GRAN SASSO SCIENCE INSTITUTE
SCHOOL OF ADVANCED STUDIES
Scuola Universitaria Superiore

Doctoral Thesis

Exploiting Social Influence to Control Opinions in Social Networks

Ph.D. Program in Computer Science: XXXII cycle

Author:
Federico Corò

Supervisors:
Gianlorenzo D'Angelo
Cristina M. Pinotti

November, 2019

GSSI Gran Sasso Science Institute
Viale Francesco Crispi, 7 - 67100 L'Aquila - Italy

Abstract

Social networks started as a place to comfortably connect with your friends. With them, we can communicate our thoughts and opinions over different topics and reach a large portion of users, even those who are not on your friend's list. This has led to making social networks a crucial part of many of us, providing for example information, entertainment, and learning. Many users prefer to access social networks, like Facebook or Twitter, to have access to news as they provide faster means for information diffusion. However, as a consequence, online social networks are also exploited as a tool to alter users' opinions, especially during political campaigns. A real-life example is the 2018 Cambridge Analytica scandal when it was revealed that the company had harvested personal data from Facebook users and used it for political advertising purposes. The idea was to target users with specific messages, which were meant to alter or reinforce user opinions. This is a concern for the health of our democracies which rely on having access to information providing diverse viewpoints. The aim of this work is to address the research issue of designing strategies to understand and overcome these processes that may have drastic consequences in our society.

We first consider the scenario in which a set of candidates are running for the elections and a social network of voters will decide the winner. Some attackers could be interested in changing the outcome of the elections by targeting a subset of voters with advertisement and/or (possibly fake) news. In this scenario we present two possible models that, exploiting influence in social networks, manipulate a voting process in order to make a target candidate win or lose the elections. We start by defining a model in which the preference list of each voter is known and give a constant factor approximation algorithm that can be used in *arbitrary scoring rule voting systems*, e.g., Plurality rule or Borda count. However, this assumption is not always satisfied in a realistic scenario as voters can be undecided on their preferences or they may not reveal them to the manipulator. Thus, we extend this model to design a scenario in which the manipulator can only guess a probability distribution over the candidates for each voter, instead of a deterministic preference list. Interestingly, while the problem can be approximated within a constant factor in the case of full knowledge, we show that, with partial information, the election control problem is hard to approximate within any constant factor through a reduction from Densest-k-subgraph problem, under some computational complexity hypotesis. However, we are able to show that a

small relaxation of the model allows us to give a constant factor approximation algorithm.

One of the possible ways to prevent election control for the integrity of voting processes is to reduce social biases and give to the users the possibility to be exposed to multiple sources with diverse perspectives and balancing users opinions by exposing them to challenging ideas. In this perspective we first investigate the problem from a computational point of view and generalize the work introduced by Garimella et al. [1] of *balancing information exposure* in a social network. In this setting we obtain strong approximation hardness results, however, we mitigate these hardness results by designing an algorithm with an approximation factor of $\Omega\left(n^{-1/2}\right)$.

Finally, we address the same issue of reducing the bias in social networks by proposing a link recommendation algorithm that evaluates the links to suggest according to their increment in social influence. We formulate the link recommendation task as an optimization problem that asks to suggest a fixed number of new connections to a subset of users with the aim of maximizing the network portion that is reached by their generated content. Thus, enhancing the possibility to spread their opinions.

“...the trouble about arguments is, they ain’t nothing but theories, after all, and theories don’t prove nothing, they only give you a place to rest on, a spell, when you are tuckered out butting around and around trying to find out something there ain’t no way to find out... There’s another trouble about theories: there’s always a hole in them somewheres, sure, if you look close enough.”

“Tom Sawyer Abroad”, Mark Twain

Contents

Preface

This thesis has two parts. Chapters 4 and 5 deal with Election Control Problems in Social Networks. We define the problem under the well-known Linear Threshold Model and give new algorithms with approximation guarantee to solve this scenario. This part is based on the following papers: [2–6].

[2] F. Corò, E. Cruciani, G. D'Angelo, S. Ponziani. "Vote for Me! Election Control via Social Influence in Arbitrary Scoring Rule Voting Systems". Extended Abstract In *18th International Conference on Autonomous Agents and MultiAgent Systems* (AAMAS 2019).

[3] F. Corò, E. Cruciani, G. D'Angelo, S. Ponziani. "Exploiting Social Influence to Control Elections Based on Scoring Rules". In *28th International Joint Conference on Artificial Intelligence* (IJCAI 2019).

[4] M. Aboueimehrizi, F. Corò, E. Cruciani, G. D'Angelo. "Election Control with Voters' Uncertainty: Hardness and Approximation Results". Preprint: *CoRR abs/1905.04694* (2019). Currently under review in IJCAI'20.

[5] M. Aboueimehrizi, F. Corò, E. Cruciani, G. D'Angelo, S. Ponziani. "Models and Algorithms for Election Control through Influence Maximization". Extended Abstract In *20th Italian Conference on Theoretical Computer Science* (ICTCS 2019).

[6] F. Corò, E. Cruciani, G. D'Angelo, S. Ponziani. "Exploiting Social Influence to Control Elections Based on Scoring Rules". Currently under review in *Journal of Artificial Intelligence Research* (JAIR).

In the second part, Chapters 6 and 7, we propose two solutions to reduce the bias in social networks, that can be seen as one of the possible ways to prevent election control. This part is based on the following papers:

[7] R. Becker, F. Corò, G. D'Angelo, H. Gilbert. "Balancing spreads of influence in a social network". In *34th AAAI Conference on Artificial Intelligence* (AAAI 2020). To appear.

[8] F. Corò, G. DAngelo, Y. Velaj. "Recommending Links to Maximize the Influence in Social Networks". In *28th International Joint Conference on Artificial Intelligence* (IJCAI 2019).

Besides the above publications, during my PhD, that I have started in November 2016, I worked on different projects:

Drones at Work: we investigated a combinatorial problem close to the Facility Location problem in which we aim at finding the best placement for a drone in a mixed-area, i.e., an area where different distance-measures are used.

F. Corò, C.M. Pinotti, L. Bartoli and A. Shende. "Drone Delivery System in a Mixed Landscape". Extended Abstract In *4th Italian Conference on ICT for Smart Cities And Communities* (I-Cities 2018).

F. Betti Sorbelli, F. Corò, C.M. Pinotti, A. Shende. "Automated Picking System Employing a Drone". In *1st International Workshop on Wireless sensors and Drones in Internet of Things* (Wi-DroIT 2019).

L. Bartoli, F. Betti Sorbelli, F. Corò, C.M. Pinotti, A. Shende. "Exact and Approximate Drone Warehouse for a Mixed Landscape Delivery System". In *5th IEEE International Conference on Smart Computing* (SMARTCOMP 2019).

Improving Connectivity: we studied the problem of adding edges in a graph in order to maximize the number of connected pairs of vertices.

F. Corò, C.M. Pinotti and G. D'Angelo. "On the Maximum Connectivity Improvement problem". In *14th International Symposium on Algorithms and Experiments for Wireless Networks* (ALGOSENSORS 2018).

F. Corò, G. D'Angelo, V. Mkrtchyan. "On the fixed-parameter tractability of the maximum connectivity improvement problem". Preprint: *CoRR abs/1904.12000* (2019).

Random Antenna Networks: We study the possibility of creating a fully connected ad-hoc network with bidirectional links between nodes equipped with directional antennas, randomly oriented, and deployed in a circular planar region. Note that these are the results of my masters thesis.

A. Bagchi, F. Coró, C.M. Pinotti and V. Ravelomanana. "Border Effects on Connectivity for Randomly Oriented Directional Antenna Networks". In *17th Annual Mediterranean Ad Hoc Networking Workshop* (Med-Hoc-Net 2018).

Chapter 1

Introduction

All of us have specific personal opinions on certain topics, such as lifestyle or consumer products. These opinions, normally formed on personal life experience and information, can be conditioned by the interaction with our friends leading to a change in our original opinion on a particular topic if a large part of our friends holds a different opinion. In our current society, the ability of information to spread is remarkable. Social media, and the internet as a whole, has provided people with more access to information than they have ever had before. And not only more accessible but quicker access. Many users prefer to use social networks, such as Facebook or Twitter, to have access to news as they provide faster means for information diffusion. Reports state that sixty-eight percent of users on both Facebook and Twitter use the social media platforms as their primary news source, an increase of 16 and 11 percent from 2013 numbers, respectively.[1] However, it turns out that users who prefer to get their news from social media are more likely to share fake news than those who prefer to get news via conventional methods, such as newspapers.[2] Fake-news are nowadays part of digital disinformation that enhances the opportunities for malicious actors to spread manipulated content online to shape users' opinion.

At the heart of this mechanism lies the diffusion process within a social network. Essentially, the diffusion process in a network can be described as a set of nodes, called sources, that are infected or active at the beginning of the process. Recursively, the

[1] https://www.journalism.org/2018/09/10/news-use-across-social-media-platforms-2018/

[2] https://www.journalism.org/2019/06/05/many-americans-say-made-up-news-is-a-critical-problem-that-needs-to-be-fixed/

infected nodes can activate their neighbors with some probability. The process will stop when no more infection occurs. One of the fundamental problems in the study of influence spreading is the problem of Influence Maximization. It has been proposed by Domingos and Richardson in the field of viral marketing and asks to find an initial set of users to be the early adopters of new technologies in order to activate a large cascade of further adoptions in the network [9, 10]. The problem has been formalized by Kempe et al. in 2003 as follows: if we are allowed to select at most a fixed number of users, i.e., those that will first spread the information, which ones should be selected in order to maximize the number influenced users resulting from the diffusion process [11]. In general, all existing diffusion models can be categorized into three classes: cascade models, threshold models, and epidemic models. The most popular for studying social influence problems are graph-based, namely, they assume an underlying directed graph where nodes represent agents and edges represent connections between them. Each node can be either *active*, that is it spreads the information, or *inactive*. With some probability, active nodes diffuse the information to their neighbors.

The physical or conceptual diffusion over a network has been studied in many domains ranging from viral marketing [9] and population epidemics [12], to social media [13]. In recent years, there has been a growing interest in the relationship between social networks and political campaigning. Political campaigns nowadays use online social networks to lead elections in their favor; for example, by spreading fake news, on the elections outcome [14]. In general, candidates to the elections or political parties, use social media to reach out to voters or mobilize supporters, while voters can use them to get involved in campaigns or election-related issues. Such activities can reinforce the integrity of our elections but at the same time can also be exploited as a tool to alter users' opinions and affect election results.

There exists evidence of political intervention which shows the effect of social media manipulation. One of the most significant examples is the one concerning the Cambridge Analytica company. In early 2018 it was revealed that the company had harvested millions of personal users data from Facebook and used them for political advertising purposes. The company was paid to develop an app that collected data such as likes and personal information from over 87 millions of Facebook accounts. Using this knowledge, the company was able to target users with highly personalized advertising based on their personality data. Various political parties worked along with Cambridge Analytica to attempt to influence public opinion. Political events linked to the scandal include: 2015 and 2016 campaigns of United States politicians Donald Trump and Ted Cruz; Brexit vote in 2016; the 2018 Mexican general election for Institutional Revolutionary Party.[3]

Another example, that does not involve Cambridge Analytica, is that of French elections in 2017, where automated accounts in Twitter, disguising themselves as human users, spread a considerable portion of political content, known as the "MacronLeaks disinformation campaign" trying to influence the outcome [15]. Many other real-life examples have been recorded and studied [14, 16–18].

These real-life examples show us that news diffusion in social networks, whether they are managed by a malicious user or by an algorithm, are likely to create homogeneous polarized clusters. This process leads users to get less exposure to conflicting viewpoints, making them manipulable. A good illustration of this issue was given by Conover et al. [19] who studied the Twitter network during the 2010 US congressional midterm elections. The authors demonstrated that the left and right-wing users were extremely isolated from each other, and only a limited connectivity between the two sides was present. A similar finding has been obtained by the Electome[4] project at the MIT Media Lab for the 2016 US presidential elections. Consequently, instead of giving users a diverse perspective and balancing users opinions by exposing them to challenging ideas, social media platforms are likely to make users more extreme by only exposing them to views that reinforce their pre-existing beliefs [19, 20].

In conclusion, if search engines or social networks can decide what users can see or read, it might have drastic consequences in our society. In this thesis, we aim at understanding and overcoming this process on social media. We start by designing two models to control election in social networks, from the manipulator point of view. We, then, design two algorithms to help reducing bias in social networks, via either selecting nodes or recommending new connections between users. We hope that our work could be a step further in the study of how to prevent opinion manipulations for the integrity of our society and in particular our voting process.

[3] https://en.wikipedia.org/wiki/Facebook%E2%80%93Cambridge_Analytica_data_scandal

[4] http://www.electome.org/

Main contributions

Based on the goal mentioned above, in this thesis, we give the following contributions.

We first introduce the *Linear Threshold Ranking*, a natural and powerful extension of the *Linear Threshold Model* for the election scenario that takes into account the degree of influence of the voters on each other. The goal of the problem is to select a fixed-size subset of the voters to start the diffusion process in order to maximize the chances of the target candidate to win the election (constructive scenario) or lose the election (destructive scenario). We first prove that maximize the Margin of Victory (MoV) of a target candidate under this model is $NP - hard$. Then, in order to approximately solve the problem, we provide an alternative and equivalent process that allows us to give a $\left(1 - \frac{1}{e}\right)$-approximation to the problem of maximizing the score of a target candidate by proving submodularity in the general case of the voting *scoring rule* for *arbitrary* scoring function In the end, exploiting such approximation, we are to provide a $\frac{1}{3}\left(1 - \frac{1}{e}\right)$-approximation to the problem of maximizing the MoV in the constructive scenario, independently from the number of candidates and for arbitrary scoring functions. We also give a simple reduction that maps destructive control problems to constructive control ones and allows us to achieve a $\frac{1}{2}\left(1 - \frac{1}{e}\right)$-approximation to the destructive control problem.

The previous model assumes to have a full knowledge of the preferences of each voter but this information is not always available since voters can be undecided or they may not want to reveal it. So we also propose this model that extend the previous one considering that each voter is associated with a probability distribution over the candidates. For this second model, called *Probabilistic Linear Threshold Ranking*, we first prove that the election control problem is hard to approximate within any constant factor by reducing from Densest-k-Subgraph [21]. To mitigate this hardness result we provide a slight relation of this model based on the idea that a voter might slightly change his idea even if the received influence is not enough to activate it. This relaxation, that we prove to be NP-hard, allows us to approximate the solution up to a $\frac{1}{6}(1 - \frac{1}{e})$ factor by giving a reduction to Influence Maximization in the weighted Linear Threshold Model and a $\frac{1}{4}(1 - \frac{1}{e})$-approximation for the destructive election control problem.

For the second part of the thesis, we first address the main open problem given by Garimella et al. [1]. The authors studied an optimization problem that aims at balancing information exposure when two opposing campaigns propagate in a network. In this thesis we generalize their optimization problem to a setting with arbitrarily many campaigns. Following Garimella et al., we investigate two settings, a simplified setting, in which we show that the problem can be approximated within a constant factor and a more general setting for which we give a reduction from densest-k-subgraph leading strong approximation hardness results. Nevertheless, we were able to design an algorithm with an approximation factor of $\Omega(n^{-1/2})$, where n is the number of nodes.

Finally, we consider the *Influence Maximization with Augmentation problem* introduced in [22] and give a constant-factor approximation algorithm for the problem of maximizing the social influence of a given set of target users by suggesting a fixed number of new connections. We then propose several techniques that heuristically speed up the running time of our algorithm and of that in [22] and experimentally show that, with few new links and small computational time, our algorithm can increase by far the social influence of the target users. We compare our algorithm with several baselines and show that it is the most effective one in terms of increased influence.

1.1 Thesis Outline

The rest of this thesis is organized as follows. In Chapter 2, we introduce the main definitions related to voting systems and to influence maximization in complex networks. Providing an overview of different rules and models used to capture the dynamics of influence spreading in networks as well as voting rules used to conduct elections. In Chapter 3 we give a review of the literature in this area.

Chapter 4 and Chapter 5 deal with Election Control Problems in Social Networks. We define the problem under the well-known Linear Threshold Model and give new algorithms with approximation guarantee to solve this scenario.

In Chapter 6 and Chapter 7 we propose two solutions to reduce the bias in social networks, that can be seen as one of the possible ways to prevent election control. We considered the problem of selecting seed or adding edges, respectively, in order to balance or maximize the spread of informations among the users of a social network.

Finally, in Chapter 8, we conclude and present several future research directions.

Chapter 2

Background

In this chapter, we describe the most widely studied models of information diffusion, *influence maximization* and notions and concepts about *voting systems* that will be used in the design and analysis of the algorithms.

2.1 Preliminaries

2.1.1 Approximation algorithms

In this subsection, we define the notion of approximation algorithms. Let Q be a maximization problem and let $\mathcal{I}$ be an instance of Q. Let $val_{\mathcal{I}}(S^*)$ and $val_{\mathcal{I}}(S)$ denote, respectively, the values of the optimal solution and the value of a solution produced by an approximation algorithm A on the instance $\mathcal{I}$. We define the measure of the quality of a solution S based on the approximation ratio as follows:

Definition 2.1. For a maximization problem Q, we say that an algorithm A is a α-approximation algorithm, with $0 \leq \alpha \leq 1$, if for every instance $\mathcal{I}$ of Q we have

$$\frac{val_{\mathcal{I}}(S)}{val_{\mathcal{I}}(S^*)} \geq \alpha$$

2.1.2 Graphs

A social network is represented by a weighted directed graph $G = (V, E, b)$, where nodes V represent users, edges E represent relationships between users, and the weight function $b : V \times V \to [0, 1]$ represents the influence between users. Moreover, we denote by N_v^- and N_v^+, respectively, the sets of incoming and outgoing neighbors for each node $v \in V$.

2.1.3 Submodular Set Functions

In order to study the spreading of information and influence maximization related problem we use properties of functions over set of individuals.

Submodularity is a property of set functions which is gaining popularity in a large number of areas and can help to model different kind of problems, such as Sensor placement [23], for gathering information, or Diverse web search and retrieval [24], the problem of finding a diverse set of articles in information retrieval and web search. In their seminal work about influence maximization Kempe et al. [11] have introduced the submodularity into the area of influence maximization.

Even if many different definitions for submodular functions can be found in the literature, Nemhauser et al. [25] proved the equivalence between each of them. In the same paper, they introduced a number of interesting properties about submodular functions that Kempe et al. [11] used as a foundation for their results, the most important result is presented here as Theorem 2.4. In general a function z is submodular if it satisfies the following property: the marginal gain from adding an element to a set S is at least as high as the marginal gain from adding the same element to a superset of S. Formally,

Definition 2.2 (Submodular)**.** For a ground set N, a function $z : 2^N \to \mathbb{R}$ is *submodular* if for any two sets S, T such that $S \subseteq T \subseteq N$ and for any element $e \in N \setminus T$ it holds that

$$z(S \cup \{e\}) - z(S) \geq z(T \cup \{e\}) - z(T).$$

Definition 2.3 (Monotone Submodular Function)**.** A submodular function z is called monotone if satisfies

$$z(S \cup \{e\}) \geq z(S), \quad \forall e.$$

Theorem 2.4 ([25, 26])**.** *Let $z(\cdot)$ be a non-negative monotone submodular function. Then the greegy algorithm that (for B iterations) adds an element with the largest marginal increase in $z(\cdot)$ produces a B-element set A^* such that*

$$z(A^*) \geq \left(1 - \frac{1}{e}\right) \max_{|A|=B} z(A)$$

2.2 Voting Systems

Voting systems are sets of rules that regulate all aspects of the voting process determining how election are conducted and how to determine the outcome. In particular a voting system decides candidates and voters eligibility, other than fixing the rules for determining the winner of the elections.

Social choice theory formally defines and analyzes voting systems, studying how the combination of individual opinions or preferences reaches a collective decision; *computational social choice*, instead, is a field at the intersection of social choice theory, theoretical computer science, and the analysis of multi-agent systems [27]. We refer interest readers to [28] for a comprehensive introduction to computational social choice theory.
It covers a whole spectrum of many different kind of problems arising from Voting Theory and Fair Allocation. In particular a category of our interest, consists on the analysis of the computational complexity of computing an outcome of voting rules that can serve as a barrier against strategic manipulation in elections [27, 29–31]. The usefulness of a particular voting system can be severely limited if it takes a very long time to calculate the winner of an election. Therefore, it is important to design fast algorithms that can evaluate a voting rule when given ballots as input. As is common in computational complexity theory, an algorithm is thought to be efficient if it takes polynomial time. Many popular voting rules can be evaluated in polynomial time in a straightforward way (i.e., counting), such as the Borda count, approval voting, or the plurality rule.

Certain voting systems, however, are computationally difficult to evaluate [32]. This has led to the development of approximation algorithms and fixed-parameter tractable algorithms to improve the theoretical calculation of such problems [33–35]. We also point out a recent work about fixed-parameter tractable algorithms on protecting elections [36].

In this work we mainly focus on two *single-winner* voting systems:

- *Plurality rule*: Each voter can only express a single preference among the candidates and that with the *plurality* of the votes wins, i.e., it is sufficient to have the highest number of votes and there is no need of an absolute majority (50%+1 of votes).
- *Scoring rule*: Each voter expresses his preference as a *ranking*; each candidate is then assigned a *score*, computed as a function of the positions he was ranked among the voters.

The former is arguably the simplest scenario and is one of the most commonly used for national legislatures and presidential elections. The latter is a very general definition, but can include several popular election methods by choosing an adequate *scoring function*:

- if the scoring function assigns 1 point to the first candidate and 0 to all the others this is equivalent to the *plurality rule*;
- if the scoring function assigns 1 point to the first t candidates and 0 to the others then it is equivalent to the t*-approval*, where each voter approves t candidates;
- if the scoring function assigns 1 point to the first $m - t$ candidates and 0 to the remaining t, where m is the total number of candidates, then it is equivalent to the t*-veto* or *anti-plurality* rule;
- if the scoring function assigns $m - l$ points to the candidate in position l then it is equivalent to the *borda count*, in which each voter ranks the candidates and each candidate gets a score equal to the number of candidates ranked lower in each list.

2.3 Influence Maximization

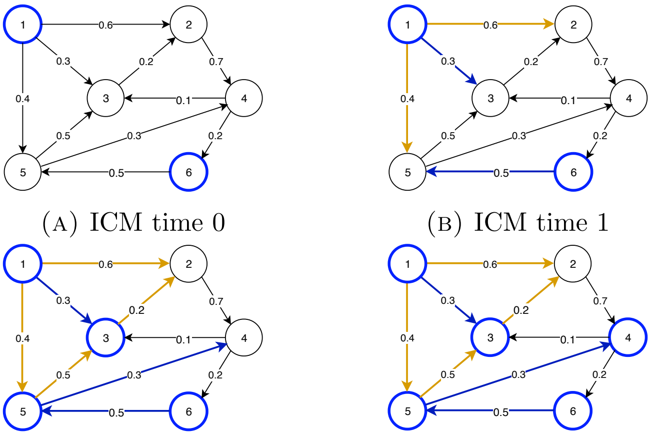
2.3.1 Influence Diffusion Models

In general, all existing diffusion models can be categorized into three classes: cascade models, threshold models, and epidemic models. The most popular for studying social influence problems are the *Independent Cascade Model* (ICM) and the *Linear Threshold Model* (LTM) [37]. These models are graph-based, namely they assume an underlying directed graph where nodes represent users and edges represent connections between them, e.g., friendships. We can distinguish in the process between nodes that spreads information, called *active*, and all the others, called *inactive*. Active nodes are the ones that, with some probability, diffuse the information to their neighbors.

For both models, the diffusion process proceeds iteratively in a synchronous way along a discrete time-axis, starting from an initial set of nodes, usually called *seeds*. Let $A_0 \subseteq V$ be the set of *active* nodes at the beginning of the process. More in general, let $A_t \subseteq V$ be the set of nodes active at time t. When a node is active, it influences its neighbors and increases the chance of making them change their preference list. The process has *quiesced* at the first time $\tilde{t}$ such that the set of active nodes would not change in the next round, i.e., time $\tilde{t}$ is such that $A_{\tilde{t}} = A_{\tilde{t}+1}$. We define the eventual set of active nodes as $A := A_{\tilde{t}}$.

Given a weighted graph $G = (V, E, b)$:

- **Independent Cascade Model.** Each edge $(u, v) \in E$ has a weight $b_{uv} \in [0, 1]$. During the process, if a node v is active at time $t \geq 0$ but was not active at time $t - 1$, i.e., $v \in A_t \setminus A_{t-1}$ (formally let $A_{-1} = \emptyset$), it tries to activate each neighbor u, independently, with a probability of success equal to b_{vu}. In case of success u becomes active for step $t + 1$, i.e., $u \in A_{t+1}$.
- **Linear Threshold Model.** Each node $v \in V$ has a threshold $t_v \in [0, 1]$ sampled uniformly at random and independently from the other nodes and each edge $(u, v) \in E$ has a weight $b_{uv} \in [0, 1]$ with the constraint that, for each $v \in V$, the sum of the weights of the incoming edges of v is less or equal to 1, i.e., $\sum_{(u,v)\in E} b_{uv} \leq 1$. During the process, an inactive node v becomes active if the sum of the weights of the edges coming from nodes that are active at the



(A) ICM time 0

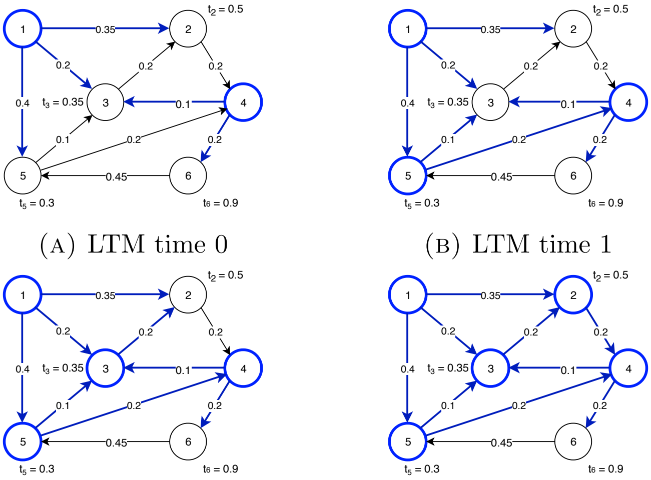
(B) ICM time 1

(C) ICM time 2

(D) ICM time 3

FIGURE 2.1: Diffusion process in ICM.

previous round is greater than or equal to its threshold t_v, i.e., $v \in A_t$ if and only if $v \in A_{t-1}$ or $\sum_{u \in A_{t-1}:(u,v) \in E} b_{uv} \geq t_v$.



(A) LTM time 0

(B) LTM time 1

(C) LTM time 2

(D) LTM time 2

FIGURE 2.2: Diffusion process in LTM.

2.3.2 Influence Maximization problem

The *influence maximization* problem studies a social network represented as a graph and has the goal of finding the B-sized set of influential nodes that can maximize the spread of information [11]. We first define the influence of a set of nodes:

Definition 2.5 (Influence)**.** The influence of a set of nodes A_0 denoted by $\sigma(A_0)$, is the expected number of active nodes at the end of the process, given that A_0 is the initial active set. Formally, $\sigma(A_0) = \mathbf{E}\left[|A_t|\right]$, where t is the (random) time of quiescence, i.e., when no activations occur from round t to $t+1$.

The Influence Maximization problem on a graph $G = (V, E, b)$ is defined as follows:

Definition 2.6 (Influence Maximization Problem)**.** Find a subset A_0^* of V with $|A_0^*| \leq B$ such that $\sigma(A_0^*, G) \geq \sigma(A_0)$ for every $A_0 \subseteq V$ with $|A_0| \leq B$.

Kempe et al. showed that the distribution of the set of active nodes in the graph starting from the set of seed nodes A_0, under both the ICM and LTM process is equivalent to the distribution reachable from the same set A_0 in the set of random graphs called *live-edge graphs* ([11, Theorem 4.5, 4.6], here reported as Theorem 2.7), since the processes are point-wise identical, also the expected number of activated nodes have to be the same.

Given a graph $G = (V, E, b)$, a live-edge graph $G' = (V, E')$ is built as follows:

ICM Every node $v \in V$ picks a subset of $T \subset N_v^-$ according to a distribution over subsets of its in-neighbors. Formally,

$$P\left(T\right) = \prod_{u \in T} b_{uv} \cdot \prod_{u \in N_v^- \setminus T} (1 - b_{uv}).$$

LTM Every node $v \in V$ picks at most one of its incoming edges with probability proportional to the weight of that edge, i.e., edge (u, v) is selected with probability b_{uv}, and no edge is selected with probability $1 - \sum_{u \in N_v^-} b_{uv}$.

Theorem 2.7 (Kempe, Kleinberg, and Tardos [11])**.** *Given a graph $G = (V, E)$ and an initial set of nodes $A_0 \subseteq V$, the distribution of the sets of* active *nodes in G after the ICM (LTM) has quiesced starting from A_0 is equal to the distribution of the sets of nodes that are* reachable *from A_0 in the set of live-edge graphs $\mathcal{G}$, i.e., $\mathbf{P}\left(v \in A\right) = \mathbf{P}\left(v \in R(A_0)\right)$, for any node $v \in V$.*

In the thesis we will use the following corollary.

Corollary 2.8. *For any set of initially active nodes A_0 and for any node v,*

$$\mathbf{P}\left(v \in R(A_0)\right) = \sum_{G' \in \mathcal{G}} \mathbf{P}\left(G'\right) \cdot \mathbf{1}_{(G', A_0, v)}.$$

Proof. By the law of total probability we have that

$$\mathbf{P}\left(v \in R(A_0)\right) = \sum_{G' \in \mathcal{G}} \mathbf{P}\left(v \in R(A_0) \mid G'\right) \cdot \mathbf{P}\left(G'\right) = \sum_{G' \in \mathcal{G}} \mathbf{P}\left(G'\right) \cdot \mathbf{1}_{(G', A_0, v)},$$

since, given a live-edge graph G' sampled form $\mathcal{G}$, the value of $\mathbf{P}\left(v \in R \mid G'\right)$ is equal to 1 if v is reachable from A_0 in G' and to 0 otherwise. □

Moreover, under the live-edge model, the problem of selecting the initial set of nodes in order to maximize the diffusion is *submodular* [11]. Therefore, exploiting a classical result [25], the influence maximization problem can be approximated to a constant factor of $1 - {}^1\!/_e$ using a simple greedy hill-climbing approach that starts with an empty solution and, for B iterations, selects a single node that gives the maximal marginal gain on the objective function with respect the solution computed so far. The algorithm is presented in a simple version in Algorithm 1.

Theorem 2.9 (Kempe, Kleinberg, and Tardos [11])**.** *In the Linear Threshold and Independent Cascade models, there is a polynomial-time algorithm approximating the maximum influence to within a factor of $(1 - {}^1\!/_e - \epsilon)$, where e is the base of the natural logarithm and ϵ is any positive real number.*

Algorithm 1 Greedy algorithm for Influence Maximization [11].

```
Start with $A_0 = \emptyset$
while $|A_0| \leq B$ do
    Add the node with largest estimate for $\sigma(A_0 \cup \{v\})$ to $A_0$.
Output the set $A_0$ of nodes
```

Algorithm 1 guarantees the best approximation, but is still computational very expensive: Evaluating the expected number of active nodes is $\#P$-hard [38, 39]. However, there exists a simulation-based approach in which a Monte-Carlo simulation is performed to evaluate the influence spread of a given seed set A_0 [11]. The standard Chernoff-Hoeffding bounds imply $1 \pm \lambda$ approximation to the expected number of active nodes by simulating a polynomial number of times the diffusion process, Theorem 2.10.

Theorem 2.10 (Kempe, Kleinberg, and Tardos [11])**.** *If the diffusion process starting with A_0 on graph G is simulated at least $\Omega\left(\frac{n^2}{\lambda^2}\ln\frac{1}{\delta}\right)$ times, then the average number of activated nodes over these simulations is a $(1 \pm \lambda)$-approximation to $\sigma(A_0)$, with probability at least $1-\delta$.*

Note that, by using $(1\pm\lambda)$-approximate values when optimizing σ the greedy algorithm still guarantees a $(1 - {}^{1}\!/_{e} - \epsilon)$ approximation factor, where ϵ depends on λ and goes to 0 as $\lambda \to 0$.

2.4 Election Control Problem

The study of controlling elections is fundamental to computational social choice: it is widely studied from a theoretical perspective, and has deep practical impact. In this thesis we model the scenario of Election Control in social networks representing the underlying connections between users as a directed graph $G = (V, E)$. Let $C = \{c_1, \dots, c_m\}$ be the set of m candidates; we refer to our *target candidate*, i.e., the one that we want to make win/lose the elections, as $c_\star$.

