportionally to the logarithm of the number of nodes. The most well-known manifestation of the concept of short average path length is the "six degrees of separation" presented by social psychologist Stanley Milgram, who concluded that most pairs of people in the United States had a path of acquaintances with a typical length of about six between them. Thus, in the constructed network G(V, E) the average shortest path between any pair of nodes A and B should be small. The average shortest path of G is denoted as L_q and is computed using Equation 5.1.

$$L_g = \sum_{A,B\in V} \frac{d(A,B)}{|V|*(|V|-1)}$$
(5.1)

where d(A, B) is the shortest path from A to B.

The other concept is the average clustering coefficient, C_g , which is based on the idea that a friend of your friend is likely to also be your friend. Mathematically, it is defined as in Equation 5.2

$$C_g = \frac{1}{|V|} \sum_{v \in V} c_v$$
 and $c_v = \frac{2 * T(v)}{deg(v) * (deg(v) - 1)}$ (5.2)

where T(v) is the number of pairs of neighbours of v that are also neighbours.

A network G with n nodes and m edges is a small-world network if it has a similar path length but a greater clustering coefficient than a random graph with the same n and m. Formally, let L_g define the average shortest path length of G, and L_{rand} that of the random graph. Further, Let C_g and C_{rand} denote the clustering coefficient of the two graphs. Then, for G to be a small-world network, $L_g \geq L_{rand}$ and $C_g \gg C_{rand}$ must hold.

- Scale free property is met by that network with a very high degree distribution, meaning a small number of nodes have a large degree and many nodes are connected to a small number of nodes. The degree distribution in most networks is believed to follow a Poisson distribution, however real-world networks have severely skewed degree distributions. Thus, scale-free networks, are those with a degree distribution that follows a power-law distribution, or at least asymptotically.
- Community structure property is defined as the natural division of the nodes into densely connected subgroups such that inter-groups connections (between different groups) are sparse, while intra-groups connections (within the same subgroup) are dense. In this work we define the communities to be a disjoint partitioning of the network. This means that every node in V belongs to a community and only one community, an example of a disjoint community structure is depicted in Figure 5.17.



Figure 5.17: Example of disjoint community structure

To evaluate the strength of the community structure in a network the assortativity coefficient is commonly utilized. It measures the similarity of connections in the graph concerning the community they belong to, also known as homophily. Mathematically, this is defined as in Equation 5.3, where M is the mixing matrix. The mixing matrix is a two-dimensional crossclassification of edges based on the community of the sender and the receiver, meaning that M_{ij} denotes the proportion of edges from a node in community *i* to a node in community *j*.

$$H = \frac{\sum_{i} M_{ii} - \sum_{i} a_{i} b_{i}}{1 - \sum_{i} a_{i} b_{i}} \quad , \quad a_{i} = \sum_{j} M_{ij} \quad \text{and} \quad b_{i} = \sum_{j} M_{ji} \tag{5.3}$$

However, the assortativity coefficient does not consider the the overall structure of the graph, as it reduces the entire graph into a matrix, losing much of the intricate complexity of the network [50]. Thus, we will be using the average Weighted External-Internal Index (WEI index) as defined in the book "Introduction to mathematical sociology" [20]. This notion is defined in Equation 5.4

$$avg_{-}WEI = \frac{1}{|V|} \sum_{vinV} \frac{\frac{e_n}{w_n} - i_n}{\frac{e_n}{w_n} + i_n}$$
(5.4)

Let e_n represent the number of inter-edges of node n and i_n the number of intra-edges. Also, w_n is set to be equal to the ratio of the number of nodes outside the community n belongs to and the number of nodes within this community. The inclusion of such weight allows us to balance the number of external (inter) edges with the proportion of external nodes. For example, let G consist of 200 nodes, where 180 belong to community one and the remaining 20 to community two. Then consider two nodes n_1 and n_2 , which belong to communities one and two respectively and are both connected to a node in each of the communities. If the weight of the nodes was to be 1 in all cases, the WEI value of both nodes would be 0. However, common sense suggests that n_2 exhibits a greater amount of homophily since there are few other nodes in community two that it could connect to. This is corrected with the inclusion of w_n , in this example the value of $w_{n_1} = \frac{20}{180} = 0.11$ and $w_{n_2} = \frac{180}{20} = 9$. The appropriate values of WEI are then 0.8 and -0.8 for the nodes n_1 and n_2 respectively, observing that node n_2 has strong homophily, as the WEI value is close to -1 while n_1 has a strong heterophily value, as the WEI value is close to 1.

Under this definition a strong homophilic network would have an average WEI value of -1 while networks that do not demonstrate homophily tend to have a value of 0.

The Homophily BA model [34], previously introduced, meets the scale-free and community structure properties, however, the clustering coefficients are extremely low. To be able to guarantee higher values if desired, the proposed model includes a clustering parameter p_t , to control the level at which friends of a node will be friends. Another contribution is that it includes a parameter p_{PA} , that controls the probability with which a node chooses to connect based on the preferential attachment or imposingly at random. Accordingly, the High Clustering Homophily Barabási-Albert (HICH-BA) model, is presented in Algorithm 9, which is found in Appendix B.

5.2.2 Experiments and evaluation

Using Algorithm 9 we will perform a sensitivity analysis of the methods. This analysis aims to determine how the approaches' performance and fairness are affected based on changes in the network characteristics. For this, we will evaluate the methods under varying homophily values as well as varying p_{PA} values, which resemble the proportion of preferential attachment edges. The average degree will also be altered, as this parameter showed importance in the previous chapter evaluation. This parameter is controlled by increasing the value of m. The characteristics of the five generated synthetic networks are summarized in Table 5.3. Additionally, the degree distributions of these networks are visualized in Figure 5.18, where the scale-free property of the networks can be observed. For all generated synthetic datasets the population of the communities r is set to be [0.54, 0.3, 0.15, 0.005, 0.005]. This consists of 5 communities of varying sizes, where we can find very small minority communities and 3 larger groups of different sizes.