Each voter $v \in V$ has a list of preferences for the elections represented as a function $\pi : C \to \mathbb{R}$, this function can, for example, denote the position of a candidate in the preference list of node, e.g., $\pi_v(c_i) = 3$ for a given candidate c_i means that in the preference list of voter v such candidate is ranked as third. Let us consider the general case of the *scoring rule*, where a *non-increasing scoring function* $f : \{1, \dots, m\} \to \mathbb{N}$ assigns a score to each position. If the lists are modified by a process the candidates might have a new position in the preference list of each node $v \in V$; we denote such new preference list as $\tilde{\pi}$. Then, we define the *score* of a candidate c_i as the number of votes that c_i obtains from the voters, that is,

$$F(c_i, \emptyset) := \sum_{v \in V} f(\pi_v(c_i))$$

and the expected score of a candidate c_i at the end of the process as

$$F(c_i, S) := \mathbf{E}\left[\sum_{v \in V} f(\tilde{\pi}_v(c_i))\right]$$

where S is the set of seed nodes.
As a running example, let $C = \{c_\star, c_2, c_3, c_4, c_5\}$ and $n = 2$. We define the two voters to have preferences, from the highest to the least preferred, $\pi_{v_1} = c_5, c_\star, c_3, c_2, c_4$ and $\pi_{v_2} = c_2, c_3, c_5, c_4, c_\star$, respectively. If we are using Borda rule then the scores of $c_\star, c_2, c_3, c_4, c_5$ following the elections are $3, 5, 5, 1, 6$. Now consider to be able to move $c_\star$ in π_{v_2} from the last to the second position, i.e., $\tilde{\pi}_{v_2} = c_2, c_\star, c_3, c_5, c_4$. Then the new scores are $6, 5, 4, 0, 5$.

In the problem of *election control* we want to maximize the chances of the target candidate to win the elections. Namely, the goal is to maximize the probability that a target candidate wins the elections. However, maximizing such probability is hard to approximate to within any multiplicative factor for both constructive and destructive control, even in elections with only two candidates [40]. Thus, to achieve our aim we focus on a different objective function, that is, we maximize its expected *Margin of Victory* (MoV) w.r.t. the most voted opponent, akin to that defined in [40]. We define the MoV(S) as the expected increase of the difference between the score of $c_\star$ and that of the most voted opponent. Formally, if c and $\bar{c}$ are respectively the candidates different from $c_\star$ with the highest score before and after the LTM process, the MoV is

$$\text{MoV}(S) := F(c, \emptyset) - F(c_\star, \emptyset) - (F(\bar{c}, S) - F(c_\star, S)).$$

Note that we preferred to use the increment in the margin of victory rather than the margin itself to have well-defined approximation ratios.

Given a budget B, the *election control problem* asks to find an initial set of seed nodes S, of size at most B, that maximizes the MoV, i.e.,

$$\begin{aligned} \arg\max_S \quad & \text{MoV}(S) \\ \text{s.t.} \quad & |S| \leq B. \end{aligned}$$

It is worth noting that MoV can also be expressed as a function of the score gained by $c_\star$ and the score lost by its most voted opponent $\bar{c}$ at the end of the LTM process. We define the score gained and lost by a candidate c_i as

$$g^{+}(c_i, S) := F(c_i, S) - F(c_i, \emptyset) \qquad \text{and} \qquad g^{\text{-}}(c_i, S) := F(c_i, \emptyset) - F(c_i, S).$$

Therefore, we can rewrite $\mathrm{MoV}(S)$ as

$$\mathrm{MoV}(S) = g^{+}\left(c_\star, S\right) + g^{\text{-}}\left(\bar{c}, S\right) - F\left(\bar{c}, \emptyset\right) + F\left(c, \emptyset\right).$$

We define similarly the *destructive election control* problem where we want to maximize the chances of the target candidate to *lose* the elections. Formally, the problem can be defined as that of finding an initial set of seed nodes S such that

$$\begin{aligned} \max_S \quad & \mathrm{MoV}_D(S) := F\left(\bar{c}, S\right) - F\left(c_\star, S\right) - \left(F\left(c, \emptyset\right) - F\left(c_\star, \emptyset\right)\right). \\ \text{s.t.} \quad & |S| \leq B, \end{aligned}$$

namely to find an initial set of seed nodes of at most size B that maximizes the expected MoV_D, i.e., minimizes the expected MoV.

2.4.1 Score Approximates Margin of Victory

In the following we show that if there exists an approximation algorithm to the problem of maximizing the increment in score of the target candidate, then, we can achieve an approximation to the original problem of maximizing its MoV, at the cost of an extra constant approximation factor.

The next theorem generalizes [40, Theorem 5.2] as it holds for any scoring rule and for any model in which we have the ability to change only the position of a target candidate $c_\star$ in the lists of a subset of voters and the increment in score of $c_\star$ is at least equal to the decrement in scoring of the other candidates.

We show that we can approximate the optimal MoV to within a constant factor by optimizing the increment in the score of candidate $c_\star$. In detail, we show that, given two solutions S and S^* that maximize $g^{+}\left(c_\star, S\right)$ and $\mathrm{MoV}(S^*)$ respectively, it holds $\mathrm{MoV}(S) \geq \frac{1}{3}\mathrm{MoV}(S^*)$. Indeed, we show a more general statement that is: If a solution S approximates $\max_T g^{+}\left(c_\star, T\right)$ within a factor α, then $\mathrm{MoV}(S) \geq \frac{\alpha}{3}\mathrm{MoV}(S^*)$.

Theorem 2.11. *An α-approximation algorithm for the problem of maximizing the increment in score of a target candidate gives an $\frac{\alpha}{3}$-approximation to the election control problem.*

Proof. Let us consider S and S^* as two solutions for the problem of maximizing the MoV for a target candidate $c_\star$, with S^* as the optimal solution to this problem. These solutions arbitrarily select a subset of voters and modify their preference list changing the score of $c_\star$.

Let us fix c and $\bar{c}$, respectively, as the candidates different from $c_\star$ with the highest score before and after the solution S is applied and $\hat{c}$ is the candidate with the highest score after the solution S^* is applied. If we do not consider the gain given by the score lost by the most voted opponent and we assume there exists an α-approximation to the problem of maximizing the increment in score of the target candidate, we have that

$$\begin{aligned}\mathrm{MoV}(S) &= g^+(c_\star, S) + g^-(\bar{c}, S) - F(\bar{c}, \emptyset) + F(c, \emptyset) \\ &\geq \alpha g^+(c_\star, S^*) - F(\bar{c}, \emptyset) + F(c, \emptyset) \\ &\geq \frac{\alpha}{3}\left[g^+(c_\star, S^*) + g^-(\bar{c}, S^*) + g^-(\hat{c}, S^*)\right] - F(\bar{c}, \emptyset) + F(c, \emptyset),\end{aligned}$$

where the last inequality holds because $g^+(c_\star, S) \geq g^-(c_i, S)$ for any solution S and candidate c_i due to the fact that the solution S is able to modify only the score of the candidate $c_\star$, increasing it, while the score of all the other candidates is decreased, and the increment in score to $c_\star$ is equal to the sum of the decrement in score of all the other candidates.

Since $F(\bar{c}, \emptyset) \leq F(c, \emptyset)$, we have that

$$\begin{aligned}\mathrm{MoV}(S) &\geq \frac{\alpha}{3}[g^+(c_\star, S^*) + g^-(\bar{c}, S^*) + g^-(\hat{c}, S^*) - F(\bar{c}, \emptyset) + F(c, \emptyset)] \\ &= \frac{\alpha}{3}\left[g^+(c_\star, S^*) + g^-(\bar{c}, S^*) - F(\bar{c}, \emptyset) + F(c, \emptyset) + g^-(\hat{c}, S^*) - F(\hat{c}, \emptyset) + F(\hat{c}, \emptyset)\right] \\ &= \frac{\alpha}{3}\left[\mathrm{MoV}(S^*) + g^-(\bar{c}, S^*) - F(\bar{c}, \emptyset) + F(\hat{c}, \emptyset)\right],\end{aligned}$$

Recall that $\hat{c}$ is the most voted opponent after the optimal solution S^* has been applied, we have, by definition of $\hat{c}$, that $F(\hat{c}, S^*) \geq F(\bar{c}, S^*)$, which implies that

$$g^-(\hat{c}, S^*) - g^-(\bar{c}, S^*) = F(\hat{c}, \emptyset) - F(\hat{c}, S^*) - F(\bar{c}, \emptyset) + F(\bar{c}, S^*) \leq F(\hat{c}, \emptyset) - F(\bar{c}, \emptyset).$$

Thus, $g^-(\bar{c}, S^*) - F(\bar{c}, \emptyset) + F(\hat{c}, \emptyset) \geq 0$ and we can conclude that $\mathrm{MoV}(S) \geq \frac{\alpha}{3}\mathrm{MoV}(S^*)$. □

Chapter 3

A survey of related works

In this chapter we survey previous work in the literature to place our work in context. We briefly review each of the main concepts that relate to models and solutions discussed in this thesis.

3.1 Election Control in Social Networks

There exist an extensive literature about manipulating a voting system without considering the underlying social network of the voters, e.g., swap bribery [41], shift bribery [42]; we point the reader to a recent survey [28]. These works study the complexity of bribery in elections, that is, the complexity of computing whether it is possible, by modifying the preferences of a given set of voters, to make some target candidate win or lose the elections.

Despite the large literature on influence diffusion and on election manipulation, there are only few studies on the problem of manipulating the outcome of a political election by using influence diffusion in social networks. The Independent Cascade Model [11] has been considered as diffusion process to guarantee that a target candidate wins/loses [43, 44]. The constructive (destructive) election control problem via influence maximization has been recently introduced by Wilder et al. and consists in selecting a subset of the nodes of a network to be the starter of the diffusion with the aim of maximizing the chances of victory or loosing of a target candidate [40]. They use the Independent Cascade Model as model of diffusion and plurality as voting system.

When a voter is reached by the social influence, it changes the ranking of the target candidate by one position. A variant of the Linear Threshold Model [11] with weights on the vertices has been considered on a graph in which each node is a cluster of voters with a specific list of candidates and there is an edge between two nodes if they differ by the ordering of a single pair of adjacent candidates [45]. They prove that the problem of making a specific candidate win in their model is *NP*-hard and fixed-parameter tractable with respect to the number of candidates. Moreover, it has been studied how to manipulate the network in order to have control on the majority opinion, e.g., bribing or adding/deleting edges, on a simple Linear Threshold Model where each node holds a binary opinion, each edge has a fixed weight, and all vertices have a threshold fixed to 1/2 [46]. In their work, they studied the hardness and the parameterized complexity of the manipulations problem they proposed. The study of opinion diffusion modeled as a majority dynamics has attracted much attention in recent literature [47–49]. In these models, each agent has an initial preference list and at each time step a subset of agents updates their opinions according to some majority-based rule that depends on their neighbors in the network.

All these previous models consider to have full knowledge over the preferences' list of the voters, however, this assumption is not always satisfied in a realistic scenario as voters can be undecided on their preferences or they may not reveal them to the manipulator. Modeling uncertainty in political elections has been already considered in the literature, for example, the study of the uncertainty introduced by incomplete data given to the problem [50–52], or models in which candidates may change during the election campaign [53, 54], or the vote of a bribed voter may or may not be counted [55].

Our Contributions. In this context, to improve the modeling of real-world scenarios we propose two new models that we call *Linear Threshold Ranking* (LTR) and the *Probabilistic Linear Threshold Rankings* (PLTR), natural and powerful extensions of the well-established *Linear Threshold Model.* These models allow us to keep into account the degree of influence that each voter exercise on the others. Compared with previous work, the new models we propose can describe scenarios in which a high influence on a voter can radically change its preference.

Differently from previous works, in the LTR model, we also consider more general voting systems, i.e., all voting systems that can be expressed with a *scoring rule*: Each voter ranks the candidates according to her preferences and a score is assigned to each position. Instead, PLTR we extend the previous model to design a scenario in which the manipulator can only guess a probability distribution over the candidates for each voter.

In the LTR setting, we prove the nontrivial fact that any scoring function is monotone and submodular with respect to the initial set of active nodes. Moreover, we exploit this fact to prove a constant-factor approximation of the election control problem in our model. In the PLTR setting, instead, we show that the election control problem is hard to approximate to within a polynomial fraction of the optimum through a reduction from Densest-k-Subgraph problem. However, we are able to show that a small relaxation of the model allows us to give a constant-factor approximation algorithm.

3.2 Multiple Campaigns in Parallel

Here we focus on the literature about multiple campaigns running simultaneously on the same network. Budak et al. [56] studied the problem of limiting as much as possible the spread of a "bad" campaign by starting the spreading of another "good" campaign that blocks the first one. The two campaigns *compete* on the nodes that they reach: once a node becomes active in one campaign it cannot change campaign. They prove that the objective function is monotone and submodular and hence they obtain a constant approximation ratio. Similar concepts of *competing cascades* in which a node can only participate in one campaign have been studied in several works [57–63]. Game theoretic aspects like the existence of Nash equilibria have been also investigated in this case [64–66]. Borodin et al. [67] consider the problem of controlling the spread of multiple campaigns by a centralized authority. Each campaign has its own objective function to maximize associated with its spread and the aim of a central authority is to maximize the social welfare defined as the sum of the selfish objective function of each campaign. They propose a truthful mechanism to achieve theoretical guarantees on the social welfare. The three main works closely related to ours are the ones of Aslay et al. [68], Matakos et al. [69] and Garimella et al. [1].

Aslay et al. [68] tackles an item-aware information propagation problem in which a centralized agent must recommend some articles to a small set of seed users such that the spread of these articles maximizes the expected diversity of exposure of the agents. The diversity exposure is measured by a sum of agent-dependent functions that takes into account user leanings. The authors show that the *NP*-hard problem they define amounts to optimizing a monotone and submodular function under a matroid constraint and design a constant factor approximation algorithm. Matakos et al. [69] models the problem of maximizing the diversity of exposure in a social network as a quadratic knapsack problem. Here also the problem amounts to recommending a set of articles to some users in order to maximize a diversity index taking into account users' leanings and the strength of their connections in the social network. The authors show that the resulting diversity maximization problem is inapproximable and design a polynomial algorithm without an approximation guarantee.

The main work closely related to ours is the one of Garimella et al. [1]. Here the authors introduced the problem of *balancing information exposure* in a social network. Following the *influence maximization paradigm* going back to the seminal work of Kempe et al. [11], their problem involves two opposing viewpoints or campaigns that propagate in a social network following the *Independent Cascade Model.* Given initial seed sets for both campaigns, a centralized agent is then responsible for selecting a small number of additional seed users for each campaign in order to maximize the number of users that are reached by either both or none of the campaigns. The authors study this problem in two different settings, namely the *heterogeneous* and *correlated* settings. The heterogeneous setting corresponds to the general case in which there is no restriction on the probabilities with which the campaigns propagate. Contrarily, in the correlated setting, the probability distributions for different campaigns are identical and completely correlated. After proving that the optimization problem of balancing information exposure is *NP*-hard, the authors designed efficient approximation algorithms with an approximation ratio of $(1 - {}^{1}\!/\!_{e} - \epsilon)/2$ for both settings.

Our Contributions. In this context, we study the main open problem in X, by generalizing their optimization problem to a setting with arbitrarily many campaigns. This generalization is motivated by the fact that for most problems, not only two but a multitude of viewpoints are perceivable. We prove that the generalized problem can be approximated within a constant factor in the correlated setting. Moreover, we prove that in the heterogeneous scenario the problem is hard to approximate within any constant factor. Finally, we mitigate this hardness results by designing an algorithm with an approximation factor of $\Omega(n^{-1/2})$ for the case of three campaigns, where n is the number of nodes in the network.

3.3 Recommending Links in Social Networks

There exist an extensive literature on the problem of recommending links to users of a social networks; we point the reader to recent surveys [70–72]. However, there are only few studies on the problem of adding new links in a network taking into account social influence. In the following, we focus on two widely studied diffusion models, the *Independent Cascade Model* (ICM) and the *Linear Threshold Model* (LTM) [11], and review papers that study network modification problems in these two models.

ICM has been considered in several studies. D'Angelo et al. [22] introduced the *Influence Maximization with Augmentation* problem (IMA) that consists in adding a limited number of edges incident to a given set of nodes in order to maximize their capability of spreading information. They proved that such problem is *NP*-hard to be approximated within a constant factor greater than $1 - (2e)^{-1}$ and provide a greedy approximation algorithm that almost matches such upper bound. Sheldon et al. [73] study the problem of adding nodes in a network to maximize the information diffusion in a network. They provide a counterexample showing that the objective function is not submodular and propose exact integer programming formulations. Other types of graph modifications such as modifying the probability of infecting other nodes, have been considered by Wu et al. [74]. They proved that optimizing the selection of such modifications is *NP*-hard and is neither submodular nor supermodular.

We now review papers on network modification problems under LTM. Heuristics for the edge removal problem have been studied in [75, 76] but without providing an approximation guarantee. Khalil et al. [77] consider two types of graph modification, adding/deleting edges in order to minimize the information diffusion showing that this network structure modification problem has a supermodular objective and therefore can be solved by algorithms with proven approximation guarantees. Zhang et al. [78] consider the problem removing edges and nodes with the aim of minimizing the information diffusion and develop algorithms with rigorous performance guarantees and good empirical performance. Experimental studies show that increasing the connectivity or the centrality of a node, by adding edges to the graph, lead also to an increase in the expected number of nodes that the diffusion process is able to reach [79–81].

Making Recommendations to Decrease Polarization. As discussed in Chapter 1 social media has been blamed for creating "echo chamber" through algorithms, encouraging users to connect only to other like-minded users creating groups where it is easy to be influenced. Communication between different groups is usually difficult, i.e., there are many vertices for which all or most of their neighbors belong to the same group. Creating a situation where most of the information that a user can receive comes from inside the same group to which it belongs.

The existence of these chambers is an obstacle to the functioning of society and democratic processes. The studies discussed above focus mostly on optimization problems with the idea of adding edges to a graph to improve specific performance measures or to improve the ability of the graph to disseminate information. Nonetheless there exist a lot of studies that propose to solve the problem of decreasing polarization, we refer interest readers to [82–84], however, there is a lack of algorithmic approaches in these works.

The papers that are conceptually closest to this issue are the one by Interian et al. [85] and Garimella et al. [86], however, none of them take into account social influence and diffusion processes that occur in social networks. Interian et al. proposed a minimum-cardinality balanced edge addition problem, which consists in assuming that the lowest number of changes should be made in the original network. The authors proved that the problem is *NP*-hard and propose three integer linear programming formulations to solve the problem. Garimella et al., instead, formulate the problem of finding the edges to minimize the controversy score in the network. In particular, given a metric that measures the controversy of an issue their goal is to find a fixed number of edges that minimize such measure. The authors proposed an efficient algorithm, though, without providing an approximation guarantee. We remark that in both these works the diffusion process that takes place within a social network is not considered.

Our Contributions. In this context, we study an algorithmic technique for bridging these chambers and thus reduce the bias taking into account the social influence effect. We formulate a link recommendation task as an optimization problem that asks to suggest a fixed number of new connections to a subset of users to maximize the network portion that is reached by their generated content. In detail, we give a constant-factor approximation algorithm for the problem of maximizing the social influence of a given set of target users by suggesting a fixed number of new connections. Moreover, we empirically show that by adding few links to the network our algorithm can increase by far the social influence of the target users.

Chapter 4

Linear Threshold Ranking and Election Control

In this chapter, we consider the problem of exploiting social influence in a network of voters in order to change their opinion about a target candidate with the aim of increasing his chance to win/lose the election in a wide range of voting systems.

We introduce the *Linear Threshold Ranking*, a natural and powerful extension of the well-established *Linear Threshold Model*, which describes the change of opinions taking into account the amount of exercised influence. We are able to maximize the score of a target candidate up to a factor of $1 - {}^{1}\!/_{e}$ by showing submodularity. We exploit such property to provide a $\frac{1}{3}(1 - {}^{1}\!/_{e})$-approximation algorithm for the *constructive* election control problem. Similarly, we get a $\frac{1}{2}(1 - {}^{1}\!/_{e})$-approximation ratio in the *destructive* scenario. The algorithm can be used in *arbitrary scoring rule voting systems*, including *plurality rule* and *borda count*.

Most of the results presented in this chapter are included in [2, 3, 5, 6].

4.1 Problem Definition

We first introduce a deterministic model called *Linear Threshold Ranking* (LTR), based on LTM, that takes into account the degree of influence that voters exercise on each other. As in LTM, each node $v \in V$ has a threshold $t_v \in [0, 1]$; each edge $(u, v) \in E$

has a weight b_{uv} with the constraint that $\sum_{u:(u,v)\in E} b_{uv} \leq 1$. Moreover, each node v has a permutation π_v of C, i.e., its list of preferences for the elections; in this case $\pi_v(c_i)$ denote the position of candidate c_i in the preference list of node v.

We consider the LTM process starting from an initial set of active nodes $S \subseteq V$. Recall that, according to LTM, each node $v \in V$ has a threshold t_v, each edge $(u, v) \in E$ has a weight b_{uv}, and that $A \subseteq V$ is the set of active nodes at the end of the process.

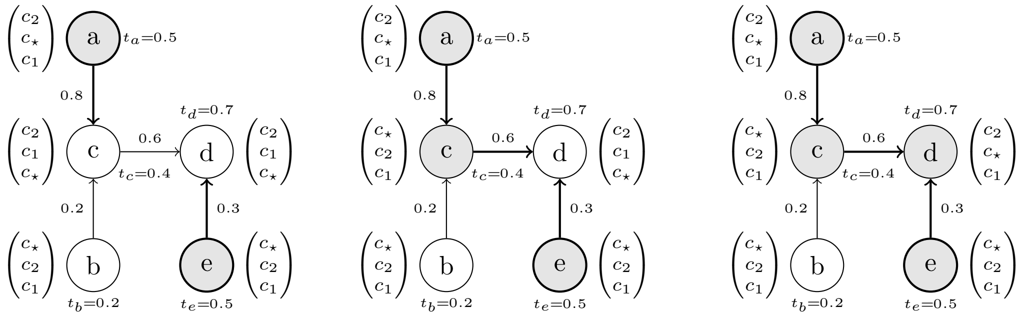
Let $B \in \mathbb{N}$ be an initial budget that can be used to select the nodes in S, i.e., the set of active nodes from which the LTM process starts. In particular, the budget constrains the size of S, namely $|S| \leq B$.

After the LTM process has quiesced, the position of $c_\star$ in the preference list of each node changes according to a function of its incoming active neighbors. The threshold t_v of each node $v \in V$ models its strength in retaining its original opinion about candidate $c_\star$: The higher is the threshold t_v the lower is the probability that v is influenced by its neighbors. Moreover the weight on an edge b_{uv} measures the influence that node u has on node v. Taking into account the role of such parameters, we define the number of positions that $c_\star$ goes up in π_v as

$$\pi_v^{\uparrow}(c_\star) := \min\left(\pi_v(c_\star) - 1,\ \left\lfloor \frac{\alpha(\pi_v(c_\star))}{t_v} \sum_{u\in A,\, (u,v)\in E} b_{uv} \right\rfloor\right), \tag{4.1}$$

where $\alpha : \{1, \ldots, m\} \to [0, 1]$ is a function that depends on the position of $c_\star$ in π_v and models the rate at which $c_\star$ shifts up. Note that α can be set arbitrarily to model different scenarios, e.g., shifting up of one position from the bottom of the list could be easier than going from the second position to the first with a suitable choice of α. As for $\pi_v^{\uparrow}(c_\star)$, it can be any integer value in $\{0, \ldots, \pi_v(c_\star) - 1\}$: The floor function guarantees a positive integer value; the minimum between such value and $\pi_v(c_\star) - 1$ guarantees that final position of $c_\star$ is at least 1, since the floor function could output too high values when the threshold is small w.r.t. the neighbors' influence. We call this process the *Linear Threshold Ranking* (LTR).

After the modification of the lists at the end of LTR, the candidates might have a new position in the preference list of each node $v \in V$; we denote such new preference list as $\tilde{\pi}$. In particular, the new position of candidate $c_\star$ will be $\tilde{\pi}_v(c_\star) := \pi_v(c_\star) - \pi_v^{\uparrow}(c_\star)$; the candidates that are overtaken by $c_\star$ will shift one position down.



(A) LTR time 0 (B) LTR time 1 (C) LTR time 2

FIGURE 4.1: Example of an election with three candidates $\{c_1, c_2, c_\star\}$. Gray nodes represent seeds. The tuples $(\cdot)$ on the side of the nodes are the voters preferences.

As a running example, let us consider five voters and relative connections as depicted in Figure 4.1. If we first consider plurality rule we have that the scores of $c_1, c_2, c_\star$ before the elections are $0, 3, 2$, thus with c_2 currently winning. Now consider to select nodes a and e as seeds of the LTR process. After one time step node c will be activated by node a (since $b_{ac} = 0.8 \geq 0.4 = t_c$), but node d will not be activated by node e (since $b_{ed} = 0.3 < 0.7 = t_d$); after two time steps also node d will become active (since $b_{ed} + b_{cd} = 0.3 + 0.6 \geq 0.7 = t_d$). Thus nodes a, c, d, e will be active at the end of the process and the position of $c_\star$ in their preference lists will be shifted up according to a function of the incoming influence of each node. In particular, using Equation (4.1) with $\alpha(1) = \alpha(2) = \alpha(3) = 1$, $c_\star$ is able to move two positions up in the preference list of c and one position up in the list of voter d. The new scores of $c_1, c_2, c_\star$ are $0, 2, 3$ and the value of MoV is MoV $= 3 - 2 - (2 - 3) = 2$. We can also consider the same example using the borda count rule. The initial scores of $c_1, c_2, c_\star$ are $1, 8, 4$ and after the LTR process they become $1, 7, 7$ with a value of MoV equal to 4. Note that in this second case, even if we failed to make the target candidate win the elections, the value of MoV is at its maximum value in this instance.

In order to maximize the MoV of $c_\star$ we will focus on the score of the target candidate before and after LTR. In Sections 4.2 and 4.3 we prove that the score of the target candidate is a monotone submodular function w.r.t. the initial set of seed nodes S in any *scoring* rule; this allows us to get a $(1 - {}^{1}\!/_{e})$-approximation of the maximum score through the use of GREEDY (Algorithm 2).

The algorithm, that starts with an empty solution, iterates B times and, at each iteration, it adds to the current solution the node that gives the maximum increment

Algorithm 2 Greedy approximate Score

```
Require: Social graph $G = (V, E)$; Budget $B$; Score function $F$
S = ∅
while $|S| \leq B$ do
    $v = \arg\max_{w \in V \setminus S} F(c_\star, S \cup \{w\}) - F(c_\star, S)$
    $S = S \cup \{v\}$
return $S$
```

in the score of $c_\star$. Note that we are not able to compute the exact value of $F(c_\star, S)$ in polynomial time since we need the to know the set of active nodes at the end of the LTM process and, as proved by Chen et al. 2010, it is $\#P$-hard to compute such value. However, we are able approximate such value by sampling a polynomial number of live-edge graphs [11]. Therefore, Greedy has a computational complexity of $\mathcal{O}(B \cdot |V| \cdot |E| \cdot L)$, where L is the number of live-edge graphs needed to compute an approximation for $F(c_\star, S)$.

In Section 4.4 we exploit the result of Theorem 2.11 and show that we can achieve a constant factor approximation for the problem of maximizing the MoV. Finally, in Section 4.5, we consider the problem of *destructive control* in this setting and prove a constant factor approximation to MoV also in this case by exploiting a simple reduction that maps it to the constructive case.

4.2 Maximizing the Score: Plurality Rule

As a warm-up, in this section we focus on the *plurality rule.* We give an algorithm to select an initial set of seed nodes to maximize the expected number of nodes that will change their opinion and have $c_\star$ as first preference at the end of LTR.

Given a set of initially active nodes S, let A be the set of nodes that are active at the end of the process. An active node v with $\pi_v(c_\star) > 1$ will have $c_\star$ as first preference if $\pi_v^{\uparrow}(c_\star) = \pi_v(c_\star) - 1$, that is if and only if

$$\frac{\alpha(\pi_v(c_\star))}{t_v} \sum_{u \in A \cap N_v^-} b_{uv} \geq \pi_v(c_\star) - 1$$

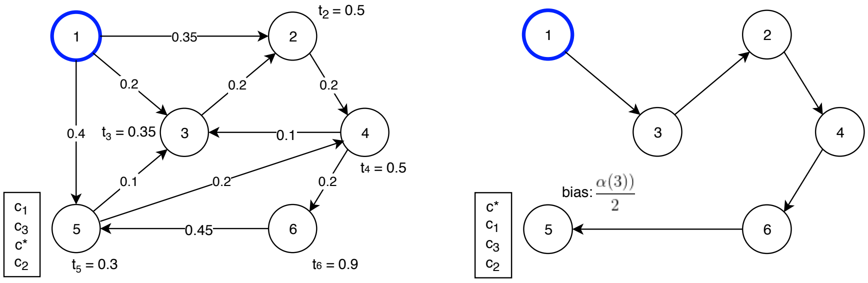
or, equivalently,

$$t_v \leq \frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star) - 1} \sum_{u \in A \cap N_v^-} b_{uv}.$$

As for the influence maximization problem, we define an alternative random process based on live-edge graphs. One possibility could be the following: For each live-edge graph evaluate which active nodes satisfy the above formula; however, in the live-edge graph process, we don't know the value of t_v since they are sampled uniformly at random at the beginning of LTM. To overcome this limitation we introduce a new process, *Live-edge Coin Flip* (LCF).

Definition 4.1. (*Live-edge Coin Flip process*)

1. Each node $v \in V$ selects at most one of its incoming edges with probability proportional to the weight of that edge, i.e., edge (u, v) is selected with probability b_{uv}, and no edge is selected with probability $1 - \sum_{u \in N_v^-} b_{uv}$.
2. Each node v with $\pi_v(c_\star) > 1$ that is reachable from S in the live-edge graph flips a biased coin and changes its list according to the outcome. This is equivalent of picking a random real number $s_v \in [0, 1]$ and setting the position of $c_\star$ according to s_v as follows: If $s_v \leq \frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star)-1}$, node v chooses $c_\star$ as its first preference (i.e., it sets $\tilde{\pi}_v(c_\star) = 1$ and shifts all the other candidates down by one position); otherwise, v maintains its original ranking.



(A) Graph $G = (V, E)$ (B) Live-edge instance $G' = (V, E')$

Figure 4.2: Example of the LCF process given a live-edge instance (B)

In the following we show that the two processes are equivalent, i.e., starting from any initial set S each node in the network has the same probability to end up with $c_\star$ in first position in both processes. This allows us to compute the function $F(c_\star, S)$, for a given S, by solving a reachability problem in graphs, as we will show later in this section.

We first prove the next Lemma which will be used to show the equivalence between the two processes and to compute $F(c_\star, S)$. The lemma shows how to compute the probability that a node v is reachable from S at the end of the LCF process by using the live-edge graphs or by using the probability of the incoming neighbors of v to be reachable from S.

We denote by $\mathcal{G}$ the set of all possible live-edge graphs sampled from G. For every $G' = (V, E') \in \mathcal{G}$ we denote by $\mathbf{P}(G')$ the probability that the live-edge graph is sampled, namely

$$\mathbf{P}(G') = \prod_{v:(u,v)\in E'} b_{uv} \prod_{v:\nexists(u,v)\in E'} \left(1 - \sum_{w:(w,v)\in E} b_{wv}\right).$$

We denote by $R(S)$ the set of nodes reachable from S at the end of the LCF process and by $R_{G'}(S)$ the set of nodes reachable from S in a fixed live-edge graph G' and by $\mathbf{1}_{(G',v)}$ the indicator function that is 1 if $v \in R_{G'}(S)$ and 0 otherwise.

The next theorem shows the equivalence between LTR and LCF.

Theorem 4.2. *Given a set of initially active nodes S, let A'_{LTR} and A'_{LCF} be the set of nodes such that $\tilde{\pi}_v(c_\star) = 1$ at the end of LTR and LCF, respectively, both starting from S. Then, for each $v \in V$, $\mathbf{P}(v \in A'_{LTR}) = \mathbf{P}(v \in A'_{LCF})$.*

Proof. We exclude from the analysis nodes v with $\pi_v(c_\star) = 1$ since they keep their original ranking in both models. We start by analyzing the LTR process. Let A be the set of active nodes at the end of the LTR process that starts from S. If U is the maximal inclusion-wise subset of active neighbors of v (i.e. $U = A \cap N_v$), then we can write the probability that $v \in A'_{\text{LTR}}$ given U, as

$$\begin{aligned}\mathbf{P}\left(v \in A'_{\text{LTR}} \mid (A \cap N_v) = U\right) &= \mathbf{P}\left(t_v \le \frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star) - 1} \sum_{u\in U} b_{uv}\right) \\ &= \frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star) - 1} \sum_{u\in U} b_{uv}.\end{aligned}$$

The overall probability that $v \in A'_{\text{LTR}}$ is

$$\mathbf{P}\left(v \in A'_{\text{LTR}}\right) = \sum_{U\subseteq N_v} \mathbf{P}\left(v \in A'_{\text{LTR}} \mid (A \cap N_v) = U\right) \cdot \mathbf{P}\left((A \cap N_v) = U\right)$$

$$= \frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star) - 1} \sum_{U \subseteq N_v} \sum_{u \in U} b_{uv} \cdot \mathbf{P}\left((A \cap N_v) = U\right).$$

Since t_v is sample from a uniform probability distribution in $[0,1]$, then

$$\sum_{u \in U} b_{uv} = \mathbf{P}\left(t_v \leq \sum_{u \in U} b_{uv}\right) = \mathbf{P}\left(v \in A \mid (A \cap N_v) = U\right).$$

Therefore, by the law of total probability,

$$\begin{aligned} \sum_{U \subseteq N_v} \sum_{u \in U} b_{uv} \cdot \mathbf{P}\left((A \cap N_v) = U\right) &= \sum_{U \subseteq N_v} \mathbf{P}\left(v \in A \mid (A \cap N_v) = U\right) \cdot \mathbf{P}\left((A \cap N_v) = U\right) \\ &= \mathbf{P}\left(v \in A\right), \end{aligned}$$

and then

$$\mathbf{P}\left(v \in A'_{\text{LTR}}\right) = \frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star) - 1} \cdot \mathbf{P}\left(v \in A\right).$$

Let us now analyze the LCF process. In order for v to be in A'_{LCF} it must hold that the coin toss has a positive outcome and that $v \in R(S)$. Thus,

$$\mathbf{P}\left(v \in A'_{\text{LCF}}\right) = \frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star) - 1} \cdot \mathbf{P}\left(v \in R(S)\right). \tag{4.2}$$

By Theorem 2.7, $\mathbf{P}\left(v \in (R(S)\right) = \mathbf{P}\left(v \in A\right)$, and hence the theorem follows. □

In Theorem 4.4 we will exploit Theorem 4.2 to show that $F(c_\star, S) - F(c_\star, \emptyset)$ is a monotone submodular function, this allows us to use Algorithm Greedy-LTR (Algorithm 2) to get a $1 - \frac{1}{e}$-approximation of the maximum score. In order to prove the theorem, we need some further notation and a lemma. For each positive integer $r \leq m$, we denote by $V^r_{c_i}$ the set of nodes that have candidate c_i in position r, and, for a graph $G' \in \mathcal{G}$, we denote by $R_{G'}(S, V^r_{c_\star})$ the subset of $V^r_{c_\star}$ of nodes reachable from a set of nodes S in G', $R_{G'}(S, V^r_{c_\star}) = \{v : v \in R_{G'}(S) \land \pi_v(c_\star) = r\}$.

The next lemma shows that $R_{G'}(S, V^r_{c_\star})$ in G' is a monotone submodular[1] function of the initial set of nodes S.

[1] For a ground set N, a function $z : 2^N \to \mathbb{R}$ is *submodular* if for any two sets S, T such that $S \subseteq T \subseteq N$ and for any element $e \in N \setminus T$ it holds that $z(S \cup \{e\}) - z(S) \geq z(T \cup \{e\}) - z(T)$.

Lemma 4.3. *Given a graph $G' \in \mathcal{G}$ and a positive integer $r \leq m$, the size of $R_{G'}(S, V^r_{c_\star})$ in G' is a monotone submodular function of the initial set of nodes S.*

Proof. The monotonicity of $|R_{G'}(S, V^r_{c_\star})|$ directly follows from the definition of reachability. We now show that $|R_{G'}(S, V^r_{c_\star})|$ is submodular. Let us consider two sets of nodes S, T such that $S \subseteq T \subseteq V$ and a node $v \in V \setminus T$. We show that $|R_{G'}(S \cup \{v\}, V^r_{c_\star})| - |R_{G'}(S, V^r_{c_\star})| \geq |R_{G'}(T \cup \{v\}, V^r_{c_\star})| - |R_{G'}(T, V^r_{c_\star})|$. Since $v \in S \cup \{v\}$, we get

$$|R_{G'}(S \cup \{v\}, V^r_{c_\star})| - |R_{G'}(S, V^r_{c_\star})| = |R_{G'}(S \cup \{v\}, V^r_{c_\star}) \setminus R_{G'}(S, V^r_{c_\star})|.$$

Moreover, for any two sets of nodes B, C we have that $R_{G'}(B \cup C, V^r_{c_\star}) = R_{G'}(B, V^r_{c_\star}) \cup R_{G'}(C, V^r_{c_\star})$. Hence

$$\begin{aligned} R_{G'}(S \cup \{v\}, V^r_{c_\star}) \setminus R_{G'}(S, V^r_{c_\star}) &= [R_{G'}(S, V^r_{c_\star}) \cup R_{G'}(\{v\}, V^r_{c_\star})] \setminus R_{G'}(S, V^r_{c_\star}) \\ &= R_{G'}(\{v\}, V^r_{c_\star}) \setminus R_{G'}(S, V^r_{c_\star}). \end{aligned}$$

Similarly,

$$|R_{G'}(T \cup \{v\}, V^r_{c_\star})| - |R_{G'}(T, V^r_{c_\star})| = |R_{G'}(\{v\}, V^r_{c_\star}) \setminus R_{G'}(T, V^r_{c_\star})|.$$

Since $S \subseteq T$, then $R_{G'}(S, V^r_{c_\star}) \subseteq R_{G'}(T, V^r_{c_\star})$ and

$$R_{G'}(\{v\}, V^r_{c_\star}) \setminus R_{G'}(S, V^r_{c_\star}) \supseteq R_{G'}(\{v\}, V^r_{c_\star}) \setminus R_{G'}(T, V^r_{c_\star}),$$

which implies the statement. □

We can now prove that (Algorithm 2) gives a constant approximation ratio.

Theorem 4.4. Greedy-LTR *(Algorithm 2) is a $\left(1 - \frac{1}{e}\right)$-approximation algorithm for the problem of maximizing the score in plurality rule voting systems.*

Proof. In the case of plurality rule, $F(c_\star, S)$ is the expected cardinality of A'_{LTR}, that is

$$F(c_\star, S) = \mathbf{E}\left[|A'_{\text{LTR}}|\right] = \sum_{v \in V} \mathbf{P}\left(v \in A'_{\text{LTR}}\right).$$

By Theorem 4.2 and Equality (4.2), this is equal to

$$\sum_{v\in V}\mathbf{P}\left(v\in A'_{\mathrm{LCF}}\right)=F\left(c_\star,\emptyset\right)+\sum_{v\in V,\,\pi_v(c_\star)>1}\frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star)-1}\cdot\mathbf{P}\left(v\in R(S)\right).$$

By Corollary 2.8, it follows that

$$F\left(c_\star,S\right)=F\left(c_\star,\emptyset\right)+\sum_{v\in V,\pi_v(c_\star)>1}\frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star)-1}\sum_{G'\in\mathcal{G}}\mathbf{P}\left(G'\right)\cdot\mathbf{1}_{(G',S,v)}.$$

We can rewrite the above formula as follows:

$$\begin{aligned}F\left(c_\star,S\right)-F\left(c_\star,\emptyset\right)&=\sum_{r=2}^{m}\sum_{v:\pi_v(c_\star)=r}\frac{\alpha(r)}{r-1}\sum_{G'\in\mathcal{G}}\mathbf{P}\left(G'\right)\cdot\mathbf{1}_{(G',S,v)}\\&=\sum_{r=2}^{m}\frac{\alpha(r)}{r-1}\sum_{G'\in\mathcal{G}}\mathbf{P}\left(G'\right)\sum_{v:\pi_v(c_\star)=r}\mathbf{1}_{(G',S,v)}\\&=\sum_{r=2}^{m}\frac{\alpha(r)}{r-1}\sum_{G'\in\mathcal{G}}\mathbf{P}\left(G'\right)\cdot\left|\{v:v\in R_{G'}(S)\wedge\pi_v(c_\star)=r\}\right|\\&=\sum_{r=2}^{m}\frac{\alpha(r)}{r-1}\sum_{G'\in\mathcal{G}}\mathbf{P}\left(G'\right)\cdot|R_{G'}(S,V^r_{c_\star})|.\end{aligned}$$

It follows that the function $F\left(c_\star,S\right)$ is a non-negative linear combination of functions $|R_{G'}(S,V^r_{c_\star})|$. By Lemma 4.3, these functions are monotone and submodular, this implies that also $F\left(c_\star,S\right)$ is monotone and submodular w.r.t. S and the same holds for $F\left(c_\star,S\right)-F\left(c_\star,\emptyset\right)$. Therefore, we can use Greedy-LTR (Algorithm 2) to find a set S whose value $F\left(c_\star,S\right)-F\left(c_\star,\emptyset\right)$ is at least $1-\frac{1}{e}$ times the optimum [25]. Moreover, we can use the same algorithm to approximate $F\left(c_\star,S\right)$ within the same approximation bound. □

4.3 Maximizing the Score: Scoring Rule

In this section we extend the results of Section 4.2 to the general case of the *scoring rule*, in which a *scoring function* f assigns a score to each candidate according to the positions he was ranked in the voters' lists. The overall approach is similar, but more general: We first define an alternative random process to LTR and show its equivalence to LTR; then we use this model to compute $F\left(c_\star,S\right)$ and show that it is a monotone submodular function of the initial set of active nodes S. This latter result allows us to compute a set S that has an approximation guarantee of $1 - {}^1\!/e$ on the maximization of the score of the target candidate.

Differently from Section 4.2, where only nodes with $c_\star$ as first preference contribute to the final score of $c_\star$, here all the voters can potentially contribute to it. Therefore in the definition of the process we consider all possible shifts of $c_\star$. The alternative random process, called *Live-edge Dice Roll* (LDR), is defined as follows.

Definition 4.5 (Live-edge Dice Roll process)**.**

1. Each node $v \in V$ selects at most one of its incoming edges with probability proportional to the weight of that edge, i.e., edge (u, v) is selected with probability b_{uv}, and no edge is selected with probability $1 - \sum_{u \in N_v^-} b_{uv}$.
2. Each node v with $\pi_v(c_\star) > 1$ that is reachable from S in the live-edge graph rolls a biased $\pi_v(c_\star)$-sided dice and changes its list according to the outcome. This is equivalent to picking a random real number s_v in $[0, 1]$ and setting the position of $c_\star$ according to s_v as follows:

$$\tilde{\pi}_v(c_\star) = \begin{cases} 1 & \text{if } s_v \leq \frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star)-1}, \\ \ell & \text{if } \frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star)-\ell+1} < s_v \leq \frac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star)-\ell}, \\ & \quad \text{for } \ell = 2, \dots, \pi_v(c_\star) - 1, \\ \pi_v(c_\star) & \text{if } s_v > \alpha(\pi_v(c_\star)). \end{cases}$$

   If $\tilde{\pi}_v(c_\star) \neq \pi_v(c_\star)$, all candidates between $\tilde{\pi}_v(c_\star)$ and $\pi_v(c_\star) - 1$ are shifted down by one position.

Note that, differently from the original live-edge process proposed by Kempe et al. 2003, LDR is able to model both the fact that a node is activated by its neighbors and the probability that the target candidate moves up in the preference list of the voter. In this case, since LDR models any arbitrarily scoring rule, we have to consider the probability that $c_\star$ can move in any position between its original and the first one.

Before extending the result of Section 4.2, we define the probability that an active node moves candidate $c_\star$ from position r to position ℓ as follows.

Definition 4.6. For each $r, \ell \in \{1, \ldots, m\}$, with $\ell \leq r$, we define:

$$\mathbf{P}\left(r, \ell\right) := \begin{cases} \dfrac{\alpha(r)}{r-1} & \text{if } \ell = 1, \\ \dfrac{\alpha(r)}{r-\ell} - \dfrac{\alpha(r)}{r-\ell+1} & \text{if } \ell = 2, \ldots, r-1, \\ 1 - \alpha(r) & \text{if } \ell = r. \end{cases}$$

In the next theorem we show that processes LTR and LDR have the same distribution.

Theorem 4.7. *Given a set of initially active nodes S and a node $v \in V$, let $\tilde{\pi}_v^{LTR}(c_\star)$ and $\tilde{\pi}_v^{LDR}(c_\star)$ be the position of node v at the end of LTR and LDR, respectively, both starting from S. Then, $\mathbf{P}\left(\tilde{\pi}_v^{LTR}(c_\star) = \ell\right) = \mathbf{P}\left(\tilde{\pi}_v^{LDR}(c_\star) = \ell\right)$, for each $\ell = 1, \ldots, \pi_v(c_\star)$.*

Proof. Let A be the set of active nodes at the end of the LTR process that starts from S. By the law of total probability, we have that

$$\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \ell\right) = \sum_{U \subseteq N_v} \mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \ell \;\middle|\; (A \cap N_v) = U\right) \mathbf{P}\left((A \cap N_v) = U\right).$$

If U is the maximal subset of active neighbors of v (i.e., $U = A \cap N_v$), then we can write the probability that $\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \ell$ given U as follows:

$$\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \ell \;\middle|\; (A \cap N_v) = U\right) =$$
$$\begin{cases} \mathbf{P}\left(t_v \leq \dfrac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star) - 1} \displaystyle\sum_{u \in U} b_{uv}\right) & \text{if } \ell = 1, \\ \mathbf{P}\left(\dfrac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star) - \ell + 1} \displaystyle\sum_{u \in U} b_{uv} < t_v \leq \dfrac{\alpha(\pi_v(c_\star))}{\pi_v(c_\star) - \ell} \displaystyle\sum_{u \in U} b_{uv}\right) & \text{if } \ell = 2, \ldots, \pi_v(c_\star) - 1, \\ \mathbf{P}\left(t_v > \alpha(\pi_v(c_\star)) \displaystyle\sum_{u \in U} b_{uv}\right) & \text{if } \ell = \pi_v(c_\star). \end{cases}$$

We analyze two cases, depending on whether $\ell < \pi_v(c_\star)$ or $\ell = \pi_v(c_\star)$.

- If $\ell < \pi_v(c_\star)$, by Definition 4.6, follows that

$$\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \ell \;\middle|\; (A \cap N_v) = U\right) = \mathbf{P}(\pi_v(c_\star), \ell) \sum_{u \in U} b_{uv}.$$

Therefore,

$$\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \ell\right) = \mathbf{P}(\pi_v(c_\star), \ell) \sum_{U \subseteq N_v} \sum_{u \in U} b_{uv}\, \mathbf{P}\left((A \cap N_v) = U\right).$$

Since $\sum_{u \in U} b_{uv} = \mathbf{P}\left(t_v \le \sum_{u \in U} b_{uv}\right) = \mathbf{P}\left(v \in A \mid (A \cap N_v) = U\right)$, then

$$\begin{aligned}&\sum_{U \subseteq N_v} \sum_{u \in U} b_{uv} \mathbf{P}\left((A \cap N_v) = U\right) \\ &= \sum_{U \subseteq N_v} \mathbf{P}\left(v \in A \mid (A \cap N_v) = U\right) \mathbf{P}\left((A \cap N_v) = U\right) = \mathbf{P}\left(v \in A\right),\end{aligned}$$

and

$$\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \ell\right) = \mathbf{P}(\pi_v(c_\star), \ell) \cdot \mathbf{P}\left(v \in A\right) \ .$$

- If $\ell = \pi_v(c_\star)$, since $\alpha(\pi_v(c_\star)) \sum_{u \in U} b_{uv} \le \sum_{u \in U} b_{uv}$, we have

$$\begin{aligned}\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \pi_v(c_\star) \;\middle|\; (A \cap N_v) = U\right) &= \mathbf{P}\left(t_v > \alpha(\pi_v(c_\star)) \sum_{u \in U} b_{uv}\right) \\ &= \mathbf{P}\left(\alpha(\pi_v(c_\star)) \sum_{u \in U} b_{uv} < t_v \le \sum_{u \in U} b_{uv}\right) + \mathbf{P}\left(t_v > \sum_{u \in U} b_{uv}\right) \\ &= \sum_{u \in U} b_{uv}\left(1 - \alpha(\pi_v(c_\star))\right) + \mathbf{P}\left(t_v > \sum_{u \in U} b_{uv}\right) \\ &= \mathbf{P}(\pi_v(c_\star), \pi_v(c_\star)) \sum_{u \in U} b_{uv} + \mathbf{P}\left(t_v > \sum_{u \in U} b_{uv}\right).\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \pi_v(c_\star)\right) &= \mathbf{P}(\pi_v(c_\star), \pi_v(c_\star)) \sum_{U \subseteq N_v} \sum_{u \in U} b_{uv}\, \mathbf{P}\left((A \cap N_v) = U\right) \\ &\quad + \sum_{U \subseteq N_v} \mathbf{P}\left(t_v > \sum_{u \in U} b_{uv}\right) \cdot \mathbf{P}\left((A \cap N_v) = U\right) \\ &= \mathbf{P}(\pi_v(c_\star), \pi_v(c_\star)) \cdot \mathbf{P}\left(v \in A\right) \\ &\quad + \sum_{U \subseteq N_v} \mathbf{P}\left(t_v > \sum_{u \in U} b_{uv}\right) \cdot \mathbf{P}\left((A \cap N_v) = U\right) \ .\end{aligned}$$

Since $\mathbf{P}\left(t_v > \sum_{u \in U} b_{uv}\right) = \mathbf{P}\left(v \notin A \mid (A \cap N_v) = U\right)$, then the second term of the above formula is equal to

$$\sum_{U \subseteq N_v} \mathbf{P}\left(v \notin A \mid (A \cap N_v) = U\right) \cdot \mathbf{P}\left((A \cap N_v) = U\right) = \mathbf{P}\left(v \notin A\right) \ ,$$

which implies

$$\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \pi_v(c_\star)\right) = \mathbf{P}(\pi_v(c_\star), \pi_v(c_\star)) \cdot \mathbf{P}\left(v \in A\right) + \mathbf{P}\left(v \notin A\right) \ .$$

To summarize, in LTR,

$$\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LTR}}(c_\star) = \ell\right) = \begin{cases} \mathbf{P}(\pi_v(c_\star), \ell) \cdot \mathbf{P}\left(v \in A\right) & \text{if } \ell < \pi_v(c_\star) \\ \mathbf{P}(\pi_v(c_\star), \pi_v(c_\star)) \cdot \mathbf{P}\left(v \in A\right) + \mathbf{P}\left(v \notin A\right) & \text{if } \ell = \pi_v(c_\star) \ . \end{cases}$$

In LDR, from Definition 4.6 follows that, for a node v reached from S, the probability that the second step of LDR yields $\tilde{\pi}_v(c_\star) = \ell$, for $\ell = 1, \ldots, \pi_v(c_\star)$, is $\mathbf{P}(\pi_v(c_\star), \ell)$. Therefore, there are two possibilities for a node to have $c_\star$ in position ℓ at the end of LDR: Either it is reached by S in the graph produced in the first step and, with probability $\mathbf{P}(\pi_v(c_\star), \ell)$, it moves $c_\star$ to position ℓ, or it is not reached by S and it already had $c_\star$ in position ℓ. In other words

$$\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LDR}}(c_\star) = \ell\right) = \begin{cases} \mathbf{P}(v \in R(S)) \cdot \mathbf{P}(\pi_v(c_\star), \ell) & \text{if } \ell < \pi_v(c_\star) \\ \mathbf{P}(v \in R(S)) \cdot \mathbf{P}(\pi_v(c_\star), \pi_v(c_\star)) + \mathbf{P}\left(v \notin R(S)\right) & \text{if } \ell = \pi_v(c_\star). \end{cases}$$

By Theorem 2.7 follows that $R(S) = A$, which implies the statement of the theorem. □

We exploit Theorem 4.7 to show that $F(c_\star, S) - F(c_\star, \emptyset)$ is a monotone submodular function and hence we can use Algorithm Greedy-LTR (Algorithm 2) to find a $\left(1 - \frac{1}{e}\right)$-approximation to the problem of maximizing the score of a target candidate.

Theorem 4.8. Greedy-LTR *(Algorithm 2) is a $\left(1 - \frac{1}{e}\right)$-approximation algorithm for the problem of maximizing the score in any scoring rule voting systems.*

Proof. By definition, the value of $F(c_\star, S) - F(c_\star, \emptyset)$ is

$$\begin{aligned}
F(c_\star, S) - F(c_\star, \emptyset) &= \mathbf{E}\left[\sum_{v \in V} f(\tilde{\pi}_v(c_\star))\right] - \sum_{v \in V} f(\pi_v(c_\star)) \\
&= \mathbf{E}\left[\sum_{v \in V} f(\tilde{\pi}_v(c_\star)) - f(\pi_v(c_\star))\right] \\
&= \sum_{v \in V} \mathbf{E}\left[f(\tilde{\pi}_v(c_\star)) - f(\pi_v(c_\star))\right] \\
&= \sum_{v \in V} \sum_{\ell=1}^{\pi_v(c_\star)} \left(f(\ell) - f(\pi_v(c_\star))\right) \mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LDR}}(c_\star) = \ell\right) \\
&= \sum_{v \in V} \sum_{\ell=1}^{\pi_v(c_\star)-1} \left(f(\ell) - f(\pi_v(c_\star))\right) \mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LDR}}(c_\star) = \ell\right),
\end{aligned}$$

where in the last equality we remove all the terms in the sum with $\ell = \pi_v(c_\star)$ since they are equal to 0.

In LDR, if $\ell < \pi_v(c_\star)$, we have that $\mathbf{P}\left(\tilde{\pi}_v^{\mathrm{LDR}}(c_\star) = \ell\right) = \mathbf{P}(v \in R(S)) \cdot \mathbf{P}(\pi_v(c_\star), \ell)$, moreover, by Corollary 2.8, $\mathbf{P}(v \in R(S)) = \sum_{G' \in \mathcal{G}} \mathbf{P}(G') \mathbf{1}_{(G',S,v)}$. Then,

$$\begin{aligned}
F(c_\star, S) - F(c_\star, \emptyset) &= \sum_{r=2}^{m} \sum_{\ell=1}^{r-1} (f(\ell) - f(r)) \mathbf{P}(r, \ell) \sum_{G' \in \mathcal{G}} \mathbf{P}\left(G'\right) \mathbf{1}_{(G',S,v)} \\
&= \sum_{r=2}^{m} \sum_{\ell=1}^{r-1} (f(\ell) - f(r)) \mathbf{P}(r, \ell) \sum_{G' \in \mathcal{G}} \mathbf{P}\left(G'\right) |\{v \in R_{G'}(S) \wedge \pi_v(c_\star) = r\}| \\
&= \sum_{r=2}^{m} \sum_{\ell=1}^{r-1} (f(\ell) - f(r)) \mathbf{P}(r, \ell) \sum_{G' \in \mathcal{G}} \mathbf{P}\left(G'\right) |R_{G'}(S, V_{c_\star}^r)|.
\end{aligned} \tag{4.3}$$

Thus, the increment in score of $c_\star$, i.e., $F(c_\star, S) - F(c_\star, \emptyset)$, is a non-negative linear combination of monotone submodular functions $|R_{G'}(S, V_{c_\star}^r)|$ (see Lemma 4.3). Thus, we can use Greedy-LTR (Algorithm 2) to find a $\left(1 - \frac{1}{e}\right)$-approximation to the problem of maximizing the score of a target candidate [25]. □

4.4 Approximating Margin of Victory

We have seen in previous sections that we can map the problem of maximizing the score of the target candidate to that of influence maximization both in the *plurality* (Section 4.2) and in an arbitrary *scoring* rules (Section 4.3); we also defined two alternative processes (Definitions 4.1 and 4.5) and showed their equivalence to LTR for both rules (Theorems 4.2 and 4.7). By showing that the objective function is monotone and submodular w.r.t. the initial set of seed nodes (Lemma 4.3) it follows that Greedy (Algorithm 2) finds a $(1-{}^{1}\!/_{e})$-approximation of the optimum [25].

Given the equivalence of the processes with LTR, we can formulate our original objective function as the average $\text{MoV}_{G'}$ computed on a sampled live-edge graph G', namely $\mathbf{E}\left[\text{MoV}(S)\right] = \mathbf{E}\left[\text{MoV}_{G'}(S)\right]$, where

$$\text{MoV}_{G'}(S) = g^{+}_{G'}\left(c_\star, S\right) + g^{-}_{G'}\left(\bar{c}, S\right) - F\left(\bar{c}, \emptyset\right) + F\left(c, \emptyset\right).$$

and $g^{+}_{G'}\left(c_\star, S\right)$, $g^{-}_{G'}\left(\bar{c}, S\right)$ are the change in margin on a fixed G' by $c_\star$ and the most voted opponent after the process. In particular in the simple case of the *plurality* rule we have that

$$g^{+}_{G'}\left(c_\star, S\right) = \sum_{r=2}^{m} \frac{\alpha(r)}{r-1} |R_{G'}(S, V^r_{c_\star})|$$
$$g^{-}_{G'}\left(\bar{c}, S\right) = \sum_{r=2}^{m} \frac{\alpha(r)}{r-1} |R_{G'}(S, V^r_{c_\star} \cap V^1_{\bar{c}})|$$

Similarly, in the general case of arbitrary *scoring* rules, we have

$$g^{+}_{G'}\left(c_\star, S\right) = \sum_{r=2}^{m} \sum_{\ell=1}^{r-1} \mathbf{P}(r,\ell)\, |R_{G'}(S, V^r_{c_\star})|\, (f(\ell) - f(r))$$
$$g^{-}_{G'}\left(\bar{c}, S\right) = \sum_{r=2}^{m} \sum_{\ell=1}^{r-1} \sum_{h=\ell}^{r-1} \mathbf{P}(r,\ell)\, |R_{G'}(S, V^r_{c_\star} \cap V^h_{\bar{c}})|\, (f(h) - f(h+1))$$

This latter formulation is just a generalization of the plurality case whenever we choose f such that $f(1) = 1$ and $f(r) = 0$, for each $r \in \{2, \ldots, m\}$. In this way we would have that the gain in score would be just 1 and that $\frac{\alpha(r)}{r-1} = \mathbf{P}\left(r, 1\right)$.

Theorem 4.9. Greedy *(Algorithm 2) is a* $\frac{1}{3}(1-{}^{1}\!/_{e})$*-approximation algorithm for the problem of election control in arbitrary scoring rule voting systems.*

Proof. It follows directly from Theorem 2.11 given that $F(c_\star, S) - F(c_\star, \emptyset)$ is a monotone and submodular function. □

4.5 Destructive Election Control

In this section we focus on the *destructive election control* problem. The model is similar to the *constructive* one: Here we define, for each node $v \in V$, the number of positions of which $c_\star$ shifts down after the LTR process as

$$\pi_v^{\downarrow}(c_\star) := \min\left(m - \pi_v(c_\star), \left\lfloor \frac{\alpha(\pi_v(c_\star))}{t_v} \sum_{u \in A,\, (u,v) \in E} b_{uv} \right\rfloor\right).$$

The final position of $c_\star$ in v will be $\tilde{\pi}_v(c_\star) := \pi_v(c_\star) + \pi_v^{\downarrow}(c_\star)$.

Similarly to the constructive case, we aim at decreasing the overall score of a target candidate $c_\star$ as much as possible since, as before, in this way we can achieve a constant factor approximation. To do that we provide a reduction from the destructive to the constructive case. Given an instance of destructive control, we build an instance of constructive control in which we simply reverse the rankings of each node and complement the scoring function to its maximum value. Roughly speaking, this reduction maintains invariant the absolute value of the change in margin of the score of any candidate between the two cases. Formally, for each $v \in V$, the new instance has a preference list defined as $\pi'_v(c) := m - \pi_v(c) + 1$ for each candidate $c \in C$, and, for each position $r \in \{1, \ldots, m\}$, has a scoring function defined as $f'(r) := f_{\max} - f(m - r + 1)$, where $f_{\max} := \max_{r \in \{1,\ldots,m\}} f(r)$. For each $v \in V$, the ranking of $c_\star$ in the new instance is $\pi'_v(c_\star) := m - \pi_v(c_\star) + 1$.

For each solution S found in the new instance, i.e., a constructive one, the overall score of $c_\star$ after the process is

$$F'(c_\star, S) := \mathbf{E}\left[\sum_{v \in V} f'(\pi'_v(c_\star) - \pi'^{\uparrow}_v(c_\star))\right],$$

where

$$\pi'^{\uparrow}_v(c_\star) := \min\left(\pi'_v(c_\star) - 1, \left\lfloor \frac{\alpha(\pi_v(c_\star))}{t_v} \sum_{u \in A,\, (u,v) \in E} b_{uv} \right\rfloor\right).$$

Let $F_D(c_\star, \emptyset) = F(c_\star, \emptyset)$ and $F'(c_\star, \emptyset) := \sum_{v \in V} f'(\pi'_v(c_\star))$. Then the following lemma holds.

Lemma 4.10. *$F_D(c_\star, \emptyset) - F_D(c_\star, S) = F'(c_\star, S) - F'(c_\star, \emptyset)$, for every S.*

Proof. Observe that $\pi_v'^{\uparrow}(c_\star) = \pi_v^{\downarrow}(c_\star)$ and that $\pi_v(c_\star) = m - \pi'_v(c_\star) + 1$. It follows that

$$\begin{aligned} F'(c_\star, S) - F'(c_\star, \emptyset) = \mathbf{E}&\left[\sum_{v \in V} [f_{\max} - f(m - (\pi'_v(c_\star) - \pi_v'^{\uparrow}(c_\star)) + 1)]\right] \\ &- \mathbf{E}\left[\sum_{v \in V} [f_{\max} - f(m - \pi'_v(c_\star) + 1)]\right] \\ &= \mathbf{E}\left[\sum_{v \in V} [f(m - \pi'_v(c_\star) + 1) - f(m - (\pi'_v(c_\star) - \pi_v'^{\uparrow}(c_\star)) + 1)]\right] \\ &= \mathbf{E}\left[\sum_{v \in V} [f(m - \pi'_v(c_\star) + 1) - f(m - \pi'_v(c_\star) + 1 + \pi_v'^{\uparrow}(c_\star))]\right] \\ &= \mathbf{E}\left[\sum_{v \in V} [f(\pi_v(c_\star)) - f(\pi_v(c_\star) + \pi_v'^{\uparrow}(c_\star))]\right] \\ &= \mathbf{E}\left[\sum_{v \in V} [f(\pi_v(c_\star)) - f(\pi_v(c_\star) + \pi_v^{\downarrow}(c_\star))]\right] = F(c_\star, \emptyset) - F_D(c_\star, S). \end{aligned}$$

□

The reduction, together with Lemma 4.10, allows us to maximize the score of the target candidate in the constructive case and then to map it back to destructive case. Differently from the *constructive* scenario, we get a factor $\frac{1}{2}$ because we can reconstruct the optimum in the approximation by only lower bounding two terms.

Theorem 4.11. Greedy *(Algorithm 2) is a $\frac{1}{2}(1 - {}^1\!/_e)$-approximation algorithm for the problem of destructive election control in arbitrary scoring rule voting systems.*

To improve readability the proof is omitted but can be found in Appendix A. However, the proof is similar to that of Theorem 2.11 and to [40, Theorem 5.3], with the additional use of Lemma 4.10.

4.6 Hardness Result

In this section we prove that both problems of maximizing score and MoV of the target candidate are *NP*-hard. These results hold for any non-increasing scoring function and any function $\alpha : \{1, \ldots, m\} \to [0, 1]$, even in elections with only two candidates. We assume that the scoring function f is non constant as otherwise the problem becomes trivial.

Theorem 4.12. *Maximizing the score and the MoV of a target candidate is NP-hard, even in the case with two candidates.*

Proof. We prove the hardness by reduction from Influence Maximization under LTM, which is known to be *NP*-hard [11]. We consider the decision version of LTM in which, given a weighted graph $G = (V, E, b)$ with weight function $b : E \to [0, 1]$ and budget B, the goal is to find a set of seeds $S \subseteq V$ such that $|S| \leq B$ and the expected cardinality of active nodes at the end of the process is greater than an input value M. Let $\mathcal{I}_{\text{LTM}} = (G, B, M)$ be an instance of Influence Maximization under LTM. We construct and instance $\mathcal{I}_{\text{LTR}} := (G', B', C, M')$ of the decision version of LTR that corresponds to $\mathcal{I}_{\text{LTM}}$. In the decision version of the problem of maximizing the score of $c_\star$ under LTR the problem asks to find a set of seeds $S \subseteq V$ such that $F(c_\star, S) \geq M'$, for some input value M'. Instance $\mathcal{I}_{\text{LTR}}$ is built as follows:

- We consider m candidates $c_\star, c_1, \ldots, c_{m-1}$; let $k \in \{1, \ldots, m-1\}$ be the smallest position such that $f(k) > f(k+1)$, where f is the non-increasing scoring function. Note that such value must exists, otherwise f would be a constant function. For every node we let $c_\star$ be in position $k+1$ while we assign arbitrary positions to the other candidates, i.e., for every node $v \in V$ we let $\pi_v(c_\star) = k+1$, and we give an arbitrary order to the other candidates.
- We consider the same graph, i.e., $G' = G$.
- We consider the same budget, i.e., $B' = B$.
- We set the fixed minimum score value that needs to be reached in $\mathcal{I}_{\text{LTR}}$ to be

$$M' = \big(f(k) - f(k+1)\big)\, \alpha(k+1)\, M + f(k+1)\, |V| \; .$$

Let S be an initial set of seed nodes and let $\sigma(S)$ be the expected number of influenced node in LTM when S is used as seed set in $\mathcal{I}_{\mathrm{LTM}}$.