Synthetic	n	m	h	p_t	p_{PA}	Average	WEI	Assortativity	Clustering
dataset						degree		coefficient	coefficient
SG1	10000	200000	0.9	0.9	0.9	20.398	-0.909	0.714	0.626
SG2	10000	40000	0.9	0.9	0.9	5.3394	-0.982	0.955	0.351
SG3	10000	100000	0.9	0.9	0.1	9.797	-0.886	0.760	0.379
SG4	10000	100000	0.9	0.2	0.9	9.659	-0.875	0.788	0.177
SG5	10000	100000	0.2	0.9	0.9	8.968	-0.579	0.356	0.388

Table 5.3: Synthetic datasets characteristics

The first synthetic network **SG1**, consists of a network with a high clustering coefficient of 0.626. The average degree is also large, namely 20.398 and the WE index has a value of -0.909, showing strong homophily. This network is supposed to represent all the favourable conditions for the *degree* or *Stoica* method. As it was observed in the real datasets experiments, these methods benefit from a high average degree, since due to the power-law distribution the degree follows, this translates to the nodes they choose having a very large number of neighbours. Further, since most connections are established according to preferential attachment, high-degree nodes will be very powerful, as other nodes will tend to connect to them. Last a high clustering coefficient, translated to information propagating strongly in the local area of selected nodes since this area will be very interconnected. Since nodes can spread information to many other nodes, *CMIA-O* underperforms, as information spreads through many other paths than maximum probability paths. The results of this evaluation in this dataset are visualized in Figure 5.19.



Figure 5.18: Degree distribution of the synthetic datasets

Indeed in Figure 5.19(a) we observe that after the FWRRS method, the best-performing methods are *degree* and *Stoica*, whereas CMIA-O is less powerful. Due to the large average degree, the positive information propagates throughout many paths in the network, thus the benefits of the intervention tend to reach all communities, except CMIA-O and TIB-Solver which underperform significantly. Since they do not save many nodes, this is also seen for the maximin value, which is quite low. However, the proposed method FWRRS still achieves the largest value, being the fairest and best-performing method as observed in Figure 5.19(b). Further, we can observe that despite CMIA-O performing so poorly in this case, its fair alternative performs quite good. Spreading the information fairly over the communities seems to cover a larger portion of the propagation paths, and leads to better blocking the negative information, showing the value of fairness even for poorly performing methods.



(a) Evaluation on percentage of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.19: Experimental analysis in the SG1 dataset under WC model with random S_N

The SG2 network, on the other hand, aims to favour the conditions where *CMIA-O* and *Maximin-CMIA-O* perform best. This is the case when information does not propagate through many paths. This can be achieved by setting the average degree to a low value. The remaining parameters of the model are left unchanged, to explore the exact impact of the average degree alone. However, this reduction in average degree has some indirect impact on the clustering coefficient and homophily. As a result of the reduction in the number of edges being added to the network, it is more difficult to keep the clustering coefficient high. Every time a new node arrives in the network numerous new open triangles are created, however, a lower portion of them is closed as fewer edges for such purpose are generated. The homophily, on the other hand, increases. We drastically reduce the number of inter-community edges that are added from the majority community, since the majority of the added edges will be for the such community, thus homophily is positively affected by this.

In Figure 5.20(a) we observe the change in performance of the methods. In this case, CMIA-O performs significantly better as opposed to the previous synthetic network. The negative information propagated throughout fewer paths, and thus, blocking the maximum probability paths is very effective. Further, *degree* and *Stoica*, perform well for low budgets, but for larger budgets, they become outperformed by other methods. In this network, those nodes with the highest degree do not reach as many nodes as in the previous network, as their degree is lower.



(a) Evaluation on percentage of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.20: Experimental analysis in the SG2 dataset under WC model with random S_N

The change in performance of the methods is also seen in Figure 5.20(b). We can observe how the CMIA-O achieves a larger maximin value than in the previous case, as well as how the *degree* and *Stoica* values are lower. The proposed method FWRRS shows to be unaltered by this change in the network characteristics. It consistently achieves the highest number of saved nodes as well as maximin value, adapting its selection of seeds to this change in average degree.

Next, we evaluate the impact the randomness of the graph has on the way methods perform. For this, we generate **SG3**. In this network, 90% of the edges were established at random, while still maintaining the homophilic preference. This means that nodes would still connect with a probability of 0.9 to nodes in its community and otherwise to a node outside this group. In Figure 5.18(c), we observe that the degree distribution is more convex than when most of these connections were established using the preferential attachment. Those nodes with high-degree are less often chosen to be connected to, and thus other nodes become more powerful. As a result, the distribution slope is less steep. This leads us to believe that in real-datasets randomness is present, as this curve feature of the distribution was observed in Figure 5.1.



(a) Evaluation on percentage of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.21: Experimental analysis in the SG3 dataset under WC model with random S_N

In Figure 5.21(a), we observe that most methods tend to perform similarly, especially for larger values of k. The randomness of the graph allows for any node to possibly reach another user with

their information. However, FWRRS still seems to perform best followed by degree and Stoica, and eventually Maximin-CMIA-O. Because of how the network is constructed, nodes that join the graph in an early iteration will be chosen more times by other nodes for connection, since only those nodes currently present in the graph can connect. As a result, those nodes will tend to have a higher degree, and additionally will be connected to random portions of the graph, leading to their information reaching a large portion of the network.

In Figure 5.21(b) we can observe that all methods tend to be fairer. The randomness of the graph encourages information to spread randomly and reach all communities eventually. Regardless, those methods that explicitly consider the fairness objective, FWRRS and Maximin-CMIA-O, achieve the highest fairness values.

The fourth synthetic network **SG4** is a graph in which the value of p_t is significantly smaller. With this network, we want to evaluate the impact the clustering coefficient has on the methods' performance. In all previous graphs, we maintained a relatively high clustering coefficient, which is characteristic of social networks. However, in this case, the clustering coefficient is only 0.177. Because now a significantly larger portion of edges are established according to pure preferential attachment, the degree distribution seems to be steeper, as visualized in Figure 5.18(d). Now, fewer nodes with a medium degree can be found, as nodes tend to be connected to the same subset of highly connected nodes.



(a) Evaluation on percentage of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values ues

Figure 5.22: Experimental analysis in the SG4 dataset under WC model with random S_N

In Figure 5.22(a), we observe the power of *degree* and *Stoica*, which intuitively comes from the strength of preferential attachment in this network. Since nodes tend to connect to highly connected nodes, which are those being selected by these two methods, the truth is highly spread. As a result, they reach many nodes and avoid many infections. Despite this power of high degree nodes, FWRRS still performs better than *degree* and *Stoica*, achieving a portion of saved nodes close to 10% higher.

Something that can be noticed in Figure 5.22(b) is the difference in fairness between FWRRS and all remaining methods. The low clustering coefficient of the graph causes information within communities to spread less powerfully. This means, that the little positive information that was reaching those least helped communities now reaches lesser nodes in this group. To achieve a higher portion of saved nodes in these communities, more than just reaching the community is required. Since FWRRS considers the maximin value explicitly, we then guarantee that the least help received by all communities is maximized, achieving a significantly larger value of maximin.

Last we generated **SG5**, consisting of a network with low homophily. In this graph, the WEI value is only -0.579 and more dramatically the assortativity coefficient has a low value of 0.356.



(a) Evaluation on percentage of saved nodes, with (b) Evaluation on maxi-min value, with varying k valvarying k values $$\rm ues$$

Figure 5.23: Experimental analysis in the SG5 dataset under WC model with random S_N

In Figure 5.23, we perceive a very similar performance to the one observed in Figure 5.22. Because now nodes prefer to connect to others outside their community, the clustering coefficient within the communities is quite low. This leads to methods, being less fair than in previous synthetic networks. Further, since the number of inter-edges increases, the number of paths between communities also increases. CMIA-O as a result fails to contain the spread of fake news, which negatively affects both its fairness and effectiveness. Explicit consideration of the fairness objective is then required, as FWRRS and Maximin-CMIA-O achieve the highest fairness and effectiveness out al all methods.

In summary, in this evaluation we were able to observe the impact the network characteristic have on the fairness and performance of the methods. High average degree and reach of the highest degree nodes are crucial aspects for methods such as *degree* and *Stoica*, to achieve good results. on the other hand for the *CMIA-O* method, the opposite is required, as it benefits from fake news spreading through a small number of paths. We can also conclude that the *TIB-Solver* method, does not perform as well as claimed, in most cases. It is only when information spreads strongly through the network, like under the uniform model described in Figure 5.14. Overall we show the power of the proposed method *FWRRS*, as it consistently achieves a larger percentage of saved nodes. Further, this method achieves such a successful outcome for the intervention, while being the fairest.

Chapter 6 Conclusions

As described in Section 1.2, this research aimed to develop an algorithm for solving the IBM problem in a fairness-aware manner. For defining fairness in this area, a novel problem namely the FIBM problem, which relies on maximin to measure the fairness of the methods. The main challenges of this goal were to define fairness in the area of information blocking as well as overcome the expected cost that fairness has on the number of saved nodes. In this chapter, we summarize the findings of this work, reflect on its limitations and explore possible future research.

6.1 Main contributions

In this thesis a novel problem in the area of information diffusion was introduced, the FIBM problem, which is defined in Section 1.2. This is a fairness-aware variant of the well-known problem of influence blocking maximization of a rumour with the use of debunkers. These debunkers will be spreading the true information to counteract the spread of fake news. The novelty of the type of research performed in this thesis, as to the best of our knowledge, no work that addresses fairness in IBM has been proposed, shall motivate the research in this area, showing its benefits beyond moral aspects.

Relating to the research questions for this work, we evaluated the fairness of numerous existing methods, covering different types of proposed approaches, like heuristic or simulations based. Given that, to the best of our knowledge, no previous work in fairness-aware IBM has been performed, we also proposed some fairness benchmarks for the performed evaluation. For this, two extensions to existing methods were proposed in Sections 4.2 and 4.3. On the one hand, the *Stoica* method, a fairness-aware IM method, was extended to the IBM problem. On the other hand, the *Maximin-CMIA-O* method, an IBM method, was adapted to have as objective maximizing fairness. The resulting pool of benchmarks then established a complete evaluation setting for analysing the fairness and effectiveness of the proposed FWRRS method.

The experimental analysis under the real-life social networks, presented in Section 5.1, showed the need for fairness-aware methods for the IBM problem since existing methods carry the inequalities in the network structure into the outcome. This leads to entire communities being left outside of the benefits of the truth campaign intervention. It was also observed that the performance of these methods was not consistent and strongly depended on the characteristics of the network. Methods such as *degree* or *Stoica* would benefit from a high value of average and upper bound of degree as well as a high clustering coefficient, for example, the experiment performed in Figure 5.3. As a result the selected nodes, those with the highest degree, would be very powerful, as they would reach a significant portion of the graph. On graphs where the clustering coefficient and average degree were lower, we observe that methods such as CMIA - O or BIOG perform better, like in Figure 5.4, since the negative information propagates through fewer paths. Blocking max-

imum probability paths shows to be a powerful strategy under these characteristics. However, the proposed method FWRRS remains unaltered to the characteristics of the graph and is observed to perform best overall, in both fairness and percentage of saved nodes. To further evaluate the impact of the network structure on the performance of the methods, we also generated synthetic networks with controlled characteristics.

The need for a model to generate synthetic networks that better resemble the characteristics of real-life social networks were also presented. A new model HICH-BA was introduced in Appendix B, which achieves a higher clustering coefficient, characteristic of social networks, than most existing models. Additionally, it allows us to define the strength of the community structure present in the graph. The methods were also evaluated under these controlled networks, showing under which circumstances each model performs best. This evaluation also confirmed the instability of the performance of the methods, and showed the value of the proposed method FWRRS, as it consistently performs best, disregarding the network characteristics.

In this evaluation the proposed method FWRRS, not only performs as the state of the art concerning fairness, but it also in all cases performs best in terms of saved nodes. Meaning it outperforms the current state-of-the-art methods for IBM under the ICM propagation model. This shows, that full filling this need for fairness does not come at a cost, in fact in most cases it is beneficial for the effectiveness of our mitigation campaign. Thus, in this thesis, we solve the main research question of this work, how to achieve the desired trade-off between fairness and effectiveness. Enforcing fairness like in the FWRRS method was shown to be a catalyst for effectiveness, by exploiting the community structure of the social networks.