We use the equivalence between LTR and LDR (Theorem 4.7) to relate expected score of $c_\star$ $F(c_\star, S)$ and $\sigma(S)$. Indeed, using S as initial set of seed nodes for $\mathcal{I}_{\mathrm{LTR}}$, by Equation (4.3), we have

$$F(c_\star, S) = \sum_{r=2}^{m} \sum_{\ell=1}^{r-1} (f(\ell) - f(r)) \mathbf{P}(r, \ell) \sum_{G' \in \mathcal{G}} \mathbf{P}(G') |R_{G'}(S, V_{c_\star}^r)| + F(c_\star, \emptyset)$$

Since in $\mathcal{I}_{\mathrm{LTR}}$ all nodes have $c_\star$ in position $k+1$, then $V_{c_\star}^{k+1} = V$, while $V_{c_\star}^r = \emptyset$, for any $r \neq k+1$. Therefore,

$$\sum_{G' \in \mathcal{G}} \mathbf{P}(G') |R_{G'}(S, V_{c_\star}^r)| = \begin{cases} \sigma(S) & \text{if } r = k+1 \\ 0 & \text{Otherwise.} \end{cases}$$

It follows that in the formula of $F(c_\star, S)$, only the term with $r = k+1$ is not 0. Moreover, $F(c_\star, \emptyset) = |V| f(k+1)$. By plugging $r = k+1$, we have that,

$$F(c_\star, S) = \sum_{\ell=1}^{k} (f(\ell) - f(k+1)) \mathbf{P}(k+1, \ell) \sigma(S) + |V| f(k+1) \ .$$

Since, $f(1) = \ldots = f(k)$, we have

$$\begin{aligned} F(c_\star, S) &= \sum_{\ell=1}^{k} (f(k) - f(k+1)) \mathbf{P}(k+1, \ell) \sigma(S) + |V| f(k+1) \\ &= (f(k) - f(k+1)) \sigma(S) \sum_{\ell=1}^{k} \mathbf{P}(k+1, \ell) + |V| f(k+1) \ . \end{aligned}$$

By Definition 4.6, we have that $\mathbf{P}(k+1, k+1) = 1 - \alpha(k+1)$ and $\sum_{\ell=1}^{k+1} \mathbf{P}(k+1, \ell) = 1$, which implies that $\sum_{\ell=1}^{k} \mathbf{P}(k+1, \ell) = \alpha(k+1)$. Therefore,

$$F(c_\star, S) = (f(k) - f(k+1)) \alpha(k+1) \sigma(S) + |V| f(k+1) \ .$$

It follows that, there exists a set of nodes S such that $F(c_\star, S) \geq M'$ only if $\sigma(S) \geq M$. The same argument can be used to show the other direction. Thus, maximizing the expected score in LTR is equivalent to maximizing the expected number of active nodes in LTM. Since influence maximization in LTM is NP-hard, then also maximizing the score in LTR is NP-hard.

Regarding the problem of maximizing the MoV we use similar arguments, and a lower bound to the MoV of M'' which we are going to define in what follows. Let us assume that an arbitrary candidate c_1 is in position 1 for all the nodes, i.e. $\pi_v(c_1) = 1$, for all $v \in V$. We analyze two cases. If $k > 1$, [2] c_1 will be the most voted opponent both before and after any diffusion process. In fact, initially it has score equal to the maximum possible, $f(1)|V|$, while, after the diffusion, it can only shift to position 2, but $f(1) = f(2) = f(k)$ and then it remains with score $f(1)|V| = f(k)|V|$. Candidate $c_\star$ initially has score $f(k+1)|V|$ and after a diffusion process starting from S it will have $F(c_\star, S) = (f(k) - f(k+1))\,\alpha(k+1)\sigma(S) + |V| f(k+1)$, as shown before. In this case the value of $\mathrm{MoV}(S)$ is

$$\begin{aligned}\mathrm{MoV}(S) &= f(k)|V| - f(k+1)|V| - (f(k)|V| - F(c_\star, S)) \\ &= (f(k) - f(k+1))\,\alpha(k+1)\sigma(S)\ ,\end{aligned}$$

and we define $M'' := (f(k) - f(k+1))\,\alpha(k+1)M$.

Let us now assume that $k = 1$. Also in this case c_1 will be the most voted opponent both before and after the diffusion process, but now the score of c_1 will change to $f(k)|V| - (f(k) - f(k+1))\,\alpha(k+1)\sigma(S)$, since any time $c_\star$ is shifted up by one position (from $k+1$ to k), c_1 is shifted down (from k to $k+1$). Therefore, the value of $\mathrm{MoV}(S)$ will be

$$\begin{aligned}\mathrm{MoV}(S) = f(k)|V| &- f(k+1)|V| \\ &- \big(f(k)|V| - (f(k) - f(k+1))\,\alpha(k+1)\sigma(S) - F(c_\star, S)\big) \\ = 2&\,(f(k) - f(k+1))\,\alpha(k+1)\sigma(S)\ ,\end{aligned}$$

and we define $M'' := 2\,(f(k) - f(k+1))\,\alpha(k+1)M$.

Note that, in both cases, there exists a set of nodes S such that $\sigma(S) \geq M$ if and only if $\mathrm{MoV}(S) \geq M''$. □

[2] We observe that this case can occur only when the number of candidates m is greater than 2.

Chapter 5

Dealing with Uncertainty in Election Control through Social Influence

In the previous chapter we show that the election control problem can be approximated within a constant ratio under several voting systems assuming a full knowledge of the preferences of each voter. However, this information is not always available since voters can be undecided or they may not want to reveal it. In this chapter we relax this assumption by considering that each voter is associated with a probability distribution over the candidates. We propose two new models in which, when a voter is reached by a campaign, it modifies its probability distribution according to the amount of influence it received from its neighbors in the network. We then study the election control problem on the new models: In the first model, under the Gap-ETH hypothesis, election control cannot be approximated within a factor better than $1/n^{o(1)}$, where n is the number of voters; in the second model the problem admits a constant factor approximation algorithm.

Most of the results presented in this chapter are included in [4, 5].

5.1 Problem Definition

In this section, we extend the previous model and consider a non-deterministic scenario in which we take into account the inherent uncertainty of a voter and we model its decision as a probabilistic function over the list of candidates. We introduce a variation of the Linear Threshold Model (LTM) [11] called *Probabilistic Linear Threshold Ranking* (PLTR). Similarly to the *Linear Threshold Ranking* (Chapter 4), PLTR takes into account the degree of influence that voters exercise on each other, but considers a probability distribution over the candidates for each voter, rather than assuming that the preference lists of the voters are known. Let $G = (V, E)$ be a directed graph, each node $v \in V$ has a probability distribution over the candidates π_v, here $\pi_v(c_i)$ denote the probability that v votes for candidate c_i; then for each $v \in V$ we have that $\pi_v(c_i) \geq 0$ for each candidate c_i and $\sum_{i=1}^{m} \pi_v(c_i) = 1$. Each node $v \in V$ has a probability distribution over the candidates π_v, where $\pi_v(c_i)$ is the probability that v votes for candidate c_i; then for each $v \in V$ we have that $\pi_v(c_i) \geq 0$ for each candidate c_i and $\sum_{i=1}^{m} \pi_v(c_i) = 1$.

For each candidate c_i, we assume that $\pi_v(c_i)$ is at least a polynomial fraction of the number of voters, i.e., $\pi_v(c_i) = \Omega(1/|V|^{\gamma})$ for some constant $\gamma > 0$.

As in LTM, each node v has a threshold $t_v \in [0, 1]$; each edge $(u, v) \in E$ has a weight b_{uv}, given in input with the graph, which models the influence of node u on v. Moreover, the total weight of the incoming edges of each node v is $\sum_{u:(u,v)\in E} b_{uv} \leq 1$. We assume that the weight of each edge (u, v) is not smaller than a polynomial fraction of the number of voters, i.e., $b_{uv} = \Omega(1/|V|^{\gamma})$ for some constant $\gamma > 0$.

Given an initial set of seed nodes S, the diffusion process proceeds as in LTM: Inactive nodes become active if the sum of the weights of incoming edges from active neighbors is greater than or equal to their threshold. In PLTR an active node increases his probability of voting for the target candidate by adding the influence coming from the active neighbors and then by normalizing to have again a probability distribution. Formally, for each node $v \in A$, where A is the set of active nodes at the end of LTM, the preference list π_v changes as follows:

$$\tilde{\pi}_v(c_\star) = \frac{\pi_v(c_\star) + \sum_{u \in A \cap N_v^-} b_{uv}}{1 + \sum_{u \in A \cap N_v^-} b_{uv}}, \tag{5.1}$$

while for any other candidate $c_i \neq c_\star$ it changes to

$$\tilde{\pi}_v(c_i) = \frac{\pi_v(c_i)}{1 + \sum_{u \in A \cap N_v^-} b_{uv}}. \tag{5.2}$$

All inactive nodes $v \in V \setminus A$ will have $\tilde{\pi}_v(c_i) = \pi_v(c_i)$ for all candidates, including $c_\star$. Let us denote by $\mathcal{G}$ the set of all possible live-edge graphs sampled from G, then, we can compute the score $F(c_i, S)$ of a candidate c_i by means of live-edge graphs used in the LTM model as

$$F(c_i, S) := \sum_{G' \in \mathcal{G}} F_{G'}(c_i, S) \cdot \mathbf{P}(G'), \tag{5.3}$$

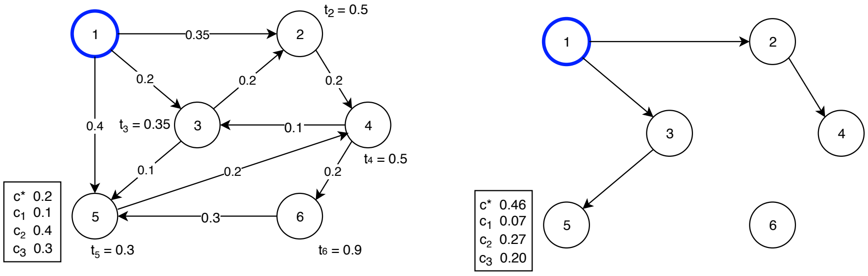
where $F_{G'}(c_i, S)$ is the score of c_i in $G' \in \mathcal{G}$ and $P(G)$ is the probability to sample the live-edge G'.

More precisely, for $c_i = c_\star$ we have

$$F_{G'}(c_\star, S) = \sum_{v \in R_{G'}(S)} \frac{\pi_v(c_\star) + \sum_{u \in R_{G'}(S) \cap N_v^-} b_{uv}}{1 + \sum_{u \in R_{G'}(S) \cap N_v^-} b_{uv}} + \sum_{v \in V \setminus R_{G'}(S)} \pi_v(c_\star),$$

where $R_{G'}(S)$ is the set of nodes reachable from S in G'. A similar formulation can be derived for $c_i \neq c_\star$.

$$F_{G'}(c_i, S) = \sum_{v \in R_{G'}(S)} \frac{\pi_v(c_i)}{1 + \sum_{u \in R_{G'}(S) \cap N_v^-} b_{uv}} + \sum_{v \in V \setminus R_{G'}(S)} \pi_v(c_i).$$



(A) Graph $G = (V, E)$ (B) Live-edge instance $G' = (V, E')$

FIGURE 5.1: Example of changing in score given a live-edge graph (B).

5.1.1 R-PLTR Model

In Section 5.3, we show that the election control problem in PLTR is hard to approximate to within a polynomial fraction of the optimum (Theorem 5.1). However, we are able to show that a small relaxation of the model allows us to approximate it to within a constant factor. In the relaxed model, that we call *Relaxed Probabilistic Linear Threshold Ranking* (R-PLTR), the probability distribution of a node is updated if it has at least an active incoming neighbor, also if the node is not active itself: Every node $v \in V$ updates its probability distribution according to Eq. (5.1) and Eq. (5.2), and not just every node $v \in A$ as in PLTR. The rationale is that a voter might slightly change its opinion about the target candidate if it receives some influence from its active incoming neighbors even if the received influence is not enough to activate it (thus making it propagate the information to its outgoing neighbors). Therefore, we include this small amount of influence in the objective function. In Section 5.3 we show that the election control problem in R-PLTR is still *NP*-hard, and in Section 5.4 we give an algorithm that guarantees a constant approximation ratio in this setting.

5.2 Influencing Voters About Other Candidates

Note that it is not always sufficient to maximize the score of the target candidate to ensure his victory, and it is easy to find counter-examples of this strategy. Moreover, in the models proposed in [40] and Chapter 4 it is sometimes convenient to increase the score of a third candidate in order to make the most voted opponent w.r.t. $c_\star$ lose score and favor $c_\star$. In the following we show that in our model the best strategy is the one that changes only the score of $c_\star$. We distinguish between three possible strategies:

- MoV_1: Influencing voters about $c_\star$.
- MoV_2: Influencing voters about $\hat{c}$, i.e., the most voted opponent w.r.t. $c_\star$ at the end of PLTR.
- MoV_3: Influencing voters about any other candidate c.

Let us now analyze the MoV of $c_\star$ in these three different cases. As described in Equation (2.4), a general formulation for MoV is the following

$$\begin{aligned}\mathrm{MoV}(S) &:= g^{+}\left(c_\star, S\right) + g^{\text{-}}\left(\hat{c}, S\right) + \Delta \\ &= F\left(c_\star, S\right) - F\left(c_\star, \emptyset\right) + F\left(\hat{c}, \emptyset\right) - F\left(\hat{c}, S\right) + \Delta,\end{aligned}$$

where S is the initial set of seed nodes and Δ is the sum of constant terms that are not modified by the process. With some algebra, it is possible to compute the MoV of $c_\star$ in such scenarios, getting the following formulations:

$$\begin{aligned}\bullet\, \mathrm{MoV}_1(S) &= \sum_{v \in A} \frac{\pi(c_\star) + \sum_{u \in A \cap N_v^-} b_{uv}}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} - \sum_{v \in A} \pi(c_\star) \\ &\quad + \sum_{v \in A} \pi(\hat{c}) - \sum_{v \in A} \frac{\pi(\hat{c})}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} + \Delta \\ &= \sum_{v \in A} \frac{(1 + \pi_v(\hat{c}) - \pi_v(c_\star)) \sum_{u \in A \cap N_v^-} b_{uv}}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} + \Delta; \\ \bullet\, \mathrm{MoV}_2(S) &= \sum_{v \in A} \frac{\pi(c_\star)}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} - \sum_{v \in A} \pi(c_\star) \\ &\quad + \sum_{v \in A} \pi(\hat{c}) - \sum_{v \in A} \frac{\pi(\hat{c}) + \sum_{u \in A \cap N_v^-} b_{uv}}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} + \Delta \\ &= \sum_{v \in A} \frac{(\pi_v(\hat{c}) - \pi_v(c_\star) - 1) \sum_{u \in A \cap N_v^-} b_{uv}}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} + \Delta; \\ \bullet\, \mathrm{MoV}_3(S) &= \sum_{v \in A} \frac{\pi(c_\star)}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} - \sum_{v \in A} \pi(c_\star) \\ &\quad + \sum_{v \in A} \pi(\hat{c}) - \sum_{v \in A} \frac{\pi(\hat{c})}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} + \Delta \\ &= \sum_{v \in A} \frac{(\pi_v(\hat{c}) - \pi_v(c_\star)) \sum_{u \in A \cap N_v^-} b_{uv}}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} + \Delta.\end{aligned}$$

We just need to observe that $\mathrm{MoV}_1(S) \geq \mathrm{MoV}_2(S)$ and that $\mathrm{MoV}_1(S) \geq \mathrm{MoV}_3(S)$ to conclude that in PLTR it is always convenient to influence the voters about the target candidate whenever you want to maximize the MoV of $c_\star$. Therefore, in the remainder of the paper we only focus at changing the score of the target candidate $c_\star$. Note that the observations above hold also for R-PLTR.

5.3 Hardness Results

In this section we provide two hardness results related to PLTR and R-PLTR. In Theorem 5.1 we show that maximizing the MoV in PLTR is at least as hard to approximate as the well-known Densest-k-Subgraph problem (up to a constant factor). Then, in Theorem 5.2 we show that maximizing the MoV in R-PLTR is still *NP*-hard. Note that the hardness result for PLTR implies several conditional hardness of approximation bounds for the election control problem. Indeed, it has been shown that the Densest-k-Subgraph problem is hard to approximate: to within any constant bound under the Unique Games with Small Set Expansion conjecture [87]; to within $n^{-1/(\log \log n)^c}$, for some constant c, under the exponential time hypothesis (ETH) [21]; to $n^{-f(n)}$ for any function $f \in o(1)$, under the Gap-ETH assumption [21].

Theorem 5.1. *An α-approximation to the election control problem in PLTR gives an $\alpha\beta$-approximation to the* Densest-k-Subgraph *problem, for a positive constant $\beta < 1$.*

Proof. [1] We prove the hardness by reduction from Densest-k-Subgraph (DkS), that is formally defined as the problem of finding the subgraph of size k with highest density. Given an instance of DkS we create a new graph in which we assign to each edge a weight equal to $\frac{1}{n^\gamma}$, for any fixed constant $\gamma \geq 4$, where n is the number of nodes in the graph. Then, for each node v we set $\pi_v(\hat{c}) = 1$ and $\pi_v(c_i) = \pi_v(c_\star) = 0$ for each $c_i \neq \hat{c}$. In this setting we have that $\mathrm{MoV}(S) = |V| - (|V| - F(c_\star, S) - F(c_\star, S)) = 2F(c_\star, S)$. Then, in order to compute the final score of $c_\star$ we average its score on a polynomial number of sampled live-edge graphs, however, in our reduction the empty live-edge graph $G'_\emptyset$ is sampled *with high probability* (i.e., $1 - \frac{1}{n^{\gamma-2}}$) and thus $\sum_{G' \neq G'_\emptyset} \mathbf{P}(G') = \mathcal{O}\left(\frac{1}{n^{\gamma-2}}\right)$. Note that in $G'_\emptyset$ the set $R_{G'_\emptyset}(S)$ is equal to S, since the graph has no edges. Therefore we can obtain that the score of $c_\star$ is $\Theta\left(\frac{\mathrm{SOL}_{DkS}(S)}{n^\gamma}\right)$ where $\mathrm{SOL}_{DkS}(S) := \sum_{v \in S} |S \cap N_v^-|$ is the number of edges of the subgraph induced by S, i.e., the value of the objective function of DkS for solution S.

Since the previous bounds hold for any set S we also have that, for two constants β_1, and β_2, $\beta_1 \frac{\mathrm{OPT}_{DkS}}{n^\gamma} \leq \mathrm{OPT} \leq \beta_2 \frac{\mathrm{OPT}_{DkS}}{n^\gamma}$, where OPT is the value of the optimal solution for PLTR and OPT_{DkS} is the value of the optimal solution for DkS. Now, if we suppose that there exists an α-approximation for PLTR we get $\mathrm{SOL}_{DkS}(S) \geq \frac{\alpha}{2}\frac{\beta_1}{\beta_2}\mathrm{OPT}_{DkS}$, i.e., the solution is an $\alpha\beta$-approximation to DkS, with $\beta := \frac{\beta_1}{2\beta_2}$. □

[1] A detailed version of the proof can be found in Appendix B

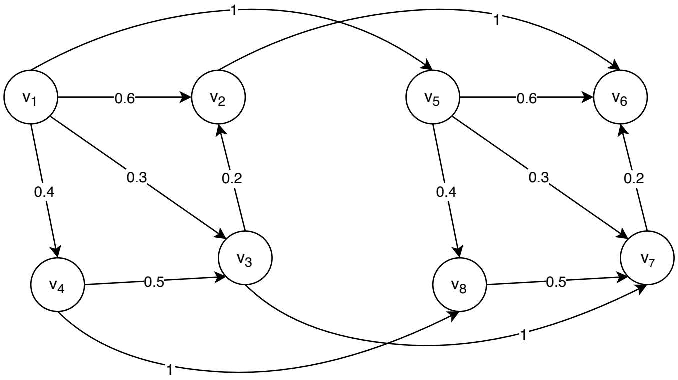


Figure 5.2: Example of reduction from LTM to R-PLTR used in Theorem 5.2 with two candidates: $\pi_{v_i}(c_\star) = 1, \pi_{v_i}(c_1) = 0\ \forall i \in [1,4]$, $\pi_{v_i}(c_\star) = 0, \pi_{v_i}(c_1) = 1\ \forall i \in [5,8]$

As a corollary of Theorem 5.1 we get the conditional hardness of approximation bounds stated at the beginning of this section.

In the next theorem we show that R-PLTR is *NP*-hard, the result is given by a reduction from Influence Maximization under LTM, which is known to be *NP*-hard [11]. The core idea is the following: Given an instance of LTM we create a new graph in which we duplicate each node and we add a new edge with weight 1 between the original node and its duplicate. Then we set $\pi_v(c_\star) = 1$ and $\pi_v(c_i) = 0$ for any original node; $\pi_v(c_\star) = 0$, $\pi_v(c_1) = 1$, and $\pi_v(c_i) = 0$ for the new nodes. Now it is easy to see that when a node becomes active in this new instance it will also activate its duplicate. Therefore MoV$(S) = |V| - |V \setminus A|$ and, thus, maximizing the MoV is equal to maximizing the active nodes in LTM.

Theorem 5.2. *Election control in R-PLTR is NP-hard.*

Proof. We prove the hardness by reduction from Influence Maximization under LTM, which is known to be *NP*-hard [11].

Consider an instance $\mathcal{I}_{\text{LTM}} = (G, B)$ of Influence Maximization under LTM. $\mathcal{I}_{\text{LTM}}$ is defined by a weighted graph $G = (V, E, b)$ with weight function $b : E \to [0,1]$ and by a budget B. Let $\mathcal{I}_{\text{R-PLTR}} := (G', B)$ be the instance that corresponds to $\mathcal{I}_{\text{LTM}}$ on R-PLTR, defined by the same budget B and by a graph $G' = (V', E', b')$ that can be built as follows:

1. Duplicate each vertex in the graph, i.e., we define the new set of nodes as $V' := V \cup \{v_{|V|+1}, \ldots, v_{2|V|}\}$.

2. Add an edge between each vertex $v \in V$ to its copy in V', i.e., we define the new set of edges as $E' := E \cup \{(v_1, v_{|V|+1}), \dots, (v_{|V|}, v_{2|V|})\}$.

3. Keep the same weight for each edge in E and we set the weights of all new edges to 1, i.e., $b'_e = b_e$ for each $e \in E$ and $b'_e = 1$ for each $e \in E' \setminus E$. Note that the constraint on incoming weights required by LTM is not violated by b'.

4. Consider m candidates $c_\star, c_1, \dots, c_{m-1}$. For each $v \in V$ we set $\pi_v(c_\star) = 1$ and $\pi_v(c_i) = 0$ for any other candidate $i \in \{1, \dots, m-1\}$. For each $v \in V' \setminus V$ we set $\pi_v(c_\star) = 0$, $\pi_v(c_1) = 1$ and $\pi_v(c_i) = 0$ for any other candidate $i \in \{2, \dots, m-1\}$.

Let S be the initial set of seed nodes of size B that maximizes $\mathcal{I}_{\text{LTM}}$ and let A be the set of active nodes at the end of the process. The value of the MoV obtained by S in $\mathcal{I}_{\text{R-PLTR}}$ is $\text{MoV}(S) = |V| - |V \setminus A|$. Indeed, each node $v \in V$ in G' has $\tilde{\pi}_v(c_\star) = \pi_v(c_\star) = 1$, because the probability of voting for the target candidate remains the same after the normalization. Moreover, each node $v_i \in V \cap A$ influences its duplicate $v_{|V|+i}$ with probability 1 and therefore $\tilde{\pi}_{v_{|V|+i}}(c_\star) = (\pi_{v_{|V|+i}}(c_\star) + 1)/2 = \frac{1}{2}$. Therefore, $F(c_\star, \emptyset) = F(c_1, \emptyset) = |V|$, $F(c_\star, S) = |V| + \frac{1}{2}|A|$, and $F(c_1, S) = |V \setminus A| + \frac{1}{2}|A|$.

Let S be the initial set of seed nodes of size B that achieves the maximum in $\mathcal{I}_{\text{R-PLTR}}$. Without loss of generality, we can assume that $S \subseteq V$, since we can replace any seed node $v_{|V|+i}$ in $V' \setminus V$ with its corresponding node v_i in V without decreasing the objective function. If A is the set of active nodes at the end of the process, then by using similar arguments as before, we can prove that $\text{MoV}(S) = |V| - |V \setminus A|$.

Note that this is the maximum that we can achieve in $\mathcal{I}_{\text{R-PLTR}}$. In fact, maximizing the MoV is equal to maximize the difference $|V| - |V \setminus A|$ that is equal to maximize the cardinality of the set A. Let us assume that S does not maximize $\mathcal{I}_{\text{LTM}}$, then, S would also not maximize $\mathcal{I}_{\text{R-PLTR}}$, which is a contradiction since S is an optimal solution for $\mathcal{I}_{\text{R-PLTR}}$.

We can prove the *NP*-hardness for the case of maximizing the score by using the same arguments. In fact, notice that maximizing the score of $c_\star$, i.e., $F(c_\star, S) = \sum_{i=1}^{2|V|} \tilde{\pi}_{v_i}(c_\star) = |V| + \frac{1}{2}|A|$, is exactly equivalent to maximize the cardinality of the active nodes in LTM. □

5.4 Approximation Results for R-PLTR

In this section we give a constant factor approximation algorithm for the election control problem in R-PLTR.

Then we show that a simple greedy hill-climbing approach gives a constant factor approximation to the problem of maximizing $g^+(c_\star, S)$, where the constant is $\frac{1}{2}(1-\frac{1}{e})$. By combining this result with Theorem 2.11, we obtain an $\frac{1}{6}(1-\frac{1}{e})$-approximation algorithm for the election control problem in R-PLTR.

In the following we show how to achieve a constant factor approximation to the problem of maximizing the MoV in R-PLTR by maximizing the increment in score of a target candidate. The core idea is to reduce the problem to an instance of the weighted version of LTM for which we are able to obtain a $(1 - {}^1\!/_e)$-approximation.

This natural extension of the LTM, presented in [11], associates to each node a non-negative weight ($w : V \to \mathbb{R}^+$) that captures the importance of activating that node. The objective function is then to find the initial seed set in order to maximize the sum of the weights of the active nodes at the end of the process, i.e., finding $\arg\max_S \sigma_w(S) = \mathbf{E}\left[\sum_{v \in A} w(v)\right]$.

A simple hill-climbing greedy algorithm achieves a constant factor approximation of $1 - {}^1\!/_e$ if the weights are polynomial in the number of nodes of the graph and the number of live-edge graph samples is polynomially large in the weights [11]. It is still an open question how well the value of $\sigma_w(S)$ can be approximated for an influence model with arbitrary node weights: Intuitively, if a node has an exponentially small probability of being sampled in the live-edge graph associated with a high weight, then a polynomial number of samples would not be enough to consider it in the solution with non-negligible probability.

We exploit this result to approximate the MoV, reducing the problem of maximizing the score to that of maximizing $\sigma_w(S)$ in the weighted LTM. We define a new graph $\hat{G}$ with the same sets of nodes and edges of G. Then, we assign a weight to each node $v \in V$ equal to $w(v) := \sum_{u \in N_v^+} b_{vu}(1 - \pi_u(c_\star))$. Note that we are able to correctly approximate the value of $\sigma_w(S)$ using such weights since the weight on each edge and the probability of not voting $c_\star$ are at least a polynomial fraction w.r.t. $|V|$, then the weight on each node in $\hat{G}$ is still bounded by a polynomial and, consequently, also the

Algorithm 3 Greedy 2

```
Require: Social graph $G = (V, E)$; Budget $B$
  $\hat{G} = (G, w)$                                          ▷ Weighted graph $\hat{G}$
  $S = \emptyset$
  while $|S| \leq B$ do
      $v = \arg\max_{u \in V \setminus S} \sigma_w(S \cup \{u\}) - \sigma_w(S)$
      $S = S \cup \{v\}$
  return $S$
```

ratio between any two weights. By applying a multiplicative form of the Chernoff bound we can get a $1 \pm \epsilon$ approximation of $\sigma_w(S)$, with high probability [11, Proposition 4.1].

Then, we can use a greedy algorithm (Algorithm 3) to maximize the influence on $\hat{G}$ by selecting at most B nodes as the seed nodes S. Note that the algorithm has the same computational complexity of the one presented in the previous chapter, i.e., $\mathcal{O}(B \cdot |V| \cdot |E| \cdot L)$, where L is the number of live-edge graphs needed to compute an approximation for $\sigma_w(S)$.

Theorem 5.3. *Algorithm 3 guarantees a $\frac{1}{6}(1 - \frac{1}{e})$-approximation factor to election control in R-PLTR.*

Proof. We first prove that Algorithm 3 achieves an $\frac{1}{2}(1-\frac{1}{e})$-approximation factor to the problem of maximizing the increment in score of the target candidate $c_\star$ in R-PLTR. Let S and $S^\star$ respectively be the set of initial seed nodes found by the greedy algorithm and the optimal one. We have that

$$\begin{aligned} g^+(c_\star, S) &= F(c_\star, S) - F(c_\star, \emptyset) \\ &= \sum_{v \in V} \frac{\pi_v(c_\star) + \sum_{u \in A \cap N_v^-} b_{uv}}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} - \sum_{v \in V} \pi_v(c_\star) \\ &= \sum_{v \in V} \frac{(1 - \pi_v(c_\star)) \sum_{u \in A \cap N_v^-} b_{uv}}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} \end{aligned}$$

and, since the denominator is at most 2, that

$$\begin{aligned} g^+(c_\star, S) &\geq \frac{1}{2} \sum_{v \in V} (1 - \pi_v(c_\star)) \sum_{u \in A \cap N_v^-} b_{uv} \\ &= \frac{1}{2} \sum_{u \in A} \sum_{v \in N_u^+} b_{uv}(1 - \pi_v(c_\star)) \end{aligned}$$

where A is the set of active nodes at the end of the process.

Note that $\sum_{u\in A}\sum_{v\in N_u^+} b_{uv}(1-\pi_v(c_\star))$ is exactly the objective function that the greedy algorithm maximizes. Hence, using the result by Kempe et al. [11], we know that

$$\sum_{u\in A}\sum_{v\in N_u^+} b_{uv}(1-\pi_v(c_\star)) \geq \left(1-\frac{1}{e}\right)\sum_{u\in A^\star}\sum_{v\in N_u^+} b_{uv}(1-\pi_v(c_\star)),$$

where $A^\star$ is the set of active nodes at the end of the process starting from $S^\star$.

Therefore $g^+\left(c_\star, S\right) \geq \frac{1}{2}(1-{}^1\!/e)\, g^+\left(c_\star, S^\star\right)$ since

$$\begin{aligned} g^+\left(c_\star, S^\star\right) &= \sum_{v\in V}\frac{(1-\pi_v(c_\star))\sum_{u\in A^\star\cap N_v^-} b_{uv}}{1+\sum_{u\in A^\star\cap N_v^-} b_{uv}} \\ &\leq \sum_{v\in V}(1-\pi_v(c_\star))\sum_{u\in A^\star\cap N_v^-} b_{uv} = \sum_{u\in A^\star}\sum_{v\in N_u^+} b_{uv}(1-\pi_v(c_\star)), \end{aligned}$$

where the inequality is due to the fact that the denominator in all the terms of $g^+\left(c_\star, S^\star\right)$ is at least 1. Thus, the greedy algorithm achieves a $\frac{1}{2}\left(1-\frac{1}{e}\right)$-approximation to the maximum increment in score.
Using Theorem 2.11 we get a $\frac{1}{6}\left(1-\frac{1}{e}\right)$-approximation ratio for the MoV. □

Chapter 6

Balancing spreads of influence in a social network

In this chapter, we investigate an optimization problem that aims at balancing information exposure in a social network when different opposing campaigns propagate in the network. That is, we assume that μ different campaigns propagate in a social network and we aim to maximize the number of people that are exposed to either ν or none of the campaigns, where $\mu \geq \nu \geq 2$. Given this possibly large number μ of viewpoints, the ν threshold parameter aims to guarantee that all influenced agents are exposed to a large enough subset of them which is hopefully more representative.

We provide dedicated approximation algorithms for both the correlated and heterogeneous settings. The heterogeneous setting corresponds to the general case in which there is no restriction on the probabilities with which the campaigns spread. Contrarily, in the correlated setting, the probability distributions for different campaigns are identical and completely correlated. We show that the problem can be approximated within a constant factor in the correlated setting for any constant values of μ and ν, instead, for the heterogeneous setting with $\nu \geq 3$, we give a reduction leading to several approximation hardness results. Maybe most importantly, we obtain that the problem cannot be approximated within a factor of $n^{-g(n)}$ for any $g(n) = o(1)$ assuming the Gap-ETH hypothesis, denoting with n the number of nodes in the social network. This complements our finding of an approximation algorithm for the heterogeneous case that for arbitrary μ and $\nu = 3$ leads to an approximation ratio of order $n^{-1/2}$.

Most of the results presented in this chapter are included in [7].

6.1 Problem Definition

Inspired by the work of Garimella et al. [1] (presented in Section 3.2), we consider several information spread processes, we also call them "campaigns", unfolding in parallel, each following the Independent Cascade model described in Section 2.3. Their problem involves two opposing viewpoints or campaigns that propagate in a social network following the independent cascade model. In this chapter we address their main open problem, by generalizing their optimization problem to a setting with arbitrarily many campaigns. This generalization is motivated by the fact that most problems raise, not only two, but a multitude of viewpoints. This can be simply explained by the complexity of these problems, but also because of the wide diversity of sensibilities present in our modern societies.

Formally, we are given a graph $G = (V, E)$ and μ probability functions $(b_i)_{i\in[\mu]}$, where each b_i is a probability function as in the Independent Cascade model described in Chapter 2, i.e., $b_i : E \to [0, 1]$.[1] For an index $i \in [\mu]$, let $X_i = (T_v)_{v\in V}$ be a possible outcome sampled using probabilities b_i, i.e., a live-edge graph. Then for a seed set $A \subseteq V$, we denote with $\rho^{(i)}_{X_i}(A)$ the set of nodes reachable from A in outcome X_i and $\sigma^{(i)}(A) = \mathrm{E}_X[|\rho^{(i)}_{X_i}(A)]$ is the expected number of nodes activated by the i'th campaign at the time of quiescence. For an arbitrary sequence $\mathcal{R} = (R_i)_{i\in[\mu]}$ of subsets of V, we define

$$\mathrm{NoSM}_{\mu,\nu}(\mathcal{R}) := \left| \left(V \setminus \bigcup_{i\in[\mu]} R_i \right) \cup \bigcup_{M\subseteq[\mu]:|M|\geq\nu} \bigcap_{i\in M} R_i \right|$$

to be the number of nodes that are contained in **N**one **or** **S**ufficiently **M**any, i.e., at least ν, of the sets in $\mathcal{R}$. The $\mathrm{NoSM}_{\mu,\nu}$-function allows us to formalize our objective function of maximizing the number of nodes that are reached by none of sufficiently many of the campaigns. Note that in the special case studied by Garimella et al. [1], the objective function can be modelled using a set difference operator. In the general case, however, such a straightforward formulation is not conceivable. Introducing the $\mathrm{NoSM}_{\mu,\nu}$-function, allows us to treat the general case nevertheless.