6.2 Limitations and future work

This work aimed at showing the possibilities of including fairness in the process of debunkers selection for the IBM problem. The novel research performed in this work counts with some limitations and shall motivate future research. Some ideas on further research directions are presented in this section.

Adaptation to alternative propagation models

To focus on the impact such fairness would have on the performance of the methods, we chose to simulate information diffusion utilizing one of the most well-known propagation models, the ICM. Deciding on this model, allowed us to include most research in the areas of IBM, as these proposed methods assume information diffused according to a cascade. However, many other extensions of the ICM model [65], as well as completely different propagation models [55, 78, 6], can be utilized.

While the nature of the independent cascade model favours diversity in maximizing the number of saved nodes, another model may not do the same. An interesting future step would be to study how a method such as FWRRS would perform under models such as the linear threshold model or the SIR model. Adaptations to the way the WRR trees are constructed would need to be made, to resemble how information now diffuses under these propagation models.

Scalability of the method

The focus of this study was on the fairness and effectiveness of the method, however, its running time was also considered. The FWRRS is shown to scale better than most methods currently in use, as well as being significantly faster than the greedy algorithm that approximates the optimal seed. Especially in cases where a large portion of the network is to be infected, the method is shown to perform faster opposed to most other benchmarks, which failed to complete within 24

hours.

The method was evaluated in numerous datasets with varying sizes and characteristics. An indepth evaluation of the methods' running times would be an interesting next step, to understanding how the characteristics of the graph impact their efficiency. Another interesting next step would be to evaluate its performance in a million-scale network, which is the common size of social networks where the interventions would take place. We anticipate the good performance of the method would only be enhanced in this case. Since as we observed in the experiments, as the number of infections grows, the performance of the FWRRS method improves. As we already showed, other methods do not scale well to an increased number of nodes considered, thus the value of this proposed method is also good in terms of running time, as it consists of a more scalable alternative for solving this problem. Additionally, the design of the proposed method allows for parallelization, which has not been considered yet, in steps such as generating the WFRs, the WRRs or in iterating over such tree structures, which will further improve the running time of the method.

Balancing fairness and effectiveness differently

The current implementation of the FIBM aims at maximizing the blocking power of such a truth campaign such that the outcome of this campaign is fair, measured in terms of maximin. Additionally, the proposed methods show to successfully achieve this. The fairness is maximized and such a goal does not cause a reduction in blocking power, meaning that in most cases the number of saved nodes is also being maximized. However alternative desired scenarios could be considered in the evaluation of the other methods. The condition 1.2b, could be removed and the objective could be altered to be a balancing function of fairness and effectiveness, as presented in Equation 6.1

$$\underset{S_P}{\operatorname{maximize}} \quad \alpha * \mathbb{M}(G, S_N, S_P) + (1 - \alpha) * \mathbb{B}(S_P, G, S_N, \emptyset)$$

$$(6.1)$$

With this new objective function, different levels of importance of both variables could be considered. For example, in the scenario in which fairness is our only goal, as we want to achieve the highest value of maximin possible, despite its effect on the performance, then α would be set to 1. It would be of interest to study how the value of α affects the number of saved nodes in other methods, to further study how to achieve the trade-off between fairness and effectiveness with them. Further parameters would have to be inputted into the algorithms.

Considering overlapping communities

In this work, communities were considered and detected to be non-overlapping. This means that it was approached as a partitioning of the population, i.e. a node must belong to one and only one community. However, it is intuitive that elements can participate simultaneously in several groups. For example, a user can be deeply connected to their family, while also be related to their sports team-mates and coworkers.

Many methods have been proposed to detect overlapping communities. Some proposed methods that could be considered are CFinder proposed by Palla et al. [51], Demon by Cocia et al. [21], COPRA proposed by Gregory [29] or Bigclam by Leskovec et al [41]. These detected communities could then be directly used as input for the proposed methods. Studying the importance of these overlapping vertices for the performance and fairness of the methods would be an interesting direction for future research.

Applicability to other problems

The method proposed in this research was specifically designed to solve the FIBM problem, however, it could be easily extended to solve many other problems in the area of information diffusion, to also introduce fairness-aware methods in these areas of research. Some examples are presented next.

- Temporal influence blocking maximization: consists of an extension of the classical IBM problem, with the inclusion of a deadline, and was introduced in the work of Song [70]. The number of saved nodes wants to be maximized within a limited time frame. The *FWRRS* could then be extended to this problem by setting the maximum depth of a WRR tree to be of such a deadline. No infections taking place after such time frame would then be considered, and we would strive to save as many nodes as possible that get infected before the predefined deadline.
- Targeted influence blocking maximization: this version of the IBM problem aims at maximizing the number of saved nodes within a selected set of nodes. This problem was introduced in the work of Zhu [83]. A simple change would be required for this other problem, namely the set S of nodes to consider would be the inputted target group. This way the *WRR* trees would resemble how these nodes could be infected and we would strive to save only these from such infection.
- Competitive influence maximization: for this problem the goal is to maximize the spread of certain information where multiple types of information are competing within a social network. This problem is presented by Bharathi [10]. The proposed method currently only considers those WRR trees in which a S_N node is present. The method aims at guaranteeing that the opponent information will intercept this negative one, saving the node from being infected. However, for this problem, we could consider all WRR trees and select the node that spreads the truth the most, instead of the one that saves the most nodes. The remaining fairness conditions would remain the same, but maximin would be in terms of the portion of nodes informed per community instead of the portion saved.
- Influence Blocking Maximization under uncertain sources: this problem was introduced in the work of Chen [15]. It considers the scenario in which the negative seed S_N is unknown, instead, their distribution is known. Adapting the method for this problem would require more extensive changes, however, the structure of WRR trees could be very beneficial for covering and considering all these cases of infection. We would have to keep track of all possible scenarios given by the negative seed distribution, to make our choices within the WRRs, which would have to be considered more than once. Since now S_N is unknown multiple paths within a single WRR might have to be covered.
- Budgeted influence blocking maximization: this alternative version of the IBM was studied by Erd [25], inspired by the work of Pham [52] who studied the budgeted version of the IM problem. The main concept is that each node has a cost associated with them, and thus we have to choose a positive seed S_P that maximizes the number of saved nodes, such that the commutative cost is within our budget. This problem could be solved by using the *FWRRS* method, by evaluating the worthiness of selecting a node, balancing its cost with the number of nodes it saves or the increase in maximin it causes. A formula for this evaluation would have to be designed, and then the method could be utilized.
- Rumor containment with user or edge blocking: The method could also be adapted for the alternative method to mitigate misinformation, namely blocking users, like in the work of Wu [76] or edges, like in the work of Yan [77], from sharing it when encountered. Instead of choosing the node present in most WRR, we would then choose the node or edge present in most paths from a S_N node to the root. These choices should then also consider maximin, as that is our objective.