Let $\mathcal{X} = (X_i)_{i\in[\mu]}$ be an outcome profile by letting X_i be a possible outcome according to distribution b_i. Then, for $\mathcal{A} = (A_i)_{i\in[\mu]}$ with $A_i \subseteq V$, we denote with $\rho_{\mathcal{X}}(\mathcal{A}) = (\rho^{(i)}_{X_i}(A_i))_{i\in[\mu]}$ the set of reached nodes in the outcome $\mathcal{X}$ from seed sets $\mathcal{A}$. For two

[1] For $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, \ldots, n\}$.

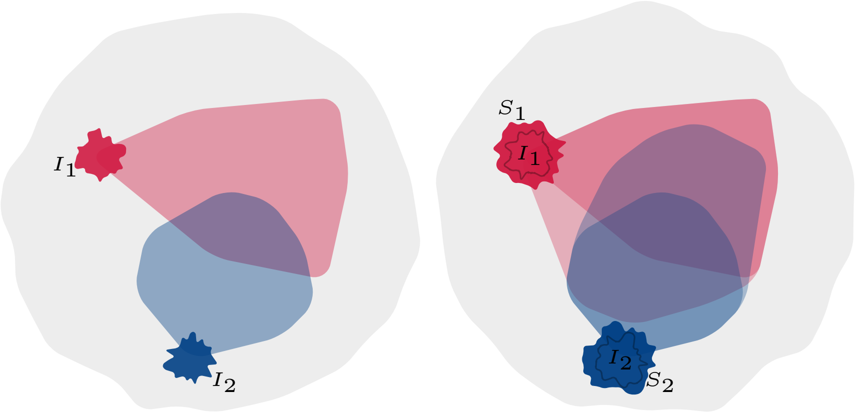


FIGURE 6.1: Example of network covered by two initial campaigns (left) and after the sets S_1 and S_2 have been added (right). Note that in the second image (right) the portion of network covered by both campaigns is greater.

sequences of sets $\mathcal{A}$, $\mathcal{A}'$, and a set A, we let $\mathcal{A} \cup \mathcal{A}' = (A_i \cup A'_i)_{i\in[\mu]}$ be the element-wise union and $\mathcal{A} \cap A = (A_i \cap A)_{i\in[\mu]}$ be the element-wise intersection with the set A.

For constant integers $\mu \geq \nu \geq 2$, we consider the following optimization problem: Given a graph $G = (V, E)$, probabilities $\mathcal{P} = (b_i)_{i\in[\mu]}$, seed sets $\mathcal{I} = (I_i)_{i\in[\mu]}$, and $B \geq 2$, the μ-ν-BALANCE problems aims at finding sets $\mathcal{S} = (S_i)_{i\in[\mu]}$ with $\sum_{i\in[\mu]} |S_i| \leq B$, such that $\Phi^{\mathcal{I}}_{\mu,\nu}(\mathcal{S})$ is maximum, where

$$\Phi^{\mathcal{I}}_{\mu,\nu}(\mathcal{S}) := \mathrm{E}_{\mathcal{X}}[\mathrm{NoSM}_{\mu,\nu}(\rho_{\mathcal{X}}(\mathcal{I} \cup \mathcal{S}))].$$

We refer to the objective function simply by $\Phi(\mathcal{S})$, in case $\mathcal{I}$, μ, and ν are clear from the context. We assume $B \leq \nu|V|$ as otherwise the problem becomes trivial by choosing $S_i = V$ for every $i \in [\nu]$. Moreover, we assume w.l.o.g. that $|V| \geq \mu$ and $B \geq \nu$, since $|V|$ and B are input parameters and μ and ν are constant numbers. Following Garimella et al. [1], we distinguish two settings. (1) The *heterogeneous* setting corresponds to the general case in which there is no restriction on $\mathcal{P}$. (2) In the *correlated* setting, the distributions b_i are identical and completely correlated for all $i \in [\mu]$. That is, if an edge (u, v) propagates a campaign to v, it propagates all campaigns that reach u to v.

6.1.1 Decomposing the Objective Function

In all of our algorithms, we use the approach of decomposing the objective function into summands and approximating the summands separately. For an outcome profile

$\mathcal{X}$, and seed sets $\mathcal{I} = (I_i)_{i\in[\mu]}$, we define $V_{\mathcal{X}}^{\ell,\mathcal{I}} \subseteq V$, for $\ell = 0,\ldots,\mu$, to be the set of nodes that are reached by exactly ℓ campaigns *from the seed sets* $\mathcal{I}$. Formally, for any value $\ell \in [\mu]$,

$$V_{\mathcal{X}}^{\ell,\mathcal{I}} := \bigcup_{\tau\in\binom{[\mu]}{\ell}} \Big(\bigcap_{i\in\tau}\rho_{X_i}^{(i)}(I_i) \setminus \bigcup_{j\in[\mu]\setminus\tau} \rho_{X_j}^{(j)}(I_j)\Big),$$

where $\binom{[\mu]}{\ell}$ denotes the set $\{\tau \subseteq [\mu] \ : \ |\tau| = \ell\}$. We write $V_{\mathcal{X}}^{\ell}$, if the initial seed sets $\mathcal{I}$ are clear from the context. In the above definition, by convention an empty union is the empty set, while an empty intersection is the whole universe, here V. Accordingly, we define $\Phi^{\ell}(\mathcal{S}) := \mathrm{E}_{\mathcal{X}}[\mathrm{NoSM}_{\mu,\nu}(\rho_{\mathcal{X}}(\mathcal{I}\cup\mathcal{S}) \cap V_{\mathcal{X}}^{\ell,\mathcal{I}})]$. Note that $\Phi^{\ell}(\mathcal{S})$ is the expected number of nodes that are reached by 0 or sufficiently many, i.e., at least ν campaigns, resulting from nodes that have been reached by exactly ℓ campaigns from $\mathcal{I}$.

Now, the objective function decomposes as

$$\begin{aligned}\Phi(\mathcal{S}) &= \mathrm{E}_{\mathcal{X}}[\mathrm{NoSM}_{\mu,\nu}(\rho_{\mathcal{X}}(\mathcal{I}\cup\mathcal{S}))]\\ &= \mathrm{E}_{\mathcal{X}}\Big[\sum_{\ell\in[\mu]} \mathrm{NoSM}_{\mu,\nu}(\rho_{\mathcal{X}}(\mathcal{I}\cup\mathcal{S}) \cap V_{\mathcal{X}}^{\ell})\Big] = \sum_{\ell\in[\mu]} \Phi^{\ell}(\mathcal{S}),\end{aligned}$$

using linearity of expectation and that sets $V_{\mathcal{X}}^{\ell}$ are disjoint. Furthermore, we denote by

$$\Phi^{\geq\ell}(\mathcal{S}) = \sum_{i=\ell}^{\mu} \Phi^{i}(\mathcal{S}) = \mathrm{E}_{\mathcal{X}}\Big[\mathrm{NoSM}_{\mu,\nu}(\rho_{\mathcal{X}}(\mathcal{I}\cup\mathcal{S}) \setminus (\cup_{j=0}^{\ell-1} V_{\mathcal{X}}^{j}))\Big].$$

Again, $\Phi^{\geq\ell}(\mathcal{S})$ denotes the expected number of nodes that are reached by sufficiently many campaigns or none of them resulting from nodes that have previously been reached by *at least* ℓ campaigns. Clearly, $\Phi(\mathcal{S}) = \Phi^{\geq 0}(\mathcal{S})$. For convenience, in what follows, we will often refer to $\mathcal{S}$ as a set of pairs in $\hat{V} := V \times [\mu]$, where picking pair (v,i) into $\mathcal{S}$ corresponds to picking v into set S_i. We fix the following observations:

- For $\ell = 0$, $\Phi^{0}(\mathcal{S})$ is optimal when $\mathcal{S} = (\emptyset)_{i\in[\mu]}$. The achieved value is the expected size of $V_{\mathcal{X}}^{0}$:

$$\Phi^{0}(\mathcal{S}) = \mathrm{E}_{\mathcal{X}}\Big[\mathrm{NoSM}_{\mu,\nu}(\rho_{\mathcal{X}}(\mathcal{I}\cup(\emptyset)_{i\in[\mu]}) \cap V_{\mathcal{X}}^{0})\Big] = \mathrm{E}_{\mathcal{X}}\left[\left|V_{\mathcal{X}}^{0}\right|\right].$$

- For $\ell = \nu - 1$, the function $\Phi^{\geq\nu-1}(\mathcal{S}) = \sum_{i=\nu-1}^{\mu} \Phi^{i}(\mathcal{S})$ is monotone and submodular, as it is equal to the number of nodes reached by a single campaign. Thus, it can be approximated within a factor of $(1 - 1/e - \epsilon)$ for any $\epsilon < 1$.

The Correlated Case. For the correlated setting, where probability functions are identical for all campaigns and the cascade processes are completely correlated, we introduce an additional function called Ψ. First note that in the correlated setting, the outcome profile $\mathcal{X}$ in the definition of $\Phi(\mathcal{S})$ satisfies $X_1 = \ldots = X_\mu$. In order to define Ψ, we introduce an additional fictitious campaign, call it campaign 0, that spreads with the same probability $p_0 = p_1 = \ldots = p_\mu$ as the other μ campaigns. We extend the outcome $\mathcal{X} = (X_i)_{i\in[\mu]}$ with $X_1 = \ldots = X_\mu$ to contain also an identical copy X_0 and define $\Psi : 2^{V\times\{0\}} \to [n]$ by

$$\Psi(\mathcal{T}) := \mathrm{E}_{\mathcal{X}}\left[\left|(\rho^{(0)}_{X_0}(\mathcal{T}) \cap \bigcup_{j=1}^{\nu-1} V^j_{\mathcal{X}}) \cup \bigcup_{j=\nu}^{\mu} V^j_{\mathcal{X}}\right|\right].$$

Observe that $\Psi(\mathcal{T})$ measures the expected number of nodes that are either (1) reached by at least one campaign from $\mathcal{I}$ and are reached by the fictitious campaign 0 from $\mathcal{T}$ or (2) reached by more than ν campaigns from $\mathcal{I}$. Note that nodes from (2) are already reached by sufficiently many campaigns while nodes from (1) have been reached by some campaign from $\mathcal{I}$ and, as witnessed by Ψ, can be reached from the nodes in $\mathcal{T}$. Note that Ψ is monotone and submodular in $\mathcal{S}$ which follows directly from the function σ having these properties.

6.1.2 A First Structural Lemma

When applying the standard greedy hill climbing algorithm to finding a set of size B maximizing a submodular set function, the key property that is used in the analysis is the following: At any stage of the algorithm there exists an element which leads to an improvement that is at least a fraction of B of the difference between the optimal and the current solution, compare for example [88, Lemma 3.13]. Maybe the most important structural lemma underlying our algorithms is a similar result for the functions $\Phi^{\geq\ell}$.

Lemma 6.1. *Let $\ell \in [1, \nu-1]$ and $\mathcal{S} \subseteq \hat{V}$ with $|\mathcal{S}| \leq B - (\nu - \ell)$ and define $U := \{\tau \subseteq \hat{V}, |\tau| = \nu - \ell\}$. Then, $\tau^* = \arg\max\{\Phi^{\geq\ell}(S \cup \tau) : \tau \in U\}$ satisfies*

$$\Phi^{\geq\ell}(\mathcal{S} \cup \tau^*) - \Phi^{\geq\ell}(\mathcal{S}) \geq \frac{\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell}) - \Phi^{\geq\ell}(\mathcal{S})}{\binom{B}{\nu-\ell}}$$

where $\mathcal{S}^*_{\geq\ell}$ is an optimal solution of size B maximizing $\Phi^{\geq\ell}$.

Proof. Let $\mathcal{X}$ be an outcome profile and let v be an arbitrary node in $V' := V \setminus \bigcup_{j=0}^{\ell-1} V_{\mathcal{X}}^j$. Let us denote by $\mathbf{1}_{\mathcal{X}}^{\mathcal{S}}(v)$ the indicator function that is one if v is reached by at least ν campaigns in outcome profile $\mathcal{X}$ from seed sets $\mathcal{I} \cup \mathcal{S}$ and zero otherwise. We note that $\Phi^{\geq \ell}(\mathcal{S}) = \mathrm{E}_{\mathcal{X}}\left[\sum_{v \in V'} \mathbf{1}_{\mathcal{X}}^{\mathcal{S}}(v)\right]$. Now, define $Y := \{\tau \subseteq \mathcal{S}_{\geq \ell}^* : |\tau| = \nu - \ell\}$, i.e., Y are the sets of nodes in $\mathcal{S}_{\geq \ell}^*$ of size $\nu - \ell$. We now argue that the following inequality holds for v and $\mathcal{X}$:

$$\mathbf{1}_{\mathcal{X}}^{\mathcal{S}_{\geq \ell}^*}(v) - \mathbf{1}_{\mathcal{X}}^{\mathcal{S}}(v) \leq \sum\nolimits_{\tau \in Y} \left(\mathbf{1}_{\mathcal{X}}^{\mathcal{S} \cup \tau}(v) - \mathbf{1}_{\mathcal{X}}^{\mathcal{S}}(v)\right). \tag{6.1}$$

If the left hand side is not positive, the inequality holds, since the right hand side cannot be negative by monotonicity. Hence, assume that the left hand side is positive. In that case it holds that $\mathbf{1}_{\mathcal{X}}^{\mathcal{S}_{\geq \ell}^*}(v) = 1$, but $\mathbf{1}_{\mathcal{X}}^{\mathcal{S}}(v) = 0$, i.e., in outcome profile $\mathcal{X}$, v is reached by at least ν campaigns from seed sets $\mathcal{I} \cup \mathcal{S}_{\geq \ell}^*$ but not from seed sets $\mathcal{I} \cup \mathcal{S}$. For such v, there must be a set $\tau \in Y$ such that adding τ to $\mathcal{S}$ results in v being reached by ν campaigns (recall that $v \in V'$ and thus v is already reached by at least ℓ campaigns). Thus, there exists a set in Y that contributes a value of 1 on the right hand side and we may conclude that Eq. (6.1) holds. Now, using linearity of expectation and Eq. (6.1), we obtain

$$\begin{aligned}\Phi^{\geq \ell}(\mathcal{S}_{\geq \ell}^*) - \Phi^{\geq \ell}(\mathcal{S}) &= \mathrm{E}_{\mathcal{X}}\Big[\sum_{v \in V'} \Big(\mathbf{1}_{\mathcal{X}}^{\mathcal{S}_{\geq \ell}^*}(v) - \mathbf{1}_{\mathcal{X}}^{\mathcal{S}}(v)\Big)\Big] \\ &\leq \mathrm{E}_{\mathcal{X}}\Big[\sum_{v \in V'} \sum_{\tau \in Y} \left(\mathbf{1}_{\mathcal{X}}^{\mathcal{S} \cup \tau}(v) - \mathbf{1}_{\mathcal{X}}^{\mathcal{S}}(v)\right)\Big].\end{aligned}$$

Using linearity of expectation again, we obtain that the right hand side above is equal to $\sum_{\tau \in Y}\left(\Phi^{\geq \ell}(\mathcal{S} \cup \tau) - \Phi^{\geq \ell}(\mathcal{S})\right)$. Then, the statement follows by the maximality of τ^* and the fact that $|Y| \leq \binom{B}{\nu - \ell}$. □

6.1.3 Approximating Ψ and $\Phi^{\geq \ell}$

As mentioned in Section 2.3, already in the standard Independent Cascade process, it is not feasible to evaluate the function σ exactly. However, σ can be approximated to within a factor of $(1 \pm \epsilon)$ by sampling a polynomial number of times. A very similar approach works for approximating the functions Ψ and $\Phi^{\geq \ell}$ for $\ell \in [0, \nu]$. That is, there is an algorithm approx$(f, \mathcal{S}, \mathcal{I}, \nu, \epsilon, \delta)$ that, for $f \in \{\Psi, \Phi^{\geq 0}, \ldots, \Phi^{\geq \nu}\}$, sets $\mathcal{S}$

and $\mathcal{I}$, and parameters ν, ϵ, δ returns a $(1 \pm \epsilon)$-approximation of $f(S)$ with probability $1 - \delta$.

We summarize this result in the following lemma, the proof of which as well as the corresponding pseudo-code implementation are deferred to the Appendix C.1 to improve readibilty. The proof relies on a Chernoff bound and is very similar to the original proof of Proposition 4.1 in [11] for the σ-function, here reported as Theorem 2.10. We also show in Appendix C.1 how to satisfy the condition $\Phi^{\geq \ell}(\mathcal{S}) \geq 1$ for every set $\mathcal{S}$ at the cost of an additive arbitrarily small ϵ in the approximation factor.

Lemma 6.2. *Let $f \in \{\Psi, \Phi^{\geq 0}, \ldots, \Phi^{\geq \nu}\}$ and let $\mathcal{S}$ be such that $f(\mathcal{S}) \geq 1$. Let $\tilde{f}(\mathcal{S}) :=$ $\mathrm{approx}(f, \mathcal{S}, \mathcal{I}, \nu, \epsilon, \delta)$ for some $0 < \delta \leq 1/2$ and $0 < \epsilon < 1$, then $\tilde{f}(\mathcal{S})$ is a $(1 \pm \epsilon)$-approximation of $f(\mathcal{S})$ with probability at least $1 - \delta$.*

All of our algorithms are of a greedy flavor, that is, we greedily choose sets in order to build the output set $\mathcal{S}$. Motivated by Lemma 6.1, we now investigate the impact of the approximation on this approach in the following lemma. Namely, how the error in approximating $f \in \{\Psi, \Phi^{\geq 1}, \ldots, \Phi^{\geq \nu}\}$ affects the error of the difference $f(S \cup v) - f(S)$. In other words, we quantify how much we lose while maximizing f by picking an element τ with respect to an approximation of f only. To this end, let f be a function from $\{\Psi, \Phi^{\geq 1}, \ldots, \Phi^{\geq \nu}\}$ and, for some $0 < \epsilon \leq 1$, let $\tilde{f}$ be a $(1 \pm \epsilon')$-approximation of f with $\epsilon' := \epsilon / (e \cdot \binom{B}{\lambda(f)})$, where $\lambda(f)$ depends on f, namely $\lambda(f) := \nu - \ell$ for $f = \Phi^{\geq \ell}$ and $\lambda(f) := 1$ for $f = \Psi$. We denote with D_f the universe over which f is defined, i.e., $D_f := \hat{V}$ for $f = \Phi^{\geq \ell}$, while $D_f := V \times \{0\}$ for $f = \Psi$.

Lemma 6.3. *Let f and $\tilde{f}$ be as above for some $0 < \epsilon \leq 1$. Let $U := \{\tau \subseteq D_f, |\tau| = \lambda(f)\}$, $\mathcal{S} \subseteq D_f$ with $|\mathcal{S}| \leq B - \lambda(f)$, and let $\mathcal{S}^*$ denote a set maximizing f of size B. Then, either*

$$f(\mathcal{S}) \geq \left(1 - \frac{1}{e}\right) \cdot f(\mathcal{S}^*) \quad \text{or} \quad f(\mathcal{S} \cup \tilde{\tau}) - f(\mathcal{S}) \geq (1 - \epsilon) \cdot (f(\mathcal{S} \cup \tau^*) - f(\mathcal{S})),$$

where $\tau^* := \arg\max\{f(S \cup \tau) : \tau \in U\}$, and $\tilde{\tau} := \arg\max\{\tilde{f}(S \cup \tau) : \tau \in U\}$.

We defer the proof to Appendix C.1. In summary: either $\mathcal{S}$ already yields a $(1 - 1/e)$-approximation of the optimum of f or a set τ of size $\lambda(f)$ maximizing an approximation $\tilde{f}$ of f can lead to a progress of at least an $(1 - \epsilon)$-fraction of the maximum progress possible.

6.1.4 Maximizing $\Phi^{\geq \nu-1}$ and Ψ

Here, we fix the result that the standard greedy hill climbing algorithm, we refer to it as Greedy$(f, \epsilon, \delta, \mathcal{I}, \nu, B)$, can be applied in order to approximate both $f \in \{\Phi^{\geq \nu-1}, \Psi\}$ to within a factor of $1 - 1/e - \epsilon$ for any $0 < \epsilon < 1$ with probability at least $1 - \delta$ for any $0 < \delta \leq 1/2$. This is based on the fact that these functions are submodular and monotone set functions. See Appendix C.1 for a pseudo-code implementation and a proof of the submodularity property. Since we can only evaluate $\Phi^{\geq \nu-1}$ and Ψ approximately, we obtain the additive ϵ-term.

Lemma 6.4. *Let $f \in \{\Phi^{\geq \nu-1}, \Psi\}$ and let $0 < \epsilon < 1$ and $0 < \delta \leq 1/2$. With probability at least $1 - \delta$, Greedy$(f, \epsilon, \delta, \mathcal{I}, \nu, B)$ returns $\mathcal{S}$ satisfying $f(\mathcal{S}) \geq (1 - 1/e - \epsilon) \cdot f(\mathcal{S}^*)$, where $\mathcal{S}^*$ is an optimal solution of size B to maximizing f.*

6.2 Hardness of Approximation for the Heterogeneous Case

In this section, we show that in the heterogeneous setting for $\nu \geq d + 1$, the μ-ν-Balance problem is as hard to approximate as the Densest-k-Sub-d-hypergraph problem [89], where $d \geq 2$ is a constant. Notably, this result has the following consequences: if $d = 2$ there is no $n^{-g(n)}$-approximation algorithm with $g(n) = o(1)$ for μ-ν-Balance. This result hold under the Gap Exponential Time Hypothesis (Gap-ETH), which states that distinguishing between a satisfiable 3-SAT formula and one which is not even $(1 - \epsilon)$-satisfiable requires exponential time for some constant $\epsilon > 0$. Gap-ETH, first formalized in [90], [91], is a stronger version of the Exponential Time Hypothesis, which only asserts that no subexponential time algorithms can decide whether a given 3-SAT formula is satisfiable. For general $d \geq 3$, we get that there is no $n^{-\epsilon}$-approximation algorithm for a given constant $\epsilon > 0$ which depends on d under the assumption that a particular class of one way functions (or that certain pseudorandom generators) exists [92].

We recall the definition of the Densest-k-Sub-d-hypergraph problem. Given a d-Regular Hypergraph $G = (V, E)$, given an integer $k \geq d$, the Densest-k-Sub-d-hypergraph problems aims at finding a set $S \subseteq V$ with $|S| \leq k$, s.t. $|E(S)|$ is maximum, where $E(S) := \{e \in E : e \subseteq S\}$.

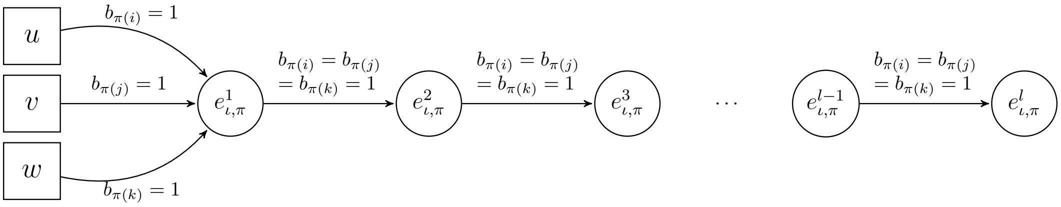


FIGURE 6.2: This figure illustrates the case $d = 3$. For an hyperedge $e = \{u, v, w\}$ in G, we get $d!\binom{\mu-\nu+d}{d}$ schemes of the above type, one for each set $\iota = \{i, j, k\} \in J$ and for each way of ordering them given by a permutation $\pi \in S_d$. Probabilities that are not given are equal to 0.

A d-regular hypergraph is a hypergraph in which all hyperedges are composed of exactly d vertices, where d is a constant. When $d = 2$, DENSEST-k-SUB-d-HYPERGRAPH is known as the DENSEST-k-SUBGRAPH problem. For the hardness of approximation proof, we consider the following transform τ of an instance $(G = (V, E), k)$ of the DENSEST-k-SUB-d-HYPERGRAPH problem into an instance $\tau(G, k) = (\overline{G} = (\overline{V}, \overline{A}), \mathcal{P}, \mathcal{I}, B)$ of the μ-ν-BALANCE problem.

- Define $\overline{V} := V_\square \cup V_\circledcirc$, where $V_\square := V$, i.e., for each node $v \in V$, we get a node v in $\overline{V}$. Moreover, let $J := \binom{[\mu-\nu+d]}{d}$, and S_d be the set of permutations of $[d]$; we then define $V_\circledcirc$ as $V_\circledcirc := \{e^t_{\iota,\pi} : e \in E, \iota \in J, \pi \in S_d, t \in [l]\}$, i.e., for each edge $e \in E$, we create λl nodes, where $l := |V| + 1$ and $\lambda := |S_d| \cdot |J| = d!\binom{\mu-\nu+d}{d}$. That is, each set ι of d campaigns in J, induces l nodes $e^t_{\iota,\pi}, t \in [l]$ for each π in S_d.

- The arc set $\overline{A}$ and the probabilities are defined as shown in Figure 6.2 illustrating the case of $d = 3$ (a more detailed illustration is provided in Appendix C.2 in Fig. C.1). We get this scheme in $\overline{G}$ for every edge $e = \{v_1, \dots, v_d\} \in E$, for each permutation π in S_d, and for each set in J of d campaigns.

- The initial seed sets $\mathcal{I}$ are defined as $I_1 = I_2 = \ldots = I_{\mu-\nu+d} = \emptyset$, $I_{\mu-\nu+d+1} = \ldots = I_\mu = \overline{V}$.

- The budget is the same as in the DENSEST-k-SUB-d-HYPERGRAPH problem, i.e., $B = k$.

Note that each node in $\overline{G}$ is already covered by $\nu - d$ campaigns and that the instance generated is deterministic, in the sense that probability values are either 0 or 1.

Let us now fix a μ-ν-Balance instance $P = (\overline{G} = (\overline{V}, \overline{A}), \mathcal{P}, \mathcal{I}, B)$ resulting from the transform τ as image of a Densest-k-Sub-d-hypergraph instance $Q = (G = (V, E), k)$. Clearly, $\overline{V}$ is of cardinality $|V| + \lambda l|E|$ and $\overline{A}$ is of cardinality $\lambda(l + d - 1)|E|$. Let us denote by Σ the set of feasible solutions for P. For each $\mathcal{S} \in \Sigma$, it holds that the objective function $\Phi(\mathcal{S})$ can be decomposed as $\Phi(\mathcal{S}) = \Phi_{\square}(\mathcal{S}) + \Phi_{\circledcirc}(\mathcal{S})$, where $\Phi_{\square}(\mathcal{S}) := \mathrm{NoSM}_{\mu,\nu}(\rho_{\mathcal{X}}(\mathcal{I} \cup \mathcal{S}) \cap V_{\square})$ and $\Phi_{\circledcirc}(\mathcal{S}) := \mathrm{NoSM}_{\mu,\nu}(\rho_{\mathcal{X}}(\mathcal{I} \cup \mathcal{S}) \cap V_{\circledcirc})$, for $\mathcal{X}$ being the only possible (deterministic) outcome profile. Now, let $\mathcal{S}^*$, $\mathcal{S}^*_{\square}$, and $\mathcal{S}^*_{\circledcirc}$ denote optimal solutions to the problem of maximizing Φ, $\Phi_{\square}$, and $\Phi_{\circledcirc}$, respectively, over Σ. The following lemma whose proof can be found in Appendix C.2 collects three statements. The first statement says that an optimal solution to Φ also maximizes $\Phi_{\circledcirc}$. The second statement says that there exists a feasible solution to P which achieves at least a multiple of $l \cdot p$ of the objective value in Densest-k-Sub-d-hypergraph with $p = d!/d^d$. In the third statement, we observe that from a feasible solution to P, we can construct a feasible solution to Q while loosing only a factor of λl in objective value.

Lemma 6.5. *The following statements hold:*

1. *An optimal solution to Φ also maximizes $\Phi_{\circledcirc}$, i.e., $\Phi_{\circledcirc}(\mathcal{S}^*_{\circledcirc}) = \Phi_{\circledcirc}(\mathcal{S}^*)$.*
2. *It holds that $\Phi_{\circledcirc}(\mathcal{S}^*_{\circledcirc}) \geq l \cdot p \cdot \mathrm{DKSH}^*_d$, where DKSH^*_d is the optimal value of* Densest-k-Sub-d-hypergraph *in Q and $p = d!/d^d$.*
3. *Given $\mathcal{S} \in \Sigma$, we can, in polynomial time, build a feasible solution S of Q such that $|E(S)| \geq \Phi_{\circledcirc}(\mathcal{S})/(\lambda l)$.*

We are now ready to show the following relations between the complexity of the two problems. Note that the assumption that g is non-increasing is w.l.o.g.

Theorem 6.6. *Let $d \geq 2$, $\nu \geq d + 1$, and $p = d!/d^d$, then we have the following two cases:*

***Case** $d = 2$**:** Let $\alpha(n) = n^{-g(n)}$ with g being non-increasing, $g(n) = o(1)$ and $\alpha(n) \in (0, 1]$ and $\beta(n) = \frac{p \cdot n^{-6g(n)}}{2\lambda}$.*

***Case** $d \geq 3$**:** Let $\alpha(n) = n^{-\epsilon(d)}$ where $\epsilon(d) > 0$ is a constant which depends on d, $\alpha(n) \in (0, 1]$ and $\beta(n) = \frac{p \cdot n^{-\epsilon'(d)}}{2\lambda}$, with $\epsilon'(d) = (d + 4) \cdot \epsilon(d)$.*

*In both cases the following statement holds: If there is an $\alpha(|\overline{V}|)$-approximate algorithm for the deterministic μ-ν-*Balance *problem, then there is a $\beta(|V|)$-approximate algorithm for* Densest-k-Sub-d-hypergraph*. Here $|\overline{V}|$ and $|V|$ denote the number of vertices in the μ-ν-*Balance *and the* Densest-k-Sub-d-hypergraph *problems, respectively and $\lambda = d!|J|$.*

Proof. Let $Q = (G, k)$ be an instance of the Densest-k-Sub-d-hypergraph problem and let $P := (\overline{G} = (\overline{V}, \overline{E}), \mathcal{P}, \mathcal{I}, B) = \tau(G, k)$ be the instance of the μ-ν-Balance problem obtained by the transform τ. For brevity, let $n := |V|$ and $\overline{n} := |\overline{V}|$. Moreover, let $\mathcal{S}$ be an $\alpha(|\overline{V}|)$-approximate solution to P, that is $\Phi(\mathcal{S}) \geq \alpha(|\overline{V}|)\Phi(\mathcal{S}^*)$. We show how to construct a $\beta(n)$-approximate solution S to Q.

Using Lemma 6.5- (3), we obtain a feasible solution S to Q with $|E(S)| \geq \Phi_{\circledcirc}(\mathcal{S})/(\lambda l)$. We proceed by lower-bounding $\Phi_{\circledcirc}(\mathcal{S})$. We can w.l.o.g. assume that $\mathcal{S} \cap V_{\circledcirc} = \emptyset$ and that $\Phi_{\circledcirc}(\mathcal{S}) \geq l$. Indeed, if $\Phi_{\circledcirc}(\mathcal{S}) < l$ then $\Phi_{\circledcirc}(\mathcal{S}) = 0$ and we can build in polynomial-time a better solution by identifying one edge $(v_1, \ldots, v_d)$ and propagating campaign i in v_i. This further implies that $\Phi_{\circledcirc}(\mathcal{S}) \geq \Phi_{\square}(\mathcal{S})$ as $l > n \geq \Phi_{\square}(\mathcal{S})$. We obtain

$$\begin{aligned}\Phi_{\circledcirc}(\mathcal{S}) &\geq \frac{\Phi(\mathcal{S})}{2} \geq \frac{\alpha(\overline{n})}{2} \cdot \Phi(\mathcal{S}^*) \geq \frac{\alpha(\overline{n})}{2} \cdot \Phi_{\circledcirc}(\mathcal{S}^*) \\ &= \frac{\alpha(\overline{n})}{2} \cdot \Phi_{\circledcirc}(\mathcal{S}^*_{\circledcirc}) \geq \frac{\alpha(\overline{n}) \cdot l \cdot p}{2} \cdot \mathrm{DKSH}^*_d,\end{aligned}$$

using Lemma 6.5- (1) and (2) in the last two steps. In summary, we have $|E(S)| \geq \frac{\alpha(\overline{n}) \cdot p}{2\lambda}\,\mathrm{DKSH}^*_d$. Note that $2\lambda/p$ is a constant.

Case $d = 2$: Since g is non-increasing, we get

$$\alpha(\overline{n}) = \overline{n}^{-g(\overline{n})} = 2^{-g(\overline{n})\log(\overline{n})} \geq 2^{-g(n)\log(2\lambda n^3)} \geq 2^{-6g(n)\log(n)} = n^{-6g(n)},$$

where we used $2 \leq n \leq \overline{n} \leq 2\lambda n^3$ and $\lambda \leq \mu^2 \leq n^2$ (as $d = 2$). This completes this case.

Case $d \geq 3$: In this case $\alpha(n) = n^{-\epsilon(d)}$. It follows that

$$\alpha(\overline{n}) = \overline{n}^{-\epsilon(d)} = 2^{-\epsilon(d)\log(\overline{n})} \geq 2^{-\epsilon(d)\log(2\lambda n^3)} \geq 2^{-(d+4)\epsilon(d)\log(n)} = n^{-(d+4)\epsilon(d)},$$

where we used $2 \leq n \leq \overline{n} \leq 2\lambda n^3$ and $\lambda \leq \mu^d \leq n^d$. This completes this case. □

To sum up, our reduction shows that: (1) as Densest-k-Sub-d-hypergraph cannot be approximated within $1/n^{\epsilon}$ for some constant $\epsilon > 0$ which depends on d, if a particular class of one way functions exists [92], we have shown that the same hardness result holds for any μ-ν-Balance problem with $\nu \geq d + 1 \geq 4$; (2) moreover as Densest-k-Subgraph cannot be approximated within $1/n^{o(1)}$, if the Gap-ETH holds [21], we have shown that the same hardness result holds for any μ-ν-Balance problem with $\nu \geq 3$.

Other approximation hardness results exist for Densest-k-Subgraph. We review them here, highlighting the hardness results that our reduction implies in each case.

- Densest-k-Subgraph cannot be approximated within any constant, if the Unique Games with Small Set Expansion (UGSSE) conjecture holds [87]. Therefore, under the UGSSE conjecture it is easy to prove that the reduction given above shows that any μ-ν-Balance problem with $\nu \geq 3$ cannot be approximated within any constant.
- Densest-k-Subgraph cannot be approximated within $n^{-(\log \log n)^{-c}}$, for some constant c if the exponential time hypothesis holds [21]. Under the same conjecture, our reduction implies the same hardness result for any μ-ν-Balance problem with $\nu \geq 3$.

6.3 Approximation Algorithm for the Heterogeneous Case

Our approach for maximizing $\Phi(\mathcal{S})$ is to decompose it as $\Phi(\mathcal{S}) = \Phi^0(\mathcal{S}) + \Phi^{\geq 1}(\mathcal{S})$ and work on each summand separately. In this section, we give two different algorithms GreedyTuple and GreedyIter for maximizing $\Phi^{\geq 1}(\mathcal{S})$. These two algorithms are inspired by a similar approach for the so-called maximum coverage with pairs problem [93].

6.3.1 Greedily Picking Tuples

In this paragraph, we present GreedyTuple$(\epsilon, \delta, \ell, \mathcal{I}, \nu, B)$ (Algorithm 4) that, for given ℓ, computes a solution to maximizing $\Phi^{\geq \ell}$. For $\ell = \nu - 1$ the algorithm is identical to the standard greedy hill climbing algorithm. For the general case of $\ell \leq \nu - 1$, we will

Algorithm 4 GreedyTuple($\epsilon, \delta, \ell, \mathcal{I}, \nu, B$)

```
$t := \lceil \frac{B}{\nu-\ell} \rceil \binom{|\hat{V}|}{\nu-\ell}$
$\delta' \leftarrow \delta/t$
$\epsilon' \leftarrow \epsilon/(2e \cdot \binom{B}{\nu-\ell})$
$\mathcal{S} \leftarrow \emptyset$
while $|\mathcal{S}| \leq B - (\nu - \ell)$ do
    Compute $\tau \leftarrow \arg\max_{\tau \subseteq \hat{V}, |\tau| = \nu-\ell}\{\text{approx}(\Phi^{\geq \ell}, \mathcal{S} \cup \tau, \mathcal{I}, \nu, \epsilon', \delta')\}$
return $\mathcal{S}$
```

show the following theorem. The algorithm is inspired by a greedy algorithm, called Greedy1, due to [93] for solving the so-called maximum coverage with pairs problem.

Theorem 6.7. *Let $\epsilon \in (0, 1)$, $\delta \leq 1/2$, and $\ell \in [1, \nu - 1]$. If $B \geq 2\nu/\epsilon$, with probability at least $1 - \delta$, the algorithm* GreedyTuple($\epsilon, \delta, \ell, \mathcal{I}, \nu, B$) *returns a solution $\mathcal{S}$ satisfying*

$$\Phi^{\geq \ell}(\mathcal{S}) \geq \frac{1 - 1/e - \epsilon}{\binom{B-1}{\nu-\ell-1}} \cdot \Phi^{\geq \ell}(\mathcal{S}^*_{\geq \ell}),$$

where $\mathcal{S}^*_{\geq \ell}$ is an optimal solution to $\Phi^{\geq \ell}$.

We let $\mathcal{S}^i$ denote the set $\mathcal{S}$ at the end of iteration i of the algorithm. The main idea underlying the analysis of GreedyTuple is very much related to the analysis of the standard greedy algorithm. That is (ignoring the approximation issue), every step of the algorithm incurs a factor of $1 - \left(1 - 1/\binom{B}{\nu-\ell}\right)$. For $\ell = \nu - 1$, this coincides with the standard case.

Lemma 6.8. *Let $0 < \epsilon < 1$, $\delta \leq 1/2$, and $\ell \in [1, \nu - 1]$. With probability at least $1 - \delta$, after each iteration i of Algorithm 4, it either holds that*

$$\Phi^{\geq \ell}(\mathcal{S}^i) \geq \left(1 - \left(1 - \frac{1 - \frac{\epsilon}{2}}{\binom{B}{\nu-\ell}}\right)^i\right) \cdot \Phi^{\geq \ell}(\mathcal{S}^*_{\geq \ell}) \quad \text{or} \quad \Phi^{\geq \ell}(\mathcal{S}^i) \geq \left(1 - \frac{1}{e}\right) \cdot \Phi^{\geq \ell}(\mathcal{S}^*_{\geq \ell}).$$

The proof of this lemma can be found in Appendix C.3, it uses Lemmata 6.1 and 6.3. We are now ready to give the proof of Theorem 6.7.

Proof of Theorem 6.7. Let $\mathcal{S}$ denote the set returned by the algorithm. Clearly we have that $\Phi^{\geq \ell}(\mathcal{S}) \geq \Phi^{\geq \ell}(\mathcal{S}^\iota)$, where ι denotes the number of iterations of the while loop in the algorithm. By assumption $B \geq 2\nu/\epsilon$ and thus $\iota = \lfloor \frac{B}{\nu-\ell} \rfloor \geq \frac{B}{\nu-\ell} - 1 \geq (1 - \frac{\epsilon}{2}) \cdot \frac{B}{\nu-\ell}$.

Using Lemma 6.8 for $\mathcal{S}^\iota$ yields that either $\Phi^{\geq\ell}(\mathcal{S}^\iota) \geq (1-{}^{1}\!/_{e})\cdot\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell})$ or

$$\begin{aligned}\Phi^{\geq\ell}(\mathcal{S}^\iota) &\geq \Big(1-\Big(1-\frac{1-\frac{\epsilon}{2}}{\binom{B}{\nu-\ell}}\Big)^{\iota}\Big)\cdot\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell})\\ &\geq \Big(1-\Big(1-\frac{1-\frac{\epsilon}{2}}{\binom{B}{\nu-\ell}}\Big)^{(1-\frac{\epsilon}{2})\frac{B}{\nu-\ell}}\Big)\cdot\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell}).\end{aligned}$$

For the former case, note that $1-{}^{1}\!/_{e}$ is greater than the approximation factor required by the theorem. For the latter case note that, as $1-x\leq\exp(-x)$ for any real x, we have

$$1-\Big(1-\frac{1-\frac{\epsilon}{2}}{\binom{B}{\nu-\ell}}\Big)^{(1-\frac{\epsilon}{2})\frac{B}{\nu-\ell}} \geq 1-\exp\Big(\frac{-(1-\frac{\epsilon}{2})^2}{\binom{B-1}{\nu-\ell-1}}\Big) \geq \frac{1-\frac{1}{e}-\epsilon}{\binom{B-1}{\nu-\ell-1}},$$

where the last inequality uses that $1-\exp(-x)\leq x\cdot(1-\exp(-1))$ and $(1-{}^{1}\!/_{e})(1-x)\leq 1-{}^{1}\!/_{e}-x$ for any $x\geq 0$. This completes the proof. □

6.3.2 Being Iteratively Greedy

Recall that, at the beginning of this section, we have defined

$$\Phi^{\geq\ell}(\mathcal{S}) := \mathrm{E}_{\mathcal{X}}\Big[\mathrm{NoSM}_{\mu,\nu}(\rho_{\mathcal{X}}(\mathcal{I}\cup\mathcal{S})\setminus(\cup_{j=0}^{\ell-1}V_{\mathcal{X}}^{j,\mathcal{I}}))\Big].$$

We now extend this notation by letting

$$\Phi^{\geq\ell}_{\beta}(\mathcal{R},\mathcal{S}) := \mathrm{E}_{\mathcal{X}}\Big[\mathrm{NoSM}_{\mu,\beta}(\rho_{\mathcal{X}}(\mathcal{R}\cup\mathcal{S})\setminus\bigcup_{j=0}^{\ell-1}V_{\mathcal{X}}^{j,\mathcal{R}})\Big]$$

where $\ell\in[\nu-1]$ and $\beta\in[\nu]$; we will mainly be working with the case $\beta=\ell+1$. The function measures the expected number of nodes that are reached by at least β campaigns from $\mathcal{R}\cup\mathcal{S}$ within the set of nodes that have originally been reached by at least ℓ campaigns from $\mathcal{R}$. Our goal now is to maximize $\Phi(\cdot)$ through the following iterative scheme: for ℓ from 1 to $\nu-1$, we find sets $\mathcal{S}^{[\ell]}$ of size $\lfloor k/(\nu-1)\rfloor$ maximizing $\Phi^{\geq\ell}_{\ell+1}(\mathcal{R}^{[\ell]},\cdot)$, where $\mathcal{R}^{[\ell]} := \mathcal{I}\cup\bigcup_{j=1}^{\ell-1}\mathcal{R}^{[j]}$. That is, in the ℓ^{th} iteration, we maximize the number of nodes reached by $\ell+1$ campaigns that have previously been reached by at least ℓ campaigns. The approach is motivated by the observation that, for any $\ell\in[\nu-1]$ and initial sets $\mathcal{R}$, the function $\Phi^{\geq\ell}_{\ell+1}(\mathcal{R},\mathcal{S})$ is monotone and submodular in $\mathcal{S}$, compare with Section 6.1.4 where we used this fact for $\ell=\nu-1$. Using Lemma 6.4 applied to $\Phi^{\geq\ell}_{\ell+1}(\mathcal{R},\cdot)$ with $\nu=\ell+1$ we get that the standard greedy algorithm can be used in

Algorithm 5 GreedyIter$(\epsilon, \delta, \mathcal{I}, \nu, B)$

```
$\delta' \leftarrow \delta/\nu$
$\epsilon' \leftarrow \epsilon/2$
$\mathcal{R}^{[1]} \leftarrow \mathcal{I}$
**for** $\ell = 1, \ldots, \nu - 1$ **do**
    $\mathcal{S}^{[\ell]} \leftarrow$ Greedy$(\Phi^{\geq \ell}_{\ell+1}(\mathcal{R}^{[\ell]}, \cdot), \epsilon', \delta', \mathcal{R}^{[\ell]}, \ell + 1, \lfloor B/(\nu - 1) \rfloor)$
    $\mathcal{R}^{[\ell+1]} \leftarrow \mathcal{R}^{[\ell]} \cup \mathcal{S}^{[\ell]}$
**return** $\bigcup_{i=1}^{\nu-1} \mathcal{S}^{[i]}$
```

order to obtain a $(1 - 1/e - \epsilon)$-approximate solution when maximizing $\Phi^{\geq \ell}_{\ell+1}(\mathcal{R}, \cdot)$. Note that our algorithm, called GreedyIter(Algorithm 5) is inspired by a similar greedy algorithm called Greedy2 from [93] that is used there for the maximum coverage with pairs problem. We will prove the following theorem in this section.

Theorem 6.9. *Let $0 < \epsilon < 1$ and $\delta \leq 1/2$. With probability $1 - \delta$, the algorithm* GreedyIter$(\epsilon, \delta, \mathcal{I}, \nu, B)$ *returns $\mathcal{S}$ satisfying*

$$\Phi(\mathcal{S}) \geq \frac{(1 - \frac{1}{e} - \epsilon)^{\nu-1}}{\nu^{2\nu-3}} \left(\frac{B}{2|V|} \right)^{\nu-2} \cdot \Phi^{\geq 1}(\mathcal{I}, \mathcal{S}^*_{\geq 1}),$$

where $\mathcal{S}^*_{\geq 1}$ is a set of cardinality B maximizing $\Phi^{\geq 1}(\mathcal{I}, \cdot)$.

The proof of Theorem 6.9 relies on the following two lemmata whose proofs are given in the appendix, see Section C.3. In a sense the first lemma quantifies the loss in approximation of the first iteration of GreedyIter, while the second lemma quantifies the loss of the later iterations. Both proofs rely on the submodularity of $\Phi^{\geq \ell}_{\ell+1}(\mathcal{R}, \cdot)$.

Lemma 6.10. *Let $\epsilon > 0$ and assume that $B \geq 2(\nu - 1)/\epsilon$. If $\mathcal{S}^{[1]} \subseteq \hat{V}$ is the set of cardinality $\lfloor B/(\nu - 1) \rfloor$ selected in the first iteration of* GreedyIter$(\epsilon, \delta, \mathcal{I}, \nu, B)$*, then, with probability at least $1 - \delta/\nu$, it holds that*

$$\Phi^{\geq 1}_{2}(\mathcal{I}, \mathcal{S}^{[1]}) \geq \frac{1 - \frac{1}{e} - \epsilon}{\nu} \cdot \Phi^{\geq 1}(\mathcal{I}, \mathcal{S}^*_{\geq 1}),$$

where $\mathcal{S}^*_{\geq 1}$ is a set of cardinality B maximizing $\Phi^{\geq 1}(\mathcal{I}, \cdot)$.

Lemma 6.11. *Let $\epsilon > 0$, $\ell \geq 2$ and assume that $B \geq 2(\nu - 1)/\epsilon$. If $\mathcal{S}^{[\ell]} \subseteq \hat{V}$ is the set of cardinality $\lfloor B/(\nu - 1) \rfloor$ selected in the ℓ'th iteration of* GreedyIter$(\epsilon, \delta, \mathcal{I}, \nu, B)$*,* *then, with probability at least* $1 - \delta/\nu$, *it holds that*

$$\Phi_{\ell+1}^{\geq \ell}(\mathcal{R}^{[\ell]}, \mathcal{S}^{[\ell]}) \geq \frac{(1 - \frac{1}{e} - \epsilon)B}{2(\ell+1)(\nu-1)|V|} \cdot \Phi_{\ell}^{\geq \ell-1}(\mathcal{R}^{[\ell-1]}, \mathcal{S}^{[\ell-1]}).$$

Proof of Theorem 6.9. Since $\mathcal{S} = \bigcup_{i=1}^{\nu-1} \mathcal{S}^{[i]}$, we obtain $\Phi(\mathcal{S}) \geq \Phi_{\nu}^{\geq \nu-1}(\mathcal{R}^{[\nu-1]}, \mathcal{S}^{\nu-1})$. Using the union bound, $\nu - 2$ times Lemma 6.11 and then Lemma 6.10 yield that, with probability at least $1 - \delta$, it holds that

$$\begin{aligned}\Phi(\mathcal{S}) &\geq \Big(\frac{(1 - \frac{1}{e} - \epsilon)B}{2\nu(\nu-1)|V|}\Big)^{\nu-2} \Phi_{2}^{\geq 1}(\mathcal{R}^{[1]}, \mathcal{S}^{[1]}) \\ &\geq \frac{(1 - \frac{1}{e} - \epsilon)^{\nu-1}}{\nu^{2\nu-3}} \Big(\frac{B}{2|V|}\Big)^{\nu-2} \Phi_{\nu}^{\geq 1}(\mathcal{I}, \mathcal{S}_{\geq 1}^{*}).\end{aligned}$$

□

6.3.3 Algorithm for General Heterogeneous μ-ν-Balance problem

Our approach to solving the general μ-ν-Balance problem is now to use both algorithms presented above.

The complete approximation algorithm works as follows: Using the algorithm GreedyTuple($\epsilon, \delta/2, 1, \mathcal{I}, \nu, B$), we obtain a set $\mathcal{S}^1$ that with probability $1 - \delta/2$ satisfies $\Phi^{\geq 1}(\mathcal{S}^1) \geq \alpha_1 \cdot \Phi^{\geq 1}(\mathcal{S}_{\geq 1}^{*})$, where $\mathcal{S}_{\geq 1}^{*}$ denotes an optimal solution of size B to maximizing $\Phi^{\geq 1}$ and α_1 is the approximation factor that will be achieved by GreedyTuple, see Theorem 6.7. On the other hand GreedyIter($\epsilon, \delta/2, \mathcal{I}, \nu, B$) outputs a set $\mathcal{S}^2$ that with probability $1 - \delta/2$ satisfies $\Phi(\mathcal{S}^2) \geq \alpha_2 \cdot \Phi^{\geq 1}(\mathcal{S}_{\geq 1}^{*})$, where α_2 is as in Theorem 6.9.

Now, we define $\mathcal{S}'$ to be the solution that achieves the maximum $\max\{\Phi(\mathcal{S}^1), \Phi(\mathcal{S}^2)\}$ and $\mathcal{S}$ to be the solution that achieves the maximum $\max\{\Phi(\emptyset), \Phi(\mathcal{S}')\}$. It follows that

$$2\Phi(\mathcal{S}) \geq \Phi(\emptyset) + \Phi(\mathcal{S}') \geq \Phi^{0}(\emptyset) + \sqrt{\alpha_1 \cdot \alpha_2}\Phi^{\geq 1}(\mathcal{S}_{\geq 1}^{*}),$$

using that the maximum $\Phi(\mathcal{S}')$ is lower bounded by the geometric mean of $\Phi(\mathcal{S}^1)$ and $\Phi(\mathcal{S}^2)$, which are in turn lower bounded by $\alpha_1 \cdot \Phi^{\geq 1}(\mathcal{S}_{\geq 1}^{*})$ and $\alpha_2 \cdot \Phi^{\geq 1}(\mathcal{S}_{\geq 1}^{*})$, respectively. Now, let $\mathcal{S}^*$ be an optimal solution of size k to maximizing Φ. Using that the empty set maximizes Φ^0, we have $\Phi^0(\emptyset) \geq \Phi^0(\mathcal{S}^*)$. Furthermore $\Phi^{\geq 1}(\mathcal{S}^*_{\geq 1}) \geq \Phi^{\geq 1}(\mathcal{S}^*)$, thus

$$\Phi(\mathcal{S}) \geq \frac{\sqrt{\alpha_1 \alpha_2}}{2}(\Phi^0(\mathcal{S}^*) + \Phi^{\geq 1}(\mathcal{S}^*)) = \frac{\sqrt{\alpha_1 \alpha_2}}{2}\Phi(\mathcal{S}^*).$$

Plugging in α_1 and α_2, see Theorems 6.7 and 6.9, and using $B^{\nu-2} \geq \binom{B-1}{\nu-2}$, we get the following theorem.

Theorem 6.12. *Let $0 < \epsilon < 1$ and $\delta \leq 1/2$. There is an algorithm that, with probability $1-\delta$, outputs a solution $\mathcal{S}$ that satisfies $\Phi(\mathcal{S}) \geq (1-\frac{1}{e}-\epsilon)^{\frac{\nu}{2}}(\frac{1}{2|V|})^{\frac{\nu-2}{2}}\nu^{-\frac{2\nu-3}{2}} \cdot \Phi(\mathcal{S}^*)$, where $\mathcal{S}^*$ denotes a solution of size B maximizing $\Phi(\cdot)$.*

We remark that this result is mostly interesting for the case where $\nu = 3$. Indeed, in this case we obtain an algorithm with an approximation ratio of order $n^{-1/2}$.

6.4 Approximation for the Correlated Case

We now turn to the correlated case. Recall that here the probability functions are identical for all campaigns, i.e., $p_1(e) = \ldots = p_\mu(e)$ for every edge $e \in E$. Moreover, the cascade processes are completely correlated, that is, for any edge (u, v), if node u propagates campaign i to v, then node u also propagates all other campaigns that reach it to v.

We will consider the same decomposition of the objective function as in the heterogeneous case, i.e., $\Phi(\mathcal{S})$ as $\Phi(\mathcal{S}) = \Phi^{\geq 1}(\mathcal{S}) + \Phi^0(\mathcal{S})$ for a solution $\mathcal{S}$. Recall that $\Phi^{\geq 1}(\mathcal{S})$ counts the number of nodes that are reached by sufficiently many, i.e. ν, campaigns from $\mathcal{I} \cup \mathcal{S}$ and have been reached by at least one campaign from $\mathcal{I}$. Similarly, $\Phi^0(\mathcal{S})$ counts nodes that are reached by sufficiently many campaigns or none and have previously been reached by no campaign from $\mathcal{I}$. Clearly, as in the heterogeneous case, $\Phi^0(\mathcal{S})$ is optimal when $\mathcal{S} = \emptyset$. Differently from the heterogeneous case however, we will see that in the correlated setting, there is an approximation algorithm for $\Phi^{\geq 1}$ that achieves a constant factor, namely, $(1 - 1/e - \epsilon)/(\nu + 1)$. The idea is to pick ν campaigns and propagate them in the same $\lfloor B/\nu \rfloor$ nodes, exploiting that all campaigns spread in an identical manner.

To that end, we consider the problem of maximizing influence spread with one fictitious campaign, say campaign 0, spreading with the same probabilities as the others. We will consider the nodes reached by campaign 0 among the nodes that were (a) reached by at least one campaign from $\mathcal{I}$ and were (b) reached by no more than ν campaigns from $\mathcal{I}$. For this purpose, we had defined the function Ψ in Section 6.1. Recall $\Psi(\mathcal{T}) := \mathrm{E}_{\mathcal{X}}[|(\rho^{(0)}_{X_0}(\mathcal{T}) \cap \bigcup_{j=1}^{\nu-1} V^j_{\mathcal{X}}) \cup \bigcup_{j=\nu}^{\mu} V^j_{\mathcal{X}}|]$ and observe that $\Psi(\mathcal{T})$ measures the expected number of nodes that are either (1) reached by more than ν campaigns from $\mathcal{I}$ or (2) are reached by campaign 0 from $\mathcal{T}$ and were reached by at least one campaign from $\mathcal{I}$. Recall that we had shown that Ψ is submodular and that the greedy hill-climbing algorithm leads to an approximation factor of at least $1 - 1/e - \epsilon$ for any $\epsilon > 0$ when applied to maximizing Ψ. The following lemma whose proof can be found in Appendix C.4 collects three statements. The first statement relates the optimum of Ψ to the optimum of $\Phi^{\geq 1}$. The second statement says that we lose a factor of roughly ν when choosing a set of size $\lfloor B/\nu \rfloor$ instead of B when maximizing Ψ (this is due to submodularity). The last statement shows that a certain solution $\mathcal{S}'$ for $\Phi^{\geq 1}$ constructed from a solution $\mathcal{T}$ to Ψ achieves the same value.

Lemma 6.13. *The following statements hold:*

1. *If $\mathcal{T}^B \subseteq V \times \{0\}$ is a solution of size B maximizing Ψ and $\mathcal{S}^B \subseteq \hat{V}$ is a solution of size B maximizing $\Phi^{\geq 1}$, then $\Psi(\mathcal{T}^B) \geq \Phi^{\geq 1}(\mathcal{S}^B)$.*
2. *Let $\epsilon > 0$ and $B \geq \nu/\epsilon$. If $\mathcal{T}^B \subseteq V \times \{0\}$ is a solution of size B maximizing Ψ and $\mathcal{T}^{\lfloor B/\nu \rfloor} \subseteq V \times \{0\}$ is a solution of size $\lfloor B/\nu \rfloor$ maximizing Ψ, then $\Psi(\mathcal{T}^{\lfloor B/\nu \rfloor}) \geq \frac{1-\epsilon}{\nu+1} \cdot \Psi(\mathcal{T}^B)$.*
3. *Let $\mathcal{T} \subseteq V \times \{0\}$ be of size $\lfloor B/\nu \rfloor$. Then $\mathcal{S}' := \{(v, j) | (v, 0) \in \mathcal{T}, j \in [\nu]\} \subseteq \hat{V}$ is a set of size at most B such that $\Phi^{\geq 1}(\mathcal{S}') = \Psi(\mathcal{T})$.*

Now let $0 < \epsilon < 1$, $0 < \delta \leq 1/2$, $\mathcal{T} := \textsc{Greedy}(\Psi, \epsilon/2, \delta, \mathcal{I}, \nu, \lfloor B/\nu \rfloor)$ (Algorithm 8), and assume that $B \geq 2\nu/\epsilon$. Furthermore, let $\mathcal{S}' := \{(v, j) | (v, 0) \in \mathcal{T}, j \in [\nu]\} \subseteq \hat{V}$ be as in Lemma 6.13- (3). Then, according to Lemma 6.4, it holds that $\Phi^{\geq 1}(\mathcal{S}') = \Psi(\mathcal{T}) \geq \alpha' \cdot \Psi(\mathcal{T}^{\lfloor B/\nu \rfloor})$ with $\alpha' := 1 - 1/e - \epsilon/2$. Using Lemma 6.13- (2) and (1), we obtain $\Phi^{\geq 1}(\mathcal{S}') \geq \alpha'(1 - \frac{\epsilon}{2})/(\nu+1) \cdot \Psi(\mathcal{T}^B) \geq \alpha \cdot \Phi^{\geq 1}(\mathcal{S}^*_{\geq 1})$, with $\alpha := (1 - 1/e - \epsilon)/(\nu+1)$ and $\mathcal{S}^*_{\geq 1}$ being an optimal solution to $\Phi^{\geq 1}$ of size B. Now let $\mathcal{S}$ be the set among $\mathcal{S}'$ and $\emptyset$ that achieves the maximum out of $\Phi(\mathcal{S}')$ and $\Phi(\emptyset)$. Then $\mathcal{S}$ satisfies $2 \cdot \Phi(\mathcal{S}) \geq \Phi(\emptyset) + \Phi(\mathcal{S}') \geq \Phi^0(\mathcal{S}^*) + \alpha \cdot \Phi^{\geq 1}(\mathcal{S}^*) \geq \alpha \cdot \Phi(\mathcal{S}^*)$, where $\mathcal{S}^*$ is an optimal solution of size B to maximizing Φ. Thus we get the following theorem.

Theorem 6.14. *Let* $0 < \epsilon < 1$ *and* $\delta \leq 1/2$. *In the correlated setting, there is an algorithm that, with probability* $1 - \delta$, *outputs a solution* $\mathcal{S}$ *that satisfies* $\Phi(\mathcal{S}) \geq \frac{1-1/e-\epsilon}{2(\nu+1)} \cdot \Phi(\mathcal{S}^*)$, *where* $\mathcal{S}^*$ *denotes an optimal solution of size* B *to maximizing* Φ.

Chapter 7

Recommending Links to Reduce the Bias in Social Networks

News diffusion in social networks, whether they are managed by a malicious user or by an algorithm, are likely to create homogeneous polarized clusters. Situations in which information, ideas, or beliefs are amplified or reinforced by communication and repetition inside a defined system, called echo chambers.
A solution to this problem could be to create bridges that connect people of opposing views. By connecting different parts of the network, we hope to reduce this polarization phenomenon. In this chapter, we study an algorithmic technique for bridging these chambers and thus reduce the bias. We formulate the link recommendation task as an optimization problem that asks to suggest a fixed number of new connections to a subset of users with the aim of maximizing the network portion that is reached by their generated content. In detail, we give a constant-factor approximation algorithm for the problem of maximizing the social influence of a given set of target users by suggesting a fixed number of new connections. We experimentally show that, with few new links and small computational time, our algorithm is able to increase by far the social influence of the target users. We compare our algorithm with several baselines and show that it is the most effective one in terms of increased influence.

Most of the results presented in this chapter are included in [8].

7.1 Problem Definition

Given a directed graph $G = (V, E, b)$, a budget $B \in \mathbb{N}$ and a set $A \subseteq V$ of seed nodes consider a larger graph $\bar{G} = (V, \bar{E}, b)$ where $\bar{E}$ contains all the edges in E and, in addition, it contains all the remaining pairs between nodes in A and any other node in V, namely, $\bar{E} = E \cup (A \times V)$ with the constraint that $\sum_{(u,v)\in\bar{E}} b_{uv} \leq 1$ for any node $v \in V$.

For a set S of edges in $\bar{E} \setminus E$, let us denote by $G(S)$ the graph augmented with the set of edges S, i.e., $G(S) = (V, E \cup S)$. In the *Influence Maximization with Augmentation problem* (IMA) we aim at finding a set $S \subseteq \bar{E} \setminus E$ of size B in order to maximize the expected number of active nodes in $G(S)$ at the end of the ICM or LTM process. Let us denote by $\mathcal{G}(\mathcal{S})$ the set of all possible live-edges sampled from $G(S)$, then we denote as

$$\sigma(A, S) := \sum_{G' \in \mathcal{G}(\mathcal{S})} \mathbf{P}\left(G'\right) |R_A(G')|$$

the expected number of activated nodes at the end of the process with seed nodes A in graph $G(S)$, where, $R_A(G')$ is the set of reachable nodes from nodes in A in graph G', i.e., $R_A(G') = \{u \in V : \exists \text{ path from } v \in A \text{ to } u \in G'\}$. Thus, *Influence Maximization with Augmentation problem* consists in finding a set S^* such that

$$S^* = \underset{S \subseteq \bar{E} \setminus E : |S| \leq B}{\arg\max} \; \sigma(A, S).$$

7.2 Approximation for IMA under LTM

In this section, we first prove that the function $\sigma(A, S)$ under LTM is monotone and submodular w.r.t. to the set of added edges S and then we exploit this property to provide a constant approximation algorithm for the IMA problem.

Le us denote by $\mathcal{G}$ the set of all possible live-edge graphs sampled from graph G. For every $G' = (V, E') \in \mathcal{G}$ we denote by $\mathbf{P}(G')$ the probability that the live-edge graph is sampled, and by $p(v, G')$ the probability for a node $v \in V$ of having an incoming edge in G'. Therefore, $p(v, G')$ is either equal to b_{uv} if there exists an edge $(u, v) \in E'$, for some $u \in N_v^-$, or $p(v, G') = 1 - \sum_{(u,v)\in E} b_{uv}$ if no edge is selected.

Then we can easily extend this notation to a set of nodes $V' \subseteq V$ as $p(V', G') = \prod_{v \in V'} p(v, G')$. Thus the probability of a live-edge $G' = (V, E')$ is equal to

$$\mathbf{P}\left(G'\right) = p(V, G') = \prod_{(u,v) \in E'} b_{uz} \prod_{v \in V : \nexists (u,v) \in E'} \left(1 - \sum_{(u,v) \in E} b_{uv}\right).$$

Finally we denote as $\mathbf{P}_S\left(G'\right) = p(V, G', S)$ the probability of a live-edge graph $G' \in \mathcal{G}(S)$, where $\mathcal{G}(S)$ is the set of all possible live-edge graphs sampled from graph $G(S)$.

We first fix the following observation: Consider the probability of a live-edge G' sampled from $\mathcal{G}(S \cup \{e\})$ and a second live-edge graph $G'' \in \mathcal{G}(S)$ when adding a new edge $e = (a, v)$ in $G(S)$. Probabilities $\mathbf{P}_{S \cup e}\left(G'\right)$ and $\mathbf{P}_S\left(G''\right)$ differ only in the calculation concerning node v, since all the other nodes have the same set of possible in-neighbors with the same set of weights. Therefore, we have 3 cases:

1. For any live-edge graph $G' \in \mathcal{G}(S \cup e)$ that selects an edge different from e as incoming edge to v there exists a corresponding $G'' \in \mathcal{G}(S)$ with the same edge set, therefore we have that $\mathbf{P}_{S \cup e}\left(G'\right) = \mathbf{P}_S\left(G''\right)$;
2. For any live-edge graph $G' \in \mathcal{G}(S \cup e)$ in which no incoming edge is selected for the node v there exists a corresponding $G'' \in \mathcal{G}(S)$ with the same edge set. In this case $\mathbf{P}_{S \cup e}\left(G'\right) = p(V \setminus \{v\}, G', S)(1 - \sum_{z \in N_v^-} b_{zv} - b_e)$ and $\mathbf{P}_S\left(G''\right) = p(V \setminus \{v\}, G'', S)(1 - \sum_{z \in N_v^-} b_{zv})$, where $p(V \setminus \{v\}, G', S) = p(V \setminus \{v\}, G'', S)$;
3. For any live-edge graph $G' \in \mathcal{G}(S \cup e)$, $G' = (V, E')$, that selects e as incoming edge to v we have that $\mathbf{P}_{S \cup e}\left(G'\right) = p(V \setminus \{v\}, G', S) b_e$. Note that, in this case we have $|R_A(G')| \geq |R_A(G'')|$, for any live-edge graph $G'' \in \mathcal{G}(S)$ with edge set $E' \setminus \{e\}$.

Then, the following theorem holds.