Bibliography

- [1] https://aokmarketing.com/breaking-news-social-media-changed-way-consume-news. 1
- [2] https://www.nbcnews.com/news/us-news/russian-backed-election-content-reached-126million-americans-facebook-says-n815791.
- [3] http://www.arXiv.org. 31
- [4] Albert-László Barabási and Réka Albert. Emergence of Scaling in Random Networks. Science 286, pages 509–512, 1999. 44
- [5] Junaid Ali, Mahmoudreza Babaei, Abhijnan Chakraborty, Baharan Mirzasoleiman, Krishna Gummadi, and Adish Singla. On the Fairness of Time-Critical Influence Maximization in Social Networks. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 12
- [6] Roy M Anderson and Robert M May. Infectious diseases of humans: dynamics and control. Oxford university press, 1992. 54
- [7] Wissam Antoun, Fady Baly, Rim Achour, Amir Hussein, and Hazem Hajj. State of the Art Models for Fake News Detection Tasks. 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies, ICIoT 2020, (June 2021):519–524, 2020. 2
- [8] Niloofar Arazkhani, Mohammad Reza Meybodi, and Alireza Rezvanian. Influence Blocking Maximization in Social Network Using Centrality Measures. 2019 IEEE 5th Conference on Knowledge Based Engineering and Innovation, KBEI 2019, pages 492–497, 2019. 10, 11
- [9] Solon Barocas and Andrew D. Selbst. Big Data's Disparate Impact. SSRN Electronic Journal, 671:671–732, 2018.
- [10] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive Influence Maximization in Social Networks. International Workshop on Web and Internet Economics. Springer Berlin Heidelberg,, 11917 LNCS:306-311, 2007. 56
- [11] G. Bianconi and A. Barabási. Competition and multiscaling in evolving networks. EPL (Europhysics Letters), pages 54, 436, 2001. 44
- [12] 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):1–12, 2008. 42
- [13] Ulrik Brandes. The Journal of Mathematical A faster algorithm for betweenness centrality. Journal of Mathematical Sociology, (March 2012):37–41, 2010. 9
- [14] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. Proceedings of the 20th International Conference on World Wide Web, WWW 2011, pages 665–674, 2011. 6, 10, 11

- [15] Ling Chen, Yuliang Zhang, Yixin Chen, Bin Li, and Wei Liu. Negative influence blocking maximization with uncertain sources under the independent cascade model. *Information Sciences*, 564:343–367, 2021. 33, 56
- [16] Wei Chen. http://research.microsoft.com/enus/people/weic/graphdata.zip. 31
- [17] Wei Chen, Alex Collins, Rachel Cummings, Te Ke, Zhenming Liu, David Rincon, Xiaorui Sun, Yajun Wang, Wei Wei, and Yifei Yuan. Influence maximization in social networks when negative opinions may emerge and propagate. *Proceedings of the 11th SIAM International Conference on Data Mining, SDM 2011*, (January 2015):379–390, 2011. 31
- [18] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 199–207, 2009. 31
- [19] Aaron Clauset, M. E.J. Newman, and Cristopher Moore. Finding community structure in very large networks. *Physical Review E - Statistical Physics, Plasmas, Fluids, and Related Interdisciplinary Topics*, 70(6):6, 2004. 42
- [20] James S. Coleman. Introduction to mathematical sociology. Free Press of Glencoe, New York, 1964. 46
- [21] Michele Coscia, Giulio Rossetti, Fosca Giannotti, and Dino Pedreschi. Uncovering hierarchical and overlapping communities with a local-first approach. ACM Transactions on Knowledge Discovery from Data, 9(1), 2014. 55
- [22] Yushun Dong, Jing Ma, Chen Chen, and Jundong Li. Fairness in Graph Mining: A Survey. pages 1–20, 2022. 12
- [23] Sergey Edunov, Dionysios Logothetis, Cheng Wang, Avery Ching, and Maja Kabiljo. Generating synthetic social graphs with darwini. Proceedings - International Conference on Distributed Computing Systems, 2018-July:567–577, 2018. 4
- [24] Marwa El Halabi, Slobodan Mitrovic, Ashkan Norouzi-Fard, Jakab Tardos, and Jakub Tarnawski. Fairness in streaming submodular maximization: Algorithms and hardness. Advances in Neural Information Processing Systems, 2020-Decem(NeurIPS), 2020. 12
- [25] Fernando C. Erd, André L. Vignatti, and Murilo V.G.da Silva. The generalized influence blocking maximization problem. *Social Network Analysis and Mining*, 11(1):1–17, 2021. 9, 11, 33, 56
- [26] Giorgio Fagiolo. Clustering in complex directed networks. Physical Review E Statistical, Nonlinear, and Soft Matter Physics, 76(2):1–8, 2007. 9
- [27] Lidan Fan, Zaixin Lu, Weili Wu, Bhavani Thuraisingham, Huan Ma, and Yuanjun Bi. Least cost rumor blocking in social networks. *Proceedings - International Conference on Distributed Computing Systems*, pages 540–549, 2013. 31
- [28] Golnoosh Farnad, Behrouz Babaki, and Michel Gendreau. A Unifying Framework for Fairness-Aware Influence Maximization. The Web Conference 2020 - Companion of the World Wide Web Conference, WWW 2020, (1):714–722, 2020. 8
- [29] Steve Gregory. An algorithm to find overlapping community structure in networks. pages 91–102, 09 2007. 55
- [30] Xinran He, Guojie Song, Wei Chen, and Qingye Jiang. Influence blocking maximization in social networks under the competitive linear threshold model. *Proceedings of the 12th SIAM International Conference on Data Mining, SDM 2012*, (October):463–474, 2012. 10, 11, 31, 33