Theorem 7.1. *Given a graph $G = (V, E)$, $\sigma(A, (V, E \cup S))$ is a monotone submodular function of $S \subseteq \bar{E} \setminus E$.*

The proof is based on the observation that we can divide the live-edge graphs in $\mathcal{G}(S)$ in two sets: the ones that have an incoming edge for the nodes chosen as endpoints of the solution T and the one for which these nodes have no incoming edge. Then, we

Algorithm 6 Greedy algorithm for IMA with approximate estimation of marginal increment.

```
Require: Social graph $G = (V, E, b)$; Seed set $A$; Budget $B$
for $i = 1, 2, \ldots, B$ do
    for each $e \in \bar{E} \setminus (E \cup S)$ do
        Use repeated sampling to estimate a $(1 + \lambda)$-approx. of $\sigma(A, S \cup \{e\})$ with prob. $1 - \delta$
        Let $\tilde{\sigma}(A, S \cup \{e\})$ be the estimation
    $\hat{e} = \arg\max\{\tilde{\sigma}(A, S \cup \{e\}) \mid e = (a, v) \in \bar{E} \setminus (E \cup S)\}$
    $S := S \cup \{\hat{e}\}$
return $S$
```

can prove that for any live-edge graph of the former set there exists a corresponding live-edge graph in $\mathcal{G}(T)$ that is sampled with the same probability. Instead, in the latter case we notice that there exists a series of corresponding live-edge graphs in $\mathcal{G}(T)$ with equal probability. The proof of Theorem 7.1 can be found in Appendix D.

Thus, we can use a simple greedy algorithm (reported in Algorithm 6) to find a set S of edges whose value $\sigma(A, S)$ is at least $1 - 1/e$ times the one of an optimal solution for the IMA Problem. The algorithm iterates B times and, at each iteration, it adds to an initially empty solution S an edge $\hat{e} = (\hat{a}, \hat{v})$ s.t. $(\hat{a}, \hat{v}) \in \bar{E} \setminus E$ that gives the maximum marginal increment of the value of $\sigma(A, S)$. Note that as explained in Chapter 2 we are not able to compute exactly the value of $\sigma(A, S)$ in polynomial time but, with probability $1 - \delta$, we can compute an $1 + \lambda$ approximation of it by sampling a polynomial number of live-edge graphs, for any λ and δ. We then exploit the result of Nemhauser et al. that allows us to analyze the greedy algorithm in the case of monotone submodular objective functions that can be approximately evaluated [25]. The next corollary follows.

Corollary 7.2. *Algorithm 6 guarantees an approximation factor of* $\left(1 - \frac{1}{e} - \epsilon\right)$ *for the IMA problem, where* ϵ *is any positive real number.*

The computational complexity of Algorithm 6 is $\mathcal{O}(B \cdot |V| \cdot g(|V|, |E| + B))$, where $g(|V|, |E| + B)$ is the complexity of computing an approximation $\tilde{\sigma}(A, S)$ of $\sigma(A, S)$ in a graph with $|V|$ nodes and $|E| + B$ edges. Algorithms 6 runs in B iterations, each of which requires estimating the expected spread of $\mathcal{O}(|V|)$ node sets. Since $g(|V|, |E| + B) = \mathcal{O}(|E| \cdot Q)$ where Q is the number of simulations, then the complexity of the greedy algorithm is $\mathcal{O}(B \cdot |V| \cdot |E| \cdot Q)$ which is clearly infeasible, in terms of running time, for networks with millions of vertices and edges, the "normal" size of many interesting real-world networks.

7.3 Improving the running time

In what follows we propose some techniques to heuristically reduce the running time of the greedy algorithm. Note that these techniques can be applied to the greedy algorithm proposed in [22] and to Algorithm 6 and with the due modifications also to the remaining algorithms presented in this thesis. In Section 7.4 we evaluate an implementation of the algorithm that exploits a combination of these heuristics.

Exploiting submodularity. Since $\sigma(A, S)$ is submodular, we have that the increment to the expected number of active nodes after adding an edge e to $G(S)$ is monotonic non-increasing. Thus, the increment is upper bounded by any solution $S' \subseteq S$ with the addition of the new edge e, that is $\sigma(A, S' \cup \{e\}) - \sigma(A, S') \geq \sigma(A, S \cup \{e\}) - \sigma(A, S)$. We can exploit this property in Algorithm 6 to reduce the computational complexity of our algorithm. Consider the loop at line 1 for any iteration $i \geq 2$ and for some edge e, we check if the increment found so far is greater than the increment in the previous iteration, i.e., $i - 1$, with the edge e. In this case, in fact, we know that the edge e cannot increase the value of $\sigma(A, S)$ more than the maximum found so far. Therefore, in this case we prune the search.

Live-edge graph reduction. At the end of each iteration of the loop at line 1 of Algorithm 6, we reduce the size of all the live-edge graphs by removing the nodes that become influenced when adding an edge to the solution. Reducing the size of the live-edge graphs after each iteration reduces the time required to compute $\tilde{\sigma}(A, S \cup \{e\})$ for each new edge e.

Low probability candidate edge pruning. The greedy algorithm needs to compute $\tilde{\sigma}(A, S \cup \{e\})$ for each $e \in \bar{E} \setminus (E \cup S)$ to find an edge that maximizes this quantity. However, if we have that $b_{a_1 u} > b_{a_2 u}$ for some nodes $a_1, a_2 \in A$ and $u \in V \setminus A$, then $\sigma(A, S \cup \{a_1, u\}) > \sigma(A, S \cup \{a_2, u\})$. Thus, in the loop of line 2 we only consider, for each $u \in V \setminus A$, the edge $(a, u) \in \bar{E} \setminus (E \cup S)$ with the highest weight.

Reduction to Limited Seed Selection. Given a weighted directed graph with edge weights capturing influence probabilities (in ICM or LTM), an integer B', and two sets of nodes $A, L \subseteq V$, the *Limited Seed Selection problem* (LSS) aims to to find a set of B' users $S \subseteq L$ such that, by targeting $S \cup A$, the expected number of influenced users is maximum. Nodes in S are excluded from the objective function.

Problems IMA and LSS have the same objective function, but the former looks for a set of edges, while the later looks for a set of nodes to be added to a give set of seeds. In what follows we describe how to transform an instance I_{IMA} of the IMA problem into an instance I_{LLS} of the LSS problem and how to transform a solution S_{LSS} for I_{LLS} into a solution S_{IMA} for I_{IMA} with the same value.

Given $I_{IMA} = (G, A, B)$, we define $I_{LSS} = (\hat{G}, L, B)$ where $\hat{G} = (\hat{V}, \hat{E}, \hat{b})$ is a graph obtained by adding $|L| = |V \setminus A| \cdot |A|$ nodes and edges to G. Formally, let $L = \bigcup_{a \in A} L_a$ be the additional nodes, where $|L_i| = |V \setminus A|$ and $L_{a_1} \cap L_{a_2} = \emptyset$, for each $a_1, a_2 \in A$, $a_1 \neq a_2$. Then, $\hat{V} = V \cup L$ and $\hat{E} = E \cup \bigcup_{a \in A}(L_a \times (V \setminus A))$. The weights of the new edges are equal to that of the corresponding edges in $\bar{E} \setminus E$, i.e. $\hat{b}_{a'v} = b_{av}$, for each $a \in A$, $a' \in L_a$, and $v \in V \setminus A$. Any solution S_{LSS} for I_{LSS} is made of nodes in L and each of these nodes corresponds to unique edge in G, we define S_{IMA} accordingly.

We denote with σ_G and $\sigma_{\hat{G}}$ the expected number of influenced nodes in G and $\hat{G}$, respectively. The next theorem show that the two solutions have the same value.[1]

Theorem 7.3. *In both ICM and LTM, $\sigma_{\hat{G}}(A \cup S_{LSS}, \emptyset) = \sigma_G(A, S_{IMA}) + B$.*

Proof. For any graph H, we denote by $\mathcal{G}(H)$ the set of all possible live-edge graphs sampled from H. For LSS, we have that $\sigma_{\hat{G}}(A \cup S_{LSS}, \emptyset) = \sum_{G' \in \mathcal{G}(\hat{G})} \mathbf{P}\left(G'\right) |R_{A \cup S_{LSS}}(G')|$, while for IMA, we have

$$\sigma_G(A, S_{IMA}) = \sum_{G'' \in \mathcal{G}(G(S_{IMA}))} \mathbf{P}\left(G''\right) |R_A(G'')|.$$

For any live-edge graph generated from $G(S_{IMA})$ there exists a live-edge graph generated from $\hat{G}$ with the same probability and viceversa since sets $E \cup S_{IMA}$ and $\hat{E}$ are associated with the same weights.

For any $G' \in \mathcal{G}(\hat{G})$, let G'' be its corresponding graph in $\mathcal{G}(G(S_{IMA}))$, then we have $R_{A \cup S_{LSS}}(G') = R_A(G'') \cup S_{LSS}$. Therefore, $\sigma_{\hat{G}}(A \cup S_{LSS}, \emptyset) = \sigma_G(A, S_{IMA}) \cup S_{LSS}$. Since $\sigma_G(A, S_{IMA}) \cap S_{LSS} = \emptyset$ and $|S_{LSS}| = B$ the statement follows. □

[1] Note that the objective function of LSS is $\sigma_{\hat{G}}(A \cup S_{LSS}, \emptyset) - B$.

Thanks to Theorem 7.3, we can use any algorithm for LSS to solve IMA. Problem LSS is different from the influence maximization problem in [11]. However, many algorithm for this latter can be easily adapted for solving LSS. In particular, we adapt the algorithm presented in [94] in such a way that it finds a set of seed nodes $S \subseteq L$, given a limited set of nodes L.

7.4 Experimental study

In this section we experimentally evaluate the performance of our greedy algorithm and of that in [22]. For both ICM and LTM, we implemented two versions of these algorithms: GREEDY1 exploits the first three heuristics described in the previous section; and GREEDY2 exploits the reduction to LSS. We compare the number of activated nodes in a graph augmented by using the greedy solution with the number of activated nodes in the original graph and in the graph augmented by using several alternative baselines.

All our experiments have been performed on a computer equipped with two Intel Xeon E5-2643 CPUs, each with 6 cores clocked at 3.4GHz and 128GB of main memory, and our programs have been implemented in C++ (gcc compiler v4.8.2 with optimization level O3).

We evaluate the performance of the algorithm on four types of randomly generated directed networks which exhibit many of the structural features of complex networks and on real-world graphs that are suitable for our problem, taken from KONECT [95], ArnetMiner [96] and SNAP [97] repositories. The size of the graphs are reported in Table 7.1. For both synthetic and real-world networks, we choose 0.1% of the nodes in V as seeds and we add up to $B = 2 \cdot |A|$ edges. For these experiments, the seed nodes are chosen uniformly at random.

The weights on the edges in both models are generated as follows:

In ICM we are assigned the probabilities to the edges according to the weighted model, i.e., for each edge (u, v), assign $b_{uv} = 1/N_v^-$; Note that given this assignment we can further improve the running time. In fact, we do not generate $\hat{G}$ from G by adding $|L| = |V \setminus A| \cdot |A|$ nodes and edges, but we consider the heuristic in which we add only $|L| = |V \setminus A|$ new elements because the probability of an edge does not depend

Name	$\lvert V\rvert$	$\lvert E\rvert$
Software Engineering (SE)	3,141	14,787
Theoretical CS (TCS)	4,172	14,272
High-Performance Comp. (HPC)	4,869	35,036
Wiki-Vote (Wiki)	7,115	103,689
Computer Graphic (CGM)	8,336	41,925
Computer Networks (CN)	9,420	53,003
Artificial Intelligence (AI)	27,617	268,460
Slashdot (Sl)	51,083	130,370
Epinions (Epi)	75,879	508,837
Slashdot-Zoo (Sl-z)	79,116	515,397
Digg	279,630	1,731,653
Citeseer	384,413	1,751,463
Twitter	465,017	834,797

Table 7.1: Real-world networks.

on the source node but only on the sink node. For LTM model instead we generate $|L| = |V \setminus A| \cdot |A|$ nodes and edges.

In LTM instead we generate for each node $v \in V$ a random variable $\bar{b}_v \in [0, 0.5]$ that represent the probability that v does not select any edge in the live-edge graph, then we assigned for each edge (u, v) in the graph a weight equal to $\frac{1-\bar{b}_v}{N_v^-}$ and $\frac{\bar{b}_v}{2}$ is assigned to a new edge. Note that it is unlikely that more than two edges towards the same node in $V \setminus A$ are added in the solution, see Section D.2.

As a measure of the quality of the solution, we adopt the expected number of active nodes $\sigma(A, S)$. As discussed in the preliminaries, it has been proven that evaluating this function is $\#P$-complete in general. However, by simulating the diffusion process a polynomial number of times and sampling the resulting active sets, it is possible to obtain arbitrarily good approximations to $\sigma(A, S)$. We experimentally tested that 500 samples are enough to obtain a good estimation. Therefore, we run 500 trial to estimate the value of σ in the algorithms and in the final solution, see Appendix D.1.

7.4.1 Results on Real Networks

In the following tables, we use $***$ when the GREEDY1 was not able to compute the final solution for a given graph in the time limit that we set to four hours. In Table 7.4 and 7.5, we report: $\sigma(A, \emptyset)$ and $\sigma(A, \emptyset)\%$ that are the absolute and relative initial number of active nodes; $\sigma(A, S)$ and $\sigma(A, S)\%$, are the absolute and relative number of active nodes after the edge addition; the relative increment computed as $I = \frac{\sigma(A,S)-\sigma(A,\emptyset)}{\sigma(A,\emptyset)} \times 100$; and T, the time in seconds. The expected number of active nodes in GREEDY1 and GREEDY2 are similar, except from the time (GREEDY2 is faster), and a small difference is due to the sampling technique used to estimate $\tilde{\sigma}(A, S)$. From the table we can see that our algorithm is able to highly increase the number of activated nodes with respect to the original graph. Moreover, thanks to the reduction to LSS, the running time is small and this allows us to solve IMA in large networks.

We compare GREEDY1 and GREEDY2 with the following alternatives. AdamicAdar (AA): connect the seeds to a set of B nodes with the highest Adamic-Adar index [98]; PrefAtt (PA): connect the seeds to a set of B nodes chosen according to the Preferential Attachment model [99, 100]; Jaccard (J): connect the seeds to a set of B nodes with the highest Jaccard coefficient [101]; Degree (D): connect the seeds to a set of B nodes with the highest out-degree; Topk (TopK): connect the seeds to a set of B nodes with the highest harmonic centrality [102]; Prob (Prob): connect the seeds to a set of B nodes adding the edges with highest probability; Seed (KKT): connect the seeds to a set of B nodes chosen as seeds by the greedy algorithm proposed by Kempe et al. [11]; Random (R): connect the seeds to a set of B nodes extracted uniformly at random.

Note that the first three algorithms are well-known algorithms used for link recommendation. For each graph we report the relative increment of active nodes computed as $I = \frac{\sigma_{ALG}(A,S) - \sigma(A,\emptyset)}{\sigma(A,\emptyset)} \times 100$, where $\sigma_{ALG}(A, S)$ is the expected number of active nodes computed using algorithm ALG and $B = 2 \cdot |A|$.

The expected number of active nodes is similar for both algorithms and a small difference is due to the sampling technique used to estimate $\tilde{\sigma}(A, S)$. The results for the other real-world and random networks are reported in Tables 7.2.

In Figure 7.1, we compare GREEDY2 with the other approaches on the Slashdot-Zoo network. Results for other networks and for GREEDY1 are similar and are reported in table 7.4 and 7.5. The plots show the average number of affected nodes as a function of the number of added edges. The experiments clearly show that GREEDY2 outperforms all the alternative baselines in terms of expected number of active nodes. Indeed, all the other competitive algorithms require to add a large number of edges to the seed set A in order to significantly increase the expected number of influenced nodes with respect to the initial value (see value at $B = 0$), whereas our algorithm increases $\sigma(A, S)$ by a greater amount with only few added edges.

In Tables 7.2 and 7.3 we report the results on the number of nodes that are activated by GREEDY1 and GREEDY2 on real-word networks.

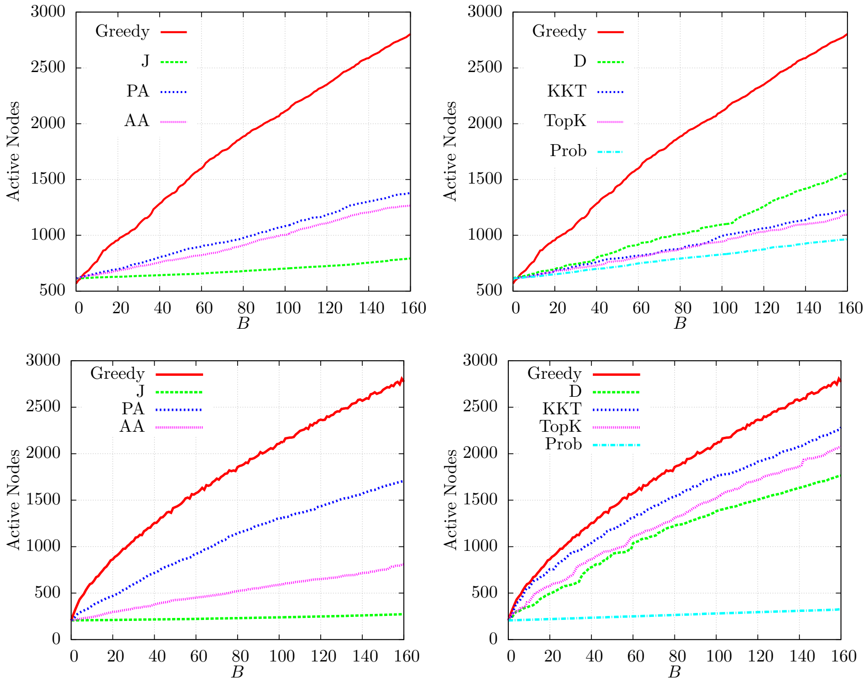


FIGURE 7.1: Comparison between GREEDY2 and the baselines on Slashdot-Zoo for ICM (top) and LTM (bottom).

In Tables 7.4 and 7.5 we report the experimental comparisons between GREEDY2 and the alternative baselines on real-world networks.

In Figure 7.2 we compare the result obtained applying GREEDY1 and GREEDY2 algorithms to the Artificial Intelligence network given a seed set A. The results for the other real-world and random networks are reported in Tables 7.2-7.3 and D.3-D.4, respectively.

G	$\sigma(A,\emptyset)$	$\sigma(A,\emptyset)\%$	GREEDY1				GREEDY2			
			$\sigma(A,S)$	$\sigma(A,S)\%$	$I\%$	time (sec.)	$\sigma(A,S)$	$\sigma(A,S)\%$	$I\%$	time (sec.)
SE	13.38	0.43	103.09	3.28	670.56	5.74	103.45	3.29	670.95	0.04
TCS	9.49	0.23	97.54	2.34	928.26	10.44	97.63	2.34	928.76	0.07
HPC	9.36	0.19	164.92	3.39	1662.23	12.18	165.75	3.40	1670.83	0.07
Wiki	10.07	0.14	338.93	4.76	3266.84	35.31	333.37	4.69	3257.87	0.12
CGM	20.34	0.24	253.84	3.05	1148.13	35.88	257.62	3.09	1166.71	0.12
CN	22.21	0.24	408.69	4.34	1740.10	49.33	397.95	4.22	1765.86	0.10
AI	68.94	0.25	1055.76	3.82	1431.44	374.66	1017.82	3.69	1362.71	0.47
Sl	126.11	0.25	621.75	1.22	393.01	2050.65	612.52	1.20	408.63	1.72
Epi	352.17	0.46	1236.71	1.63	251.17	3884.27	1230.23	1.62	249.32	3.25
Sl-z	570.84	0.72	3485.02	4.41	510.50	3730.36	3425.87	4.40	505.14	2.63
Digg	3848.57	1.38	***	***	***	***	14835.40	5.31	285.48	14.32
Citeseer	683.16	0.18	***	***	***	***	13072.36	3.40	1813.52	12.20
Twitter	2861.20	0.62	***	***	***	***	207061.00	44.53	7136.87	10.23
Pokec	23472.20	1.44	***	***	***	***	83726.90	5.13	256.71	78.91

TABLE 7.2: Results for Real-world networks (ICM).

G	$\sigma(A,\emptyset)$	$\sigma(A,\emptyset)\%$	GREEDY1				GREEDY2			
			$\sigma(A,S)$	$\sigma(A,S)\%$	$I\%$	time (sec.)	$\sigma(A,S)$	$\sigma(A,S)\%$	$I\%$	time (sec.)
SE	12.40	0.39	49.33	1.57	297.83	2.10	59.45	1.89	276.88	0.10
TCS	8.34	0.20	47.39	1.14	468.58	2.95	51.78	1.24	430.62	0.13
HPC	7.91	0.16	73.75	1.51	832.21	3.26	87.98	1.81	935.87	0.39
Wiki	9.17	0.13	119.76	1.68	1205.96	6.68	120.93	1.70	1151.57	1.33
CGM	16.69	0.20	107.49	1.29	543.95	5.46	128.96	1.55	525.37	0.29
CN	18.30	0.19	174.92	1.86	856.03	6.84	204.73	2.17	936.94	0.43
AI	53.15	0.19	407.57	1.48	666.77	20.05	530.99	1.92	767.02	4.74
Sl	87.97	0.17	638.35	1.25	625.62	44.86	663.43	1.30	592.51	6.82
Epi	174.98	0.23	2216.56	2.92	1166.74	54.39	2248.09	2.96	999.34	37.42
Sl-z	206.35	0.26	2773.19	3.51	1243.94	47.69	3203.52	4.05	1160.21	36.48
Citeseer	623.82	0.16	***	***	***	***	5901.46	1.54	846.03	42.98
Twitter	1673.07	0.36	***	***	***	***	127414.00	27.40	7515.56	13.33

Table 7.3: Results for Real-world networks (LTM).

G	$B=0$		GREEDY2	AA	PA	J	D	TopK	Prob	KKT
	$\sigma(A,\emptyset)$	$\sigma(A,\emptyset)\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$
SE	13.38	0.43	670.56	101.66	57.67	66.03	53.30	353.60	156.58	396.32
TCS	9.49	0.23	928.26	114.78	100.74	115.99	102.56	82.94	351.27	343.37
HPC	9.36	0.19	1662.23	176.33	15.33	153.93	16.15	15.99	363.73	896.97
Wiki	10.07	0.14	3266.84	330.02	1404.37	157.00	2078.44	2054.02	228.10	2284.13
CGM	20.34	0.24	1148.13	158.46	43.18	124.91	48.38	45.97	238.87	271.13
CN	22.21	0.24	1740.10	153.82	375.71	102.83	661.84	662.23	221.30	575.31
AI	68.94	0.25	1431.44	78.30	24.42	72.80	30.69	143.37	192.22	522.45
Sl	126.11	0.25	393.01	60.30	80.95	79.39	76.16	83.46	131.67	95.24
Epi	352.17	0.46	251.17	62.65	123.09	48.13	94.54	102.32	75.97	115.44
Sl-z	570.84	0.72	510.50	106.56	125.05	29.39	153.94	93.42	57.57	99.65
Digg	3848.57	1.38	285.48	126.80	149.44	10.58	145.44	121.71	39.07	86.65
Citeseer	683.16	0.18	1813.52	124.74	446.49	91.89	931.10	498.96	215.97	389.25
Twitter	2861.20	0.62	7136.87	1718.20	6286.58	5721.36	6510.76	6649.86	75.99	5148.94

Table 7.4: Baseline results for real-world networks (ICM).

G	$B=0$		GREEDY2	AA	PA	J	D	KKT	Prob	TopK
	$\sigma(A,\emptyset)$	$\sigma(A,\emptyset)\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$
SE	12.40	0.39	276.88	54.00	83.70	40.08	86.25	166.39	58.13	192.33
TCS	8.34	0.20	430.62	69.12	152.37	69.57	139.31	315.47	115.80	99.71
HPC	7.91	0.16	935.87	105.65	128.70	75.12	126.80	508.44	116.18	115.30
Wiki	9.17	0.13	1151.57	196.90	914.84	84.37	873.36	889.14	105.12	864.51
CGM	16.69	0.20	525.37	96.86	183.59	74.57	188.57	327.83	119.88	131.10
CN	18.30	0.19	936.94	115.17	310.27	56.65	435.75	518.69	129.63	430.47
AI	53.15	0.19	767.02	59.18	181.75	45.21	206.81	407.02	105.22	198.18
Sl	87.97	0.17	592.51	81.88	489.49	47.09	505.79	525.57	79.86	498.96
Epi	174.98	0.23	999.34	529.37	1018.25	38.99	901.28	1039.70	56.72	851.96
Sl-z	206.35	0.26	1160.21	291.21	724.98	32.25	753.77	1004.12	57.90	903.31
Twitter	1673.07	0.36	7515.56	735.84	5324.15	4047.27	5505.72	7191.78	61.48	6329.72

Table 7.5: Baseline results for real-world networks (LTM).

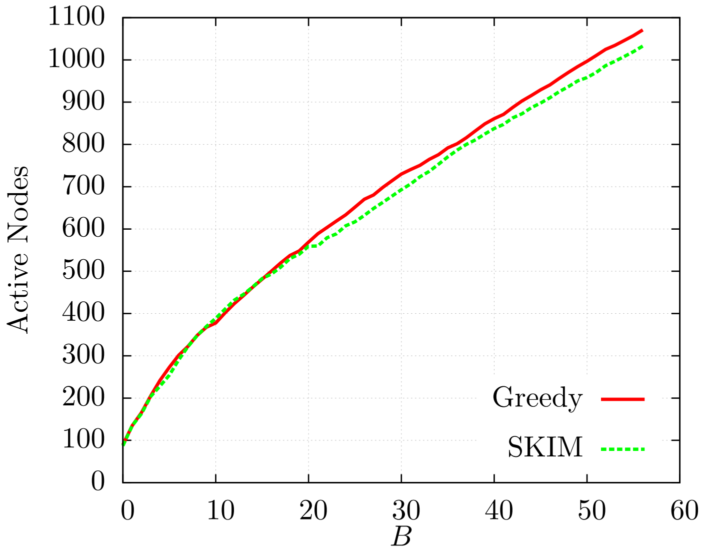


FIGURE 7.2: Comparison of GREEDY1 and GREEDY2 on AI network.

Chapter 8

Conclusion

Nowadays social media are substantial sources of information for voters; potential attackers can manipulate the outcome of elections through the spread of targeted ads and/or fake news. The integrity of elections is crucial to the functioning of democratic institutions. Political campaigning is a common legitimate mean for convincing voters. When transparent, such communication is critical to the effective functioning of democracy and can exercise considerable influence on voting behavior. Malicious control over the information spread through these channels can have a large impact, but it is hard to achieve due to the relative transparency of traditional media sources. The massive usage of social media for political campaigning is a game-changer. Fake news are an increasingly prevalent way of interfering with elections.

In this dissertation, we tackle the problem studying models to manipulate or reduce the effect of manipulators on social networks. We define new and more general models introducing the *Linear Threshold Ranking* and the *Probabilistic Linear Threshold Rankings*, natural and powerful extensions of the well-established *Linear Threshold Model*. Then we considered possible ways to prevent election control reducing social biases and give to the users the possibility to be exposed to multiple sources with diverse perspectives and balancing users opinions.

The results presented in this thesis made a progress in the study of overcoming the effect of manipulators on social networks and point out several new challenges. In the following, we list several open problems worthy of further investigation.

The first open problem, directly related to this thesis, is the study of how to prevent election control for the integrity of voting processes, e.g., through the placement of monitors in the network [103, 104] or by considering strategic settings [105, 106].

Moreover, we would like to extend our models in order to consider uncertainty models also for the diffusion process, e.g., in robust influence maximization only a probability distribution on the edge's weights is known [107].

As future research directions we would like to further study our model in a wider range of scenarios which are not currently captured, including *multi-winner* and *proportional representation* systems. We also believe that approaches that mix constructive and destructive control could be analyzed to get better approximation ratios. We are also interesting in improving the approximation guarantee for the μ-ν-Balance problem in both settings, most importantly for the heterogeneous case with $\nu > 3$.

Bibliography

[1] Kiran Garimella, Aristides Gionis, Nikos Parotsidis, and Nikolaj Tatti. Balancing information exposure in social networks. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems*, pages 4666–4674, 2017.

[2] Federico Corò, Emilio Cruciani, Gianlorenzo D'Angelo, and Stefano Ponziani. Vote for me!: Election control via social influence in arbitrary scoring rule voting systems. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS, pages 1895–1897, 2019. ISBN 978-1-4503-6309-9.

[3] Federico Corò, Emilio Cruciani, Gianlorenzo D'Angelo, and Stefano Ponziani. Exploiting social influence to control elections based on scoring rules. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI*, pages 201–207, 7 2019. doi: 10.24963/ijcai.2019/29.

[4] Mohammad Abouei Mehrizi, Federico Corò, Emilio Cruciani, and Gianlorenzo D'Angelo. Election control with voters' uncertainty: Hardness and approximation results. *CoRR*, abs/1905.04694, 2019. URL http://arxiv.org/abs/1905.04694.

[5] Mohammad Aboueimehrizi, Federico Corò, Emilio Cruciani, Gianlorenzo D'Angelo, and Stefano Ponziani. Models and algorithms for election control

through influence maximization. In *Proceedings of the 20th Italian Conference on Theoretical Computer Science, ICTCS*, 2019.

[6] Federico Corò, Emilio Cruciani, Gianlorenzo D'Angelo, and Stefano Ponziani. Exploiting social influence to control elections based on scoring rules. *J. Artif. Intell. Res. JAIR*, 2019.

[7] Ruben Becker, Federico Corò, Gianlorenzo D'Angelo, and Hugo Gilbert. Balancing spreads of influence in a social network. *CoRR*, abs/1906.00074, 2019. URL http://arxiv.org/abs/1906.00074.

[8] Federico Corò, Gianlorenzo D'Angelo, and Yllka Velaj. Recommending links to maximize the influence in social networks. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI*, pages 2195–2201, 7 2019. doi: 10.24963/ijcai.2019/304.

[9] Pedro M. Domingos and Matthew Richardson. Mining the network value of customers. In *Proceedings of the 7th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 57–66, 2001.

[10] Matthew Richardson and Pedro M. Domingos. Mining knowledge-sharing sites for viral marketing. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 61–70, 2002. doi: 10.1145/775047.775057.

[11] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11:105–147, 2015. doi: 10.4086/toc.2015.v011a004.

[12] Herbert W. Hethcote. The mathematics of infectious diseases. *SIAM Review*, 42 (4):599–653, 2000. doi: 10.1137/S0036144500371907.

[13] Jure Leskovec, Lars Backstrom, and Jon M. Kleinberg. Meme-tracking and the dynamics of the news cycle. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 497–506, 2009. doi: 10.1145/1557019.1557077.

[14] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. Working Paper 23089, National Bureau of Economic Research, January 2017. URL http://www.nber.org/papers/w23089.

[15] Emilio Ferrara. Disinformation and social bot operations in the run up to the 2017 french presidential election. *First Monday*, 22(8), 2017.

[16] Daniel Kreiss. Seizing the moment: The presidential campaigns use of twitter during the 2012 electoral cycle. *New Media & Society*, 18(8):1473–1490, 2016.

[17] Sebastian Stier, Arnim Bleier, Haiko Lietz, and Markus Strohmaier. Election campaigning on social media: Politicians, audiences, and the mediation of political communication on facebook and twitter. *Political Communication*, 35(1): 50–74, 2018. doi: 10.1080/10584609.2017.1334728.

[18] Robert M. Bond, Christopher J. Fariss, Jason J. Jones, Adam D. I. Kramer, Cameron Marlow, Jaime E. Settle, and James H. Fowler. A 61-million-person experiment in social influence and political mobilization. *Nature*, 489(7415):295–298, 2012. doi: 10.1038/nature11421.

[19] Michael D. Conover, Jacob Ratkiewicz, Matthew R. Francisco, Bruno Gonçalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. In *Proceedings of the 5th International Conference on Weblogs and Social Media*, 2011.

[20] Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H. Eugene Stanley, and Walter Quattrociocchi. The spreading of misinformation online. *Proceedings of the National Academy of Sciences*, 113(3):554–559, 2016. doi: 10.1073/pnas.1517441113.

[21] Pasin Manurangsi. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 954–961, 2017. doi: 10.1145/3055399.3055412.

[22] Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. Recommending links through influence maximization. *Theor. Comput. Sci.*, 764:30–41, 2019. doi: 10.1016/j.tcs.2018.01.017.

[23] Andreas Krause, Ajit Paul Singh, and Carlos Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9:235–284, 2008.

[24] Jingrui He, Hanghang Tong, Qiaozhu Mei, and Boleslaw K. Szymanski. Gender: A generic diversified ranking algorithm. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems*, pages 1151–1159, 2012.

[25] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978. doi: 10.1007/BF01588971.

[26] Gerard Cornuejols, Marshall L Fisher, and George L Nemhauser. Exceptional paperlocation of bank accounts to optimize float: An analytic study of exact and approximate algorithms. *Management science*, 23(8):789–810, 1977.

[27] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia, editors. *Handbook of Computational Social Choice.* Cambridge University Press, 2016. doi: 10.1017/CBO9781107446984.

[28] Piotr Faliszewski and Jörg Rothe. Control and bribery in voting. In *Handbook of Computational Social Choice*, pages 146–168. Cambridge University Press, 2016. doi: 10.1017/CBO9781107446984.008.

[29] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. A short introduction to computational social choice. In *Proceedings of 33th Conference on Current Trends in Theory and Practice of Computer Science*, pages 51–69, 2007. doi: 10.1007/978-3-540-69507-3_4.

[30] Piotr Faliszewski and Ariel D. Procaccia. Ai's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[31] Ulle Endriss. *Trends in Computational Social Choice.* Lulu.com, 2017.

[32] John Bartholdi, Craig A Tovey, and Michael A Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and welfare*, 6(2): 157–165, 1989. doi: https://doi.org/10.1007/BF00303169.