- [31] Jeffrey Herrmann, William Rand, Brandon Schein, and Neža Vodopivec. An agent-based model of urgent diffusion in social media. 18(2), 2013. 6
- [32] Adil Imad Eddine Hosni, Kan Li, and Sadique Ahmad. Minimizing rumor influence in multiplex online social networks based on human individual and social behaviors. *Information Sciences*, 512:1458–1480, 2020. 10, 11
- [33] David A Huffmant. A Method for the Construction of Minimum Redundancy Codes. Proceedings of the IRE, 40:1098–1101, 1952. 42
- [34] Fariba Karimi, Mathieu Génois, Claudia Wagner, Philipp Singer, and Markus Strohmaier. Homophily influences ranking of minorities in social networks. *Scientific Reports*, 8(1):1–12, 2018. 45, 47
- [35] David Kempe and Jon Kleinberg. Maximizing the spread of influence through a social network. Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 137–146, 2003. 12, 33
- [36] David Kempe, Jon Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. *Lecture Notes in Computer Science*, 3580:1127–1138, 2005. 6
- [37] Bryan Klimt and Yiming Yang. Introducing the Enron Corpus. Machine Learning, stitutep1:wwceasccaers2004168, 2004. 31
- [38] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over Time: Densification Laws, Shrinking Diameters and Possible Explanations. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), 2005. 31
- [39] Jure Leskovec and Andrej Krevl. SNAP Datasets: Stanford Large Network Dataset Collection, http://snap.stanford.edu/data, 2014. 30, 31
- [40] Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009. 31
- [41] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. Proceedings of the 19th International Conference on World Wide Web, WWW '10, pages 631–640, 2010. 55
- [42] Kuei Sheng Lin and Bi Ru Dai. BIOG: An Effective and efficient algorithm for influence blocking maximization in social networks, volume 1071. Springer Singapore, 2019. 10, 11
- [43] Jiaguo Lv, Bin Yang, Zhen Yang, and Wei Zhang. A community-based algorithm for influence blocking maximization in social networks. *Cluster Computing*, 22(s3):5587–5602, 2019. 9, 11
- [44] Julian Mcauley and Jure Leskovec. Learning to Discover Social Circles in Ego Networks. NIPS, pages 1–9, 2012. 30
- [45] Josiane Mothe, Karen Mkhitaryan, and Mariam Haroutunian. Community detection: Comparison of state of the art algorithms. International Scientific and Technical Conference on Computer Sciences and Information Technologies, 2018-March:125–129, 2017. 42
- [46] Hung T. Nguyen, My T. Thai, and Thang N. Dinh. Stop-and-Stare: Optimal sampling algorithms for viral marketing in billion-scale networks. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 26-June-20:695–710, 2016. 18, 33
- [47] Nam P. Nguyen, Guanhua Yan, and My T. Thai. Analysis of misinformation containment in online social networks. *Computer Networks*, 57(10):2133–2146, 2013. 31
- [48] Nam P. Nguyen, Guanhua Yan, and My T. Thai. Containment of Misinformation Spread in Online Social Networks. Computer Networks, 57(10):2133–2146, 2013. 10, 11
- [49] E. C. Nisbet, P. Beck, and R. Gunther. "Trump May Owe His 2016 Victory to 'Fake News,' New Study Suggests,". The Conversation, 2018. 1, 2
- [50] Shiva Oswal. The Popularity-Homophily Index: A new way to measure Homophily in Directed Graphs. pages 1–13, 2021. 46
- [51] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005. 55
- [52] Canh V. Pham, Hieu V. Duong, Huan X. Hoang, and My T. Thai. Competitive influence maximization within time and budget constraints in online social networks: An algorithmic approach. Applied Sciences (Switzerland), 9(11):1–29, 2019. 56
- [53] Canh V. Pham, Quat V. Phu, and Huan X. Hoang. Targeted Misinformation Blocking on Online Social Networks. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 10751 LNAI:107–116, 2018. 31
- [54] Canh V. Pham, Quat V. Phu, Huan X. Hoang, Jun Pei, and My T. Thai. Minimum budget for misinformation blocking in online social networks. *Journal of Combinatorial Optimization*, 38(4):1101–1127, 2019. 31
- [55] Dung V Pham, Hieu V Duong, Canh V Pham, Bui Q Bao, and Anh V Nguyen. Multiple topics misinformation blocking in online social networks. In 2019 11th International Conference on Knowledge and Systems Engineering (KSE), pages 1–6. IEEE, 2019. 54
- [56] Mahendra Piraveenan, Mikhail Prokopenko, and Liaquat Hossain. Percolation Centrality: Quantifying Graph-Theoretic Impact of Nodes during Percolation in Networks. *PLoS ONE*, 8(1), 2013. 9
- [57] Pascal Pons and Matthieu Latapy. Computing communities in large networks using random walks. Journal of Graph Algorithms and Applications, 10(2):191–218, 2006. 42
- [58] Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 76(3):1–12, 2007. 42
- [59] Aida Rahmattalabi, Shahin Jabbari, Himabindu Lakkaraju, Phebe Vayanos, Max Izenberg, Ryan Brown, Eric Rice, and Milind Tambe. Fair Influence Maximization: A Welfare Optimization Approach. 35th AAAI Conference on Artificial Intelligence, AAAI 2021, 13A:11630– 11638, 2021. 7
- [60] John Rawls. A Theory of Justice Harvard paperback. Harvard University Press, 2009. 7
- [61] Martin Rosvall and Carl T. Bergstrom. Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences of the United States of America, 105(4):1118–1123, 2008. 42
- [62] Arnaud Sallaberry, Faraz Zaidi, and Guy Melançon. Model for generating artificial social networks having community structures with small-world and scale-free properties. Social Network Analysis and Mining, 3(3):597–609, 2013. 45
- [63] Akrati Saxena, George Fletcher, and Mykola Pechenizkiy. Fairsna: Algorithmic fairness in social network analysis. arXiv preprint arXiv:2209.01678, 2022. 12

- [64] 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. 10, 31
- [65] 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. 54
- [66] Akrati Saxena, Harsh Saxena, and Ralucca Gera. k-truthscore: Fake news mitigation in the presence of strong user bias. In *International Conference on Computational Data and Social Networks*, pages 113–126. Springer, 2020. 10
- [67] Akrati Saxena, Pratishtha Saxena, and Harita Reddy. Fake news detection techniques for social media. In *Principles of Social Networking*, pages 325–354. Springer, 2022. 2
- [68] Akrati Saxena, Pratishtha Saxena, and Harita Reddy. Fake news propagation and mitigation techniques: A survey. In *Principles of Social Networking*, pages 355–386. Springer, 2022. 2, 6
- [69] Qihao Shi, Can Wang, Deshi Ye, Jiawei Chen, Yan Feng, and Chun Chen. Adaptive influence blocking: Minimizing the negative spread by observation-based policies. *Proceedings -International Conference on Data Engineering*, 2019-April:1502–1513, 2019. 31
- [70] Chonggang Song, Wynne Hsu, and Mong Li Lee. Temporal influence blocking: Minimizing the effect of misinformation in social networks. *Proceedings - International Conference on Data Engineering*, pages 847–858, 2017. 2, 10, 11, 56
- [71] Ana Andreea Stoica and Augustin Chaintreau. Fairness in social influence maximization. The Web Conference 2019 - Companion of the World Wide Web Conference, WWW 2019, pages 569–574, 2019. 7, 12, 24
- [72] Ana Andreea Stoica, Jessy Xinyi Han, and Augustin Chaintreau. Seeding Network Influence in Biased Networks and the Benefits of Diversity. The Web Conference 2020 - Proceedings of the World Wide Web Conference, WWW 2020, 2:2089–2098, 2020. 7, 12, 24, 25
- [73] Xinjue Wang, Ke Deng, Jianxin Li, Jeffery Xu Yu, Christian S. Jensen, and Xiaochun Yang. Targeted influence minimization in social networks., volume 23. Springer International Publishing, 2018. 31
- [74] Duncan J Watts and Steven H Strogatz. Strogatz small world network Nature. Nature, 393(June):440–442, 1998. 45
- [75] Peng Wu and Li Pan. Scalable influence blocking maximization in social networks under competitive independent cascade models. *Computer Networks*, 123:38–50, 2017. 6, 10, 11, 26, 27
- [76] Qingqing Wu, Xianguan Zhao, Lihua Zhou, Yao Wang, and Yudi Yang. Minimizing the influence of dynamic rumors based on community structure. *International Journal of Crowd Science*, 3(3):303–314, 2019. 30, 56
- [77] Ruidong Yan, Y I Li, Weili Wu, Deying Li, and Yongcai Wang. Rumor Blocking through Online Link Deletion. 13(2), 2019. 56
- [78] Lan Yang, Zhiwu Li, and Alessandro Giua. Containment of rumor spread in complex social networks. *Information Sciences*, 506:113–130, 2020. 54
- [79] Chang Yao, Yuanxing Zhang, Xiaomei Zhang, Kaigui Bian, and Lingyang Song. Competitive influence blocking in online social networks: A case study on WeChat. 2018 24th Asia-Pacific Conference on Communications, APCC 2018, pages 251–256, 2019. 9, 11

- [80] Qipeng Yao, Chuan Zhou, Ruisheng Shi, Peng Wang, and Li Guo. Topic-aware social influence minimization. WWW 2015 Companion - Proceedings of the 24th International Conference on World Wide Web, (1):139–140, 2015. 30
- [81] Qipeng Yao, Chuan Zhou, Linbo Xiang, Yanan Cao, and Li Guo. Minimizing the negative influence by blocking links in social networks. *Communications in Computer and Information Science*, 520(June):65–73, 2015. 30
- [82] Reza Zafarani, Mohammad Ali Abbasi, and Huan Liu. Social media mining: An introduction. Social Media Mining: An Introduction, 9781107018:1–320, 2014.
- [83] Wenlong Zhu, Wu Yang, Shichang Xuan, Dapeng Man, Wei Wang, Xiaojiang Du, and Mohsen Guizani. Location-Based Seeds Selection for Influence Blocking Maximization in Social Networks. *IEEE Access*, 7:27272–27287, 2019. 30, 33, 56

Appendix A Uniform model evaluation

In this Section, we include the evaluation of the methods under the uniform model with random negative seed S_N , where the edge probability values are set to be p = 0.1. This evaluation was performed on the Facebook network. We notice that because numerous nodes need to be considered in this scenario, as many nodes can reach each other with a relatively high probability, most methods scaled very poorly. For this reason, we chose a network with relatively small size 4039 nodes, but even in this case methods like *TIB*, *CMIA-O* and *Maximin-CMIA-O* failed to complete within 24 hours. In Figure A.1, we visualize then the efficiency and fairness of the remaining 3 benchmarks and the *FWRRS* method.



(a) Evaluation on Number of saved nodes, with vary- (b) Evaluation on maxi-min value, with varying k values ues

Figure A.1: Experimental analysis in the Facebook dataset under the Uniform model (p=0.1) with Random negative seed S_N

Under this propagation model, all methods perform similarly. Since information spreads easily through the network, due to the high edge probabilities, the truth can reach a vast majority of the nodes that were otherwise going to be infected. As a result, we observe in Figure A.1(a), even for a low budget such as k = 40 that all methods saved close to 100% of the population. Additionally, it can be observed that already with a low budget of k = 20, most methods achieve their maximum number of saved nodes, as their percentage curve flattens for larger budgets. However, such high effectiveness does not apply to the group level or community level. In Figure A.1(b) the maximin values are visualized, and it can be observed that only the proposed methods FWRRS and Stoica achieve a value above 0.1. All benchmarks neglect at least one community, even when saving nodes is as easy as under this propagation model. However, FWRRS achieves the highest maximin value

overall. Additionally, this fairness does not come at a cost, the percentage of saved nodes is very similar to the best-performing method.

Appendix B

Synthetic network generator model

In this section we introduce the HIgh Clustering Homophily Barabási-Albert (HICH-BA) model which generates the synthetic networks used in Section 5.2.1. The model is described in the following Algorithm 9.

Algorithm 9 HICH-BA (n, m, r, h, p_t , p_{PA})		
1:	$num_com \leftarrow \text{length of } r$	
2:	$G \leftarrow \text{empty undirected graph}$	
3:	for i:1 to num_com do	
4:	add a new node v to G with $comm_v = i$	
5:	$p_N \leftarrow 2n/m$	
6:	while number of nodes in $G < n$ do	
7:	add_node is set to True with probability p_N else to False	
8:	$\mathbf{if} \ add_node \ \mathbf{then}$	
9:	add new node <i>source</i> to G	
10:	source chooses its community c at random with probability in r_c	
11:	$options \leftarrow nodes w \text{ in } G \text{ with } comm_w = comm_{source} \text{ and } w \neq source$	
12:	preferential_attachment is set to True with probability p_{PA} else to False	
13:	if preferential_attachment then $weight_v \leftarrow deg_v$ for $v \in options$	
14:	else $weight_v \leftarrow 1$ for $v \in options$	
15:	else	
16:	$close_triangle$ is set to True with probability p_t else to False	
17:	${f if}\ close_triangle\ {f then}$	
18:	with probability p_{PA} weight $v \leftarrow deg_v + 1$ else weight $v \leftarrow 1$ for $v \in V$	
19:	$v \leftarrow \text{random node in } G \text{ with } deg_v \geq 2 \text{ according with probability } weight_v$	
20:	$source \leftarrow random neighbour of v$	
21:	intra-edge is set to True with probability h else to False	
22:	if intra-edge then	
23:	$options \leftarrow nodes w \ in \ Neigh_{inv} \setminus Neigh_{outsource} \ with \ comm_w = comm_{source}$	
24:		
25:	$options \leftarrow nodes w \ in \ Neigh_in_v \setminus Neigh_out_{source} \ with \ comm_w \neq comm_{source}$	
26:	preferential_attachment is set to True with probability p_{PA} else to False	
27:	if preferential_attachment then $weight_v \leftarrow deg_v + 1$ for $v \in options$	
28:	else $weight_v \leftarrow 1$ for $v \in options$	
29:	else	
30:	with probability p_{PA} weight $v \leftarrow deg_v + 1$ else weight $v \leftarrow 1$ for $v \in V$	

Fairness-aware Influence Blocking Maximization for combating Fake News

31:	source \leftarrow random node in G according with probability weight _{source}
32:	<i>intra-edge</i> is set to True with probability h else to False
33:	$\mathbf{if} \ intra - edge \ \mathbf{then}$
34:	options \leftarrow nodes w in G with $comm_w = comm_{source}$ and $w \neq source$
35:	else
36:	options \leftarrow nodes w in G with $comm_w \neq comm_{source}$
37:	preferential_attachment is set to True with probability p_{PA} else to False
38:	if preferential_attachment then weight _v \leftarrow deg _v for $v \in$ options
39:	else $weight_v \leftarrow 1$ for $v \in options$
40:	normalize list $weight$
41:	choose a node target out of options with probability $weight_{target}$
42:	add edge to G between <i>source</i> and <i>target</i>
43:	return G

The *HICH-BA* model takes as input 6 parameters. Parameters n and m denote respectively the number of nodes and edges desired in the network. The parameter r is a list containing probabilities, each entry r_i corresponds to the probability of a new node belonging to the community i. The homophily parameter h defines the probability a node will establish an internal connection in its community over an external one. Next, the parameter p_t is the probability with which a new edge will be chosen to close an existing triangle, the higher the value of this parameter the larger the resulting clustering coefficient will be. Last p_{PA} denotes the probability with which a new edge will be established using the preferential attachment, or at random otherwise.

As shown in Algorithm 9, the network is initialized to contain as many nodes as the number of communities in r, defined as num_com , since G initially contains a node belonging to each community. (Lines 1:4). Once the graph has been initialized, the probability of adding an edge or a node is computed. This variable p_N will control if in an iteration an edge is added or if a node is added, making sure that the resulting graph has the expected number of nodes n and edges m. Since every time a new undirected edge is added, this results in two new directed edges being included, the probability is set to be 2n/m (Line 5).

In every iteration p_N denotes if a new node or a new edge is to be added to the graph. In the scenario that a new node *source* arrives into the network the following is executed. The new node *source* first chooses the community to which it will belong, this is done according to the probabilities in the list r (Line 11). It will then be connected to the remaining graph with one edge. The nature of this edge is defined next. This node will connect to the community it chose to belong to, and the set of candidate nodes is gathered in the list *options*. Next, it is defined if the edge will be at random or according to the preferential attachment. This choice is made with the use of p_{PA} . With probability $1 - p_{PA}$ the edge is at random, so the weight of any node from *options* is set to be 1. On the other hand, with probability, p_{PA} , the edge is performed by preferential attachment. In this case, a node will prefer to connect to a node with more connections, thus, the weight of a node v in *options* is set to be $deg_v + 1$. The weights are later normalized in line 40 so that they become probabilities.

In the scenario that an edge is to be added to the graph the following is executed. Two different cases are differentiated. With probability p_t the new edge will be used to close an existing triangle and with probability $1 - p_t$ the edge will just connect two randomly chosen nodes. Under the first case, a random node v with a degree of at least 2 is chosen. This requirement is set, so that two nodes can be selected next. Additionally, v is chosen either based on its degree, with probability p_{PA} or at random otherwise. A random node *source* out of the neighbours of v is chosen, and the nature of the edge is next defined. First, if the edge will be inter-community or

intra-community is defined with the use of h. The subset of neighbours *source* can then connect to, is set to *options* (Lines 22:25). Second, we choose if the choice will be performed according to preferential attachment with probability p_{PA} or else at random, setting the weights accordingly. In the second case, where a new edge will be added to the graph, we repeat the same procedure as before, with the only difference that the *source* and *target* are chosen at random from all nodes instead of those within a neighbourhood.

In each iteration a *source*, a set of *options* and their corresponding *weights* are defined. In line 40 these weights are then transformed into probabilities by normalizing them against their sum. This means that

$$weight_v = \frac{weight_v}{\sum\limits_{w \in options} weight_w}$$

Using these probabilities and options, a weighted random choice is made, to choose target. Then an undirected edge between *source* and target is established, and the procedure continues. The process finalizes when the number of nodes in the graph is the one desired, n. Then the resulting graph G is returned.