[33] Ioannis Caragiannis, Jason A. Covey, Michal Feldman, Christopher M. Homan, Christos Kaklamanis, Nikos Karanikolas, Ariel D. Procaccia, and Jeffrey S. Rosenschein. On the approximability of dodgson and young elections. *Artif. Intell.*, 187:31–51, 2012. doi: 10.1016/j.artint.2012.04.004.

[34] Nir Ailon, Moses Charikar, and Alantha Newman. Aggregating inconsistent information: Ranking and clustering. *J. ACM*, 55(5):23:1–23:27, 2008. doi: 10.1145/1411509.1411513.

[35] Nadja Betzler, Michael R. Fellows, Jiong Guo, Rolf Niedermeier, and Frances A. Rosamond. Fixed-parameter algorithms for kemeny scores. In *Proceedings of 4th International Conference, AAIM*, pages 60–71, 2008. doi: 10.1007/978-3-540-68880-8_8.

[36] Palash Dey, Neeldhara Misra, Swaprava Nath, and Garima Shakya. A parameterized perspective on protecting elections. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence, IJCAI*, pages 238–244, 7 2019. doi: 10.24963/ijcai.2019/34.

[37] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 137–146, 2003.

[38] Wei Chen, Chi Wang, and Yajun Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1029–1038, 2010. doi: 10.1145/1835804.1835934.

[39] Wei Chen, Yifei Yuan, and Li Zhang. Scalable influence maximization in social networks under the linear threshold model. In *Proceedings of the 10th IEEE International Conference on Data Mining ICDM*, pages 88–97, 2010. doi: 10.1109/ICDM.2010.118.

[40] Bryan Wilder and Yevgeniy Vorobeychik. Controlling elections through social influence. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS*, pages 265–273, 2018.

[41] Edith Elkind, Piotr Faliszewski, and Arkadii M. Slinko. Swap bribery. In *Proceedings of the 2th International Symposium of Algorithmic Game Theory, SAGT*, pages 299–310, 2009. doi: 10.1007/978-3-642-04645-2_27.

[42] Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Complexity of shift bribery in committee elections. In *Proceedings of the 30th Conference on Artificial Intelligence,AAAI*, pages 2452–2458, 2016.

[43] John J. Bartholdi, III, Craig A. Tovey, and Michael A. Trick. How hard is it to control an election? *Math. Comput. Model.*, 16(8-9):27–40, August 1992. doi: 10.1016/0895-7177(92)90085-Y.

[44] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Anyone but him: The complexity of precluding an alternative. *Artif. Intell.*, 171(5-6):255–285, 2007. doi: 10.1016/j.artint.2007.01.005.

[45] Piotr Faliszewski, Rica Gonen, Martin Koutecký, and Nimrod Talmon. Opinion diffusion and campaigning on society graphs. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI*, pages 219–225, 2018. doi: 10.24963/ijcai.2018/30.

[46] Robert Bredereck and Edith Elkind. Manipulating opinion diffusion in social networks. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI*, pages 894–900, 2017. doi: 10.24963/ijcai.2017/124.

[47] Vincenzo Auletta, Ioannis Caragiannis, Diodato Ferraioli, Clemente Galdi, and Giuseppe Persiano. Minority becomes majority in social networks. In *Proceedings of the 11th International Conference of Web and Internet Economics, WINE*, pages 74–88, 2015. doi: 10.1007/978-3-662-48995-6_6.

[48] Markus Brill, Edith Elkind, Ulle Endriss, and Umberto Grandi. Pairwise diffusion of preference rankings in social networks. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI*, pages 130–136, 2016.

[49] Sirin Botan, Umberto Grandi, and Laurent Perrussel. Propositionwise opinion diffusion with constraints. In *Proceedings of the 4th AAMAS Workshop on Exploring Beyond the Worst Case in Computational Social Choice (EXPLORE)*, 2017.

[50] Kathrin Konczak and Jérôme Lang. Voting procedures with incomplete preferences. In *Proceedings of the IJCAI Multidisciplinary Workshop on Advances in Preference Handling, MPREF*, volume 20, 2005.

[51] Nadja Betzler, Susanne Hemmann, and Rolf Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proceedings of the 21th International Joint Conference on Artificial Intelligence, IJCAI*, pages 53–58, 2009.

[52] Lirong Xia and Vincent Conitzer. Determining possible and necessary winners given partial orders. *J. Artif. Intell. Res.*, 41:25–67, 2011. doi: 10.1613/jair.3186.

[53] Yann Chevaleyre, Jérôme Lang, Nicolas Maudet, and Jérôme Monnot. Possible winners when new candidates are added: The case of scoring rules. In *Proceedings of the 24th Conference on Artificial Intelligence, AAAI*, 2010.

[54] Dorothea Baumeister, Magnus Roos, and Jörg Rothe. Computational complexity of two variants of the possible winner problem. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems AAMAS*, pages 853–860, 2011.

[55] Lin Chen, Lei Xu, Shouhuai Xu, Zhimin Gao, and Weidong Shi. Election with bribed voter uncertainty: Hardness and approximation algorithm. *CoRR*, abs/1811.03158, 2018. URL http://arxiv.org/abs/1811.03158.

[56] Ceren Budak, Divyakant Agrawal, and Amr El Abbadi. Limiting the spread of misinformation in social networks. In *Proceedings of the 20th International Conference on World Wide Web, WWW*, pages 665–674, 2011. doi: 10.1145/1963405.1963499.

[57] Krzysztof R. Apt and Evangelos Markakis. Diffusion in social networks with competing products. In *Proceedings of the 4th International Symposium of Algorithmic Game Theory, SAGT*, pages 212–223, 2011. doi: 10.1007/978-3-642-24829-0_20.

[58] Shishir Bharathi, David Kempe, and Mahyar Salek. Competitive influence maximization in social networks. In *Proceedings of the 3th International Conference of Web and Internet Economics, WINE*, pages 306–311, 2007. doi: 10.1007/978-3-540-77105-0_31.

[59] Tim Carnes, Chandrashekhar Nagarajan, Stefan M. Wild, and Anke van Zuylen. Maximizing influence in a competitive social network: a follower's perspective. In *Proceedings of the 9th International Conference on Electronic Commerce: The Wireless World of Electronic Commerce*, pages 351–360, 2007. doi: 10.1145/1282100.1282167.

[60] Pradeep Dubey, Rahul Garg, and Bernard De Meyer. Competing for customers in a social network: The quasi-linear case. In *Proceedings of the 2th International*

Conference of Web and Internet Economics, WINE, pages 162–173, 2006. doi: 10.1007/11944874_16.

[61] Jan Kostka, Yvonne Anne Oswald, and Roger Wattenhofer. Word of mouth: Rumor dissemination in social networks. In *Proceedings of the 15th International Colloquium on Structural Information and Communication Complexity, SIROCCO*, pages 185–196, 2008. doi: 10.1007/978-3-540-69355-0_16.

[62] Wei Lu, Wei Chen, and Laks V. S. Lakshmanan. From competition to complementarity: Comparative influence diffusion and maximization. *PVLDB*, 9(2): 60–71, 2015. doi: 10.14778/2850578.2850581.

[63] Seth A. Myers and Jure Leskovec. Clash of the contagions: Cooperation and competition in information diffusion. In *Proceedings of the 12th IEEE International Conference on Data Mining, ICDM*, pages 539–548, 2012. doi: 10.1109/ICDM.2012.159.

[64] Noga Alon, Michal Feldman, Ariel D. Procaccia, and Moshe Tennenholtz. A note on competitive diffusion through social networks. *Inf. Process. Lett.*, 110 (6):221–225, 2010. doi: 10.1016/j.ipl.2009.12.009.

[65] Sanjeev Goyal and Michael J. Kearns. Competitive contagion in networks. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC*, pages 759–774, 2012. doi: 10.1145/2213977.2214046.

[66] Vasileios Tzoumas, Christos Amanatidis, and Evangelos Markakis. A game-theoretic analysis of a competitive diffusion process over social networks. In *Proceedings of the 8th International Conference of Web and Internet Economics, WINEs*, pages 1–14, 2012. doi: 10.1007/978-3-642-35311-6_1.

[67] Allan Borodin, Mark Braverman, Brendan Lucier, and Joel Oren. Strategyproof mechanisms for competitive influence in networks. *Algorithmica*, 78(2):425–452, 2017. doi: 10.1007/s00453-016-0169-0.

[68] Çigdem Aslay, Antonis Matakos, Esther Galbrun, and Aristides Gionis. Maximizing the diversity of exposure in a social network. In *Proceedings of the 18th IEEE International Conference on Data Mining, ICDM*, pages 863–868, 2018. doi: 10.1109/ICDM.2018.00102.

[69] Antonis Matakos and Aristides Gionis. Tell me something my friends do not know: Diversity maximization in social networks. In *Proceedings of the 18th IEEE International Conference on Data Mining, ICDM*, pages 327–336, 2018. doi: 10.1109/ICDM.2018.00048.

[70] Zhepeng (Lionel) Li, Xiao Fang, and Olivia R. Liu Sheng. A survey of link recommendation for social networks: Methods, theoretical foundations, and future research directions. *ACM Trans. Management Inf. Syst.*, 9(1):1:1–1:26, 2018. doi: 10.1145/3131782.

[71] Linyuan Lu and Tao Zhou. Link prediction in complex networks: A survey. *CoRR*, abs/1010.0725, 2010. URL http://arxiv.org/abs/1010.0725.

[72] Magdalini Eirinaki, Jerry Gao, Iraklis Varlamis, and Konstantinos Tserpes. Recommender systems for large-scale social networks: A review of challenges and solutions. *Future Generation Comp. Syst.*, 78:413–418, 2018. doi: 10.1016/j.future.2017.09.015.

[73] Daniel Sheldon, Bistra N. Dilkina, Adam N. Elmachtoub, Ryan Finseth, Ashish Sabharwal, Jon Conrad, Carla P. Gomes, David B. Shmoys, William Allen, Ole Amundsen, and William Vaughan. Maximizing the spread of cascades using network design. *CoRR*, abs/1203.3514, 2012. URL http://arxiv.org/abs/1203.3514.

[74] Xiaojian Wu, Daniel Sheldon, and Shlomo Zilberstein. Efficient algorithms to optimize diffusion processes under the independent cascade model. *NIPS Work. on Networks in the Social and Information Sciences*, 2015.

[75] Chris J. Kuhlman, Gaurav Tuli, Samarth Swarup, Madhav V. Marathe, and S. S. Ravi. Blocking simple and complex contagion by edge removal. In *Proceedings of the 13th IEEE International Conference on Data Mining ICDM*, pages 399–408, 2013. doi: 10.1109/ICDM.2013.47.

[76] Masahiro Kimura, Kazumi Saito, and Hiroshi Motoda. Solving the contamination minimization problem on networks for the linear threshold model. In *Proceedings of the 10th Pacific Rim International Conference on Artificial Intelligence PRICAI*, pages 977–984, 2008. doi: 10.1007/978-3-540-89197-0_94.

[77] Elias Boutros Khalil, Bistra N. Dilkina, and Le Song. Scalable diffusion-aware optimization of network topology. In *Proceedings of the 20th ACM SIGKDD*

International Conference on Knowledge Discovery and Data Mining, KDD, pages 1226–1235, 2014. doi: 10.1145/2623330.2623704.

[78] Yao Zhang, Abhijin Adiga, Anil Vullikanti, and B. Aditya Prakash. Controlling propagation at group scale on networks. In *Proceedings of the 15th IEEE International Conference on Data Mining, ICDM*, pages 619–628, 2015. doi: 10.1109/ICDM.2015.59.

[79] Pierluigi Crescenzi, Gianlorenzo D'Angelo, Lorenzo Severini, and Yllka Velaj. Greedily improving our own closeness centrality in a network. *TKDD*, 11(1): 9:1–9:32, 2016. doi: 10.1145/2953882.

[80] Nikos Parotsidis, Evaggelia Pitoura, and Panayiotis Tsaparas. Centrality-aware link recommendations. In *Proceedings of the 9th ACM International Conference on Web Search and Data Mining*, pages 503–512, 2016. doi: 10.1145/2835776. 2835818.

[81] Manos Papagelis. Refining social graph connectivity via shortcut edge addition. *TKDD*, 10(2):12:1–12:35, 2015. doi: 10.1145/2757281.

[82] Q Vera Liao and Wai-Tat Fu. Can you hear me now?: mitigating the echo chamber effect by source position indicators. In *Proceedings of the 17th ACM conference on Computer supported cooperative work & social computing*, pages 184–196. ACM, 2014.

[83] Q Vera Liao and Wai-Tat Fu. Expert voices in echo chambers: effects of source expertise indicators on exposure to diverse opinions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 2745–2754. ACM, 2014.

[84] VG Vinod Vydiswaran, ChengXiang Zhai, Dan Roth, and Peter Pirolli. Overcoming bias to learn about controversial topics. *Journal of the Association for Information Science and Technology*, 66(8):1655–1672, 2015.

[85] Ruben Interian, Jorge R Moreno, and Celso C Ribeiro. Polarization reduction by minimum-cardinality balanced edge additions: Formulations, complexity, and integer programming. In *Proceedings of the 13th Metaheuristics International Conference (MIC)*, 2019.

[86] Kiran Garimella, Gianmarco De Francisci Morales, Aristides Gionis, and Michael Mathioudakis. Reducing controversy by connecting opposing views. In *Proceedings of the 10th ACM International Conference on Web Search and Data Mining*, pages 81–90. ACM, 2017.

[87] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the 42th ACM Symposium on Theory of Computing, STOC*, pages 755–764, 2010. doi: 10.1145/1806689.1806792.

[88] Dorit S. Hochbaum. Approximation algorithms for np-hard problems. *SIGACT News*, 28:40–52, 1997. doi: 10.1145/261342.571216.

[89] Eden Chlamtác, Michael Dinitz, Christian Konrad, Guy Kortsarz, and George Rabanca. The densest k-subhypergraph problem. *SIAM J. Discrete Math.*, 32 (2):1458–1477, 2018. doi: 10.1137/16M1096402.

[90] Pasin Manurangsi and Prasad Raghavendra. A birthday repetition theorem and complexity of approximating dense csps. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 78:1–78:15, 2017. doi: 10.4230/LIPIcs.ICALP.2017.78.

[91] Irit Dinur. Mildly exponential reduction from gap 3sat to polynomial-gap label-cover. *Electronic Colloquium on Computational Complexity, ECCC*, 23:128, 2016.

[92] Benny Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. *SIAM J. Comput.*, 42(5):2008–2037, 2013. doi: 10.1137/120884857.

[93] Gianlorenzo D'Angelo, Martin Olsen, and Lorenzo Severini. Coverage centrality maximization in undirected networks. In *Proceedings of the 33th Conference on Artificial Intelligence, AAAI*, pages 501–508, 2019.

[94] Edith Cohen, Daniel Delling, Thomas Pajor, and Renato F. Werneck. Sketch-based influence maximization and computation: Scaling up with guarantees. In *Proceedings of the 23th ACM International Conference on Information and Knowledge Management, CIKM*, pages 629–638, 2014. doi: 10.1145/2661829.2662077.

[95] Jérôme Kunegis. Konect: the koblenz network collection. In *Proceedings of the 22th International Conference on World Wide Web, WWW*, pages 1343–1350, 2013.

[96] Arnetminer. Arnetminer. http://arnetminer.org, 2015. Accessed: 2015-01-15.

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

[98] Lada A. Adamic and Eytan Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003. doi: 10.1016/S0378-8733(03)00009-1.

[99] Béla Bollobás, Christian Borgs, Jennifer T. Chayes, and Oliver Riordan. Directed scale-free graphs. In *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 132–139, 2003.

[100] Mark EJ Newman. Clustering and preferential attachment in growing networks. *Physical review E*, 64(2):025102, 2001.

[101] Paul Jaccard. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37, 1901.

[102] Paolo Boldi and Sebastiano Vigna. Axioms for centrality. *Internet Mathematics*, 10(3-4):222–262, 2014. doi: 10.1080/15427951.2013.865686.

[103] Huiling Zhang, Md Abdul Alim, My T. Thai, and Hien T. Nguyen. Monitor placement to timely detect misinformation in online social networks. In *Proceedings of the IEEE International Conference on Communications, ICC*, pages 1152–1157, 2015. doi: 10.1109/ICC.2015.7248478.

[104] Marco Amoruso, Daniele Anello, Vincenzo Auletta, and Diodato Ferraioli. Contrasting the spread of misinformation in online social networks. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, AAMAS*, pages 1323–1331, 2017.

[105] Yue Yin, Yevgeniy Vorobeychik, Bo An, and Noam Hazon. Optimal defense against election control by deleting voter groups. *Artif. Intell.*, 259:32–51, 2018. doi: 10.1016/j.artint.2018.02.001.

[106] Bryan Wilder and Yevgeniy Vorobeychik. Defending elections against malicious spread of misinformation. *CoRR*, abs/1809.05521, 2018. URL http://arxiv.org/abs/1809.05521.

[107] Wei Chen, Tian Lin, Zihan Tan, Mingfei Zhao, and Xuren Zhou. Robust influence maximization. In *Proceedings of the 22th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD*, pages 795–804, 2016. doi: 10.1145/2939672.2939745.

[108] Colin McDiarmid. *Concentration*. Springer Science & Business Media, 1998. doi: 10.1007/978-3-662-12788-9_6.

[109] P. Erdős and A. Rényi. On random graphs I. *Publicationes Mathematicae*, 6, 1959.

[110] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D Sivakumar, Andrew Tomkins, and Eli Upfal. Stochastic models for the web graph. In *Proceedings 41th Annual Symposium on Foundations of Computer Science, FOCS*, pages 57–65, 2000. doi: 10.1109/SFCS.2000.892065.

[111] Flavio Chierichetti, Ravi Kumar, Silvio Lattanzi, Alessandro Panconesi, and Prabhakar Raghavan. Models for the compressible web. *SIAM J. Comput.*, 42 (5):1777–1802, 2013. doi: 10.1137/120879828.

[112] Jure Leskovec, Jon M. Kleinberg, and Christos Faloutsos. Graph evolution: Densification and shrinking diameters. *TKDD*, 1(1):2, 2007. doi: 10.1145/1217299.1217301.

Appendix A

Supplemental Material for Chapter 4

Theorem 4.11. Greedy *(Algorithm 2) is a* $\frac{1}{2}(1 - {}^{1}\!/_{e})$*-approximation algorithm for the problem of destructive election control in arbitrary scoring rule voting systems.*

Proof. Let S be the solution found by Greedy (Algorithm 2) in the election control problem and let $S^{\star}$ be the optimal solution. Let $\bar{c}$ and $\hat{c}$ respectively be the candidates that minimize the first term of $\mathbf{E}\left[\mathrm{MoV}_D(S)\right]$ and $\mathbf{E}\left[\mathrm{MoV}_D(S^{\star})\right]$. By Lemma 4.10 we have that

$$\begin{aligned}
&\mathrm{MoV}_D(S) = F\left(\bar{c}, S\right) - F\left(c_{\star}, S\right) - \left(F\left(c, \emptyset\right) - F\left(c_{\star}, \emptyset\right)\right)\\
&= F\left(c_{\star}, \emptyset\right) - F\left(c_{\star}, S\right) - F\left(c, \emptyset\right) + F\left(\bar{c}, \emptyset\right)\\
&\qquad + \sum_{r=1}^{m-1} \sum_{h=r+1}^{m} \sum_{\ell=r+1}^{m} \mathbf{P}(r, \ell) \left|R_{G'}(S, V_{c_{\star}}^{r} \cap V_{\bar{c}}^{h})\right| \left(f(h-1) - f(h)\right)\\
&= F'(c_{\star}, S) - F'(c_{\star}, \emptyset) - F\left(c, \emptyset\right) + F\left(\bar{c}, \emptyset\right)\\
&\qquad + \sum_{r=1}^{m-1} \sum_{h=r+1}^{m} \sum_{\ell=r+1}^{m} \mathbf{P}(r, \ell) \left|R_{G'}(S, V_{c_{\star}}^{r} \cap V_{\bar{c}}^{h})\right| \left(f(h-1) - f(h)\right)
\end{aligned}$$

where c is the most voted candidate before the process. Since $F'(c_{\star}, S) - F'(c_{\star}, \emptyset)$ is an instance of the score in the constructive case we are able to approximate this value, thus we get

$$
\begin{aligned}
\mathrm{MoV}_D(S) &\geq \left(1-\frac{1}{e}\right)\Big[F'(c_\star, S^\star) - F'(c_\star, \emptyset) - F\left(c, \emptyset\right) + F\left(\bar{c}, \emptyset\right) \\
&\quad + \sum_{r=1}^{m-1}\sum_{h=r+1}^{m}\sum_{\ell=r+1}^{m} \mathbf{P}(r,\ell)\,|R_{G'}(S, V_{c_\star}^r \cap V_{\bar{c}}^h)|\,(f(h-1)-f(h))\Big] \\
&\geq \frac{1}{2}\left(1-\frac{1}{e}\right)\Big[F\left(c_\star, \emptyset\right) - F_D(c_\star, S^\star) - F\left(c, \emptyset\right) + F\left(\bar{c}, \emptyset\right) \\
&\quad + \sum_{r=1}^{m-1}\sum_{h=r+1}^{m}\sum_{\ell=r+1}^{m} \mathbf{P}(r,\ell)\,|R_{G'}(S^\star, V_{c_\star}^r \cap V_{\hat{c}}^h)|\,(f(h-1)-f(h)) \\
&\quad + \sum_{r=1}^{m-1}\sum_{h=r+1}^{m}\sum_{\ell=r+1}^{m} \mathbf{P}(r,\ell)\,|R_{G'}(S, V_{c_\star}^r \cap V_{\bar{c}}^h)|\,(f(h-1)-f(h)) + F\left(\hat{c}, \emptyset\right) - F\left(\hat{c}, \emptyset\right)\Big] \\
&\geq \frac{1}{2}\left(1-\frac{1}{e}\right)\Big[\mathrm{MoV}_D(A^\star) + F\left(\bar{c}, \emptyset\right) \\
&\quad + \sum_{r=1}^{m-1}\sum_{h=r+1}^{m}\sum_{\ell=r+1}^{m} \mathbf{P}(r,\ell)\,|R_{G'}(S^{,}V_{c_\star}^r \cap V_{\bar{c}}^h)|\,(f(h-1)-f(h)) - F\left(\hat{c}, \emptyset\right)\Big]
\end{aligned}
$$

By definition of $\bar{c}$ we have that

$$
\begin{aligned}
F\left(\bar{c}, S\right) &= F\left(\bar{c}, \emptyset\right) + \sum_{r=1}^{m-1}\sum_{h=r+1}^{m}\sum_{\ell=r+1}^{m} \mathbf{P}(r,\ell)\,|R_{G'}(S, V_{c_\star}^r \cap V_{\bar{c}}^h)|\,(f(h-1)-f(h)) \\
&\geq F\left(\hat{c}, \emptyset\right) + \sum_{r=1}^{m-1}\sum_{h=r+1}^{m}\sum_{\ell=r+1}^{m} \mathbf{P}(r,\ell)\,|R_{G'}(S, V_{c_\star}^r \cap V_{\hat{c}}^h)|\,(f(h-1)-f(h)) = F\left(\hat{c}, S\right)
\end{aligned}
$$

This implies that

$$
\begin{aligned}
F\left(\bar{c}, \emptyset\right) - F\left(\hat{c}, \emptyset\right) &+ \sum_{r=1}^{m-1}\sum_{h=r+1}^{m}\sum_{\ell=r+1}^{m} \mathbf{P}(r,\ell)\,|R_{G'}(S, V_{c_\star}^r \cap V_{\bar{c}}^h)|\,(f(h-1)-f(h)) \\
&\geq \sum_{r=1}^{m-1}\sum_{h=r+1}^{m}\sum_{\ell=r+1}^{m} \mathbf{P}(r,\ell)\,|R_{G'}(S, V_{c_\star}^r \cap V_{\hat{c}}^h)|\,(f(h-1)-f(h)) \geq 0
\end{aligned}
$$

and therefore

$$\mathrm{MoV}_D(S) \geq \frac{1}{2}\left(1-\frac{1}{e}\right)\mathrm{MoV}_D(S^*).$$

□

Appendix B

Supplemental Material for Chapter 5

B.1 Reducing Densest-k-Subgraph to PLTR

Theorem 5.1. *An α-approximation to the election control problem in PLTR gives an $\alpha\beta$-approximation to the* Densest-k-Subgraph *problem, for a positive constant $\beta < 1$.*

Proof. We prove the hardness of approximation by reducing our problem from the well-known Densest-k-Subgraph (DkS) problem, which is hard to approximate within any constant factor under different hypothesis [21, 87]. Given an undirected graph $G = (V, E)$ and an integer k, Densest-k-Subgraph is the problem of finding the subgraph induced by a subset of V of size k with the highest number of edges.

The reduction works as follows: Consider the PLTR problem on G, where each undirected edge $\{u, v\}$ is replaced with two directed edges (u, v) and (v, u). Let us consider m candidates and let us assume that all nodes initially have null probability of voting for all the candidates but one, different from $c_\star$, that we denote as $\hat{c}$. Formally we have that, $\pi_v(\hat{c}) = 1$ and $\pi_v(c_i) = \pi_v(c_\star) = 0$ for each $c_i \neq \hat{c}$ and for each $v \in V$. Assign to each edge $(u, v) \in E$ a weight $b_{uv} = \frac{1}{n^\gamma}$, for any fixed constant $\gamma \geq 4$ and $n = |V|$.

We show the reduction considering the problem of maximizing the score, because in the instance considered in the reduction the MoV is exactly equal to twice the score.

In fact, with some algebra it is possible to show that the score of $\hat{c}$ after PLTR starting from any initial set S is

$$\begin{aligned}
F\left(\hat{c}, S\right) &= \sum_{v \in V} \tilde{\pi}_v(\hat{c}) = \sum_{v \in V \setminus A} \pi_v(\hat{c}) + \sum_{v \in A} \tilde{\pi}_v(\hat{c}) \\
&= |V| - |A| + \sum_{v \in A} \frac{1}{1 + \sum_{u \in A \cap N_v^-} \frac{1}{n^\gamma}} \\
&= |V| - |A| + \sum_{v \in A} \frac{1}{1 + \frac{1}{n^{\gamma+1}} |A \cap N_v^-|} \\
&= |V| - |A| + \sum_{v \in A} \frac{1}{1 + \frac{1}{n^{\gamma+1}} |A \cap N_v^-|} \\
&= |V| - \sum_{v \in A} \left(1 - \frac{1}{1 + \sum_{u \in A \cap N_v^-} \frac{1}{n^\gamma}} \right) \\
&= |V| - \sum_{v \in A} \left(\frac{\sum_{u \in A \cap N_v^-} \frac{1}{n^\gamma}}{1 + \sum_{u \in A \cap N_v^-} \frac{1}{n^\gamma}} \right) \\
&= |V| - F\left(c_\star, S\right),
\end{aligned}$$

because $(\sum_{u \in A \cap N_v^-} \frac{1}{n^\gamma})/(1 + \sum_{u \in A \cap N_v^-} \frac{1}{n^\gamma}) = \tilde{\pi}_v(c_\star)$ and $\pi_v(c_\star) = 0$ for each $v \in V$. Thus, according to the definition of MoV in Equation (6), we have that

$$\text{MoV}(S) = |V| - (|V| - F\left(c_\star, S\right) - F\left(c_\star, S\right)) = 2F\left(c_\star, S\right).$$

To compute the expected final score of the target candidate we average its score in all live-edge graphs in $\mathcal{G}$, according to Formula (5.3). In our reduction, the empty live-edge graph $G'_\emptyset = (V, \emptyset)$ is sampled *with high probability*, i.e., with probability at least $1 - \frac{1}{n^{\gamma-2}}$:

$$\begin{aligned}
\mathbf{P}\left(G'_\emptyset\right) &= \prod_{v \in V} \left(1 - \sum_{u \in N_v^-} b_{uv} \right) = \prod_{v \in V} \left(1 - \frac{|N_v^-|}{n^\gamma} \right) \\
&\geq \prod_{v \in V} \left(1 - \frac{1}{n^{\gamma-1}} \right) = \left(1 - \frac{1}{n^{\gamma-1}} \right)^n \\
&\overset{(a)}{=} \sum_{i=0}^{n} \binom{n}{i} (1)^{n-i} \left(\frac{-1}{n^{\gamma-1}} \right)^i = \sum_{i=0}^{n} \binom{n}{i} \frac{(-1)^i}{n^{i(\gamma-1)}} \\
&\overset{(b)}{\geq} \binom{n}{0} - \binom{n}{1} \frac{1}{n^{\gamma-1}} + \sum_{i=2}^{\lfloor n/2 \rfloor} \left(\binom{n}{i} \frac{1}{n^{2i(\gamma-1)}} - \binom{n}{i+1} \frac{1}{n^{(2i+1)(\gamma-1)}} \right)
\end{aligned}$$

$$\overset{(c)}{\geq} 1 - \frac{1}{n^{\gamma-2}},$$

where in (a) we used the binomial expansion, (b) is due to last negative term in the lhs that does not appear in the rhs when n is even, and (c) is due to

$$\binom{n}{i}\frac{1}{n^{2i(\gamma-1)}} \geq \binom{n}{i+1}\frac{1}{n^{(2i+1)(\gamma-1)}},$$

for any $\gamma \geq 2$. Since $\mathbf{P}\left(G'_\emptyset\right) \leq 1$, then $\mathbf{P}\left(G'_\emptyset\right) = \Theta(1)$. Moreover, $\sum_{G' \neq G'_\emptyset} \mathbf{P}\left(G'\right) = \mathcal{O}\left(\frac{1}{n^{\gamma-2}}\right)$.

The score obtained by $c_\star$ in a live-edge graph G' starting from any initial set of seed nodes S is

$$\begin{aligned} F_{G'}(c_\star, S) &= \sum_{v \in V \setminus A} \pi_v(c_\star) + \sum_{v \in A} \tilde{\pi}_v(c_\star) \\ &= \sum_{v \in R_{G'}(S)} \frac{\pi_v(c_\star) + \sum_{u \in R_{G'}(S) \cap N_v^-} \frac{1}{n^\gamma}}{1 + \sum_{u \in R_{G'}(S) \cap N_v^-} \frac{1}{n^\gamma}} \\ &= \Theta\left(\frac{1}{n^\gamma} \sum_{v \in R_{G'}(S)} |R_{G'}(S) \cap N_v^-|\right), \end{aligned}$$

since $1 \leq 1 + \sum_{u \in R_{G'}(S) \cap N_v^-} \frac{1}{n^\gamma} \leq 2$ for each $v \in R_{G'}(S)$. Note that $\sum_{v \in R_{G'}(S)} |R_{G'}(S) \cap N_v^-|$ is equal to the number of edges of the subgraph induced by the set $R_{G'}(S)$ of nodes reachable from S in G', which is not greater than n^2, and thus $F_{G'}(c_\star, S) = \mathcal{O}\left(\frac{1}{n^{\gamma-2}}\right)$.

Note that in the empty live-edge graph $G'_\emptyset$ the set $R_{G'_\emptyset}(S)$ at the end of LTM is equal to S, since the graph has no edges. Thus

$$F_{G'_\emptyset}(c_\star, S) = \frac{1}{n^\gamma} \cdot \sum_{v \in S} \frac{|S \cap N_v^-|}{1 + \sum_{u \in S \cap N_v^-} \frac{1}{n^\gamma}}$$

and since the denominator is, again, bounded by two constants we have that

$$F_{G'_\emptyset}(c_\star, S) = \Theta\left(\frac{\sum_{v \in S} |S \cap N_v^-|}{n^\gamma}\right) = \Theta\left(\frac{\mathrm{SOL}_{DkS}(S)}{n^\gamma}\right),$$

where $\mathrm{SOL}_{DkS}(S) := \sum_{v \in S} |S \cap N_v^-|$ is the number of edges of the subgraph induced by S, i.e., the value of the objective function of DkS for solution S.

Thus, the expected final score of the target candidate is

$$\begin{aligned}F\left(c_\star, S\right) &= \sum_{G' \in \mathcal{G}} F_{G'}(c_\star, S) \cdot \mathbf{P}(G') \\ &= F_{G'_\emptyset}(c_\star, S) \cdot \mathbf{P}(G'_\emptyset) + \sum_{G' \neq G'_\emptyset} F_{G'}(c_\star, S) \cdot \mathbf{P}(G').\end{aligned}$$

Since $F_{G'}(c_\star, S)$ and $\sum_{G' \neq G'_\emptyset} \mathbf{P}\left(G'\right)$ are in $\mathcal{O}\left(\frac{1}{n^{\gamma-2}}\right)$, then

$$\begin{aligned}\sum_{G' \neq G'_\emptyset} F_{G'}(c_\star, S) \cdot \mathbf{P}(G') &= \mathcal{O}\left(\frac{1}{n^{\gamma-2}}\right) \sum_{G' \neq G'_\emptyset} \mathbf{P}(G') \\ &= \mathcal{O}\left(\frac{1}{n^{2(\gamma-2)}}\right) = \mathcal{O}\left(\frac{\mathrm{SOL}_{DkS}(S)}{n^\gamma}\right),\end{aligned}$$

for any $\gamma \geq 4$. Thus

$$F\left(c_\star, S\right) = \Theta\left(\frac{\mathrm{SOL}_{DkS}(S)}{n^\gamma}\right) \cdot \Theta(1) + \mathcal{O}\left(\frac{\mathrm{SOL}_{DkS}(S)}{n^\gamma}\right)$$

which means that $F\left(c_\star, S\right) = \Theta\left(\frac{\mathrm{SOL}_{DkS}(S)}{n^\gamma}\right)$. We apply the Bachmann-Landau definition of Θ notation: There exist three positive constants n_0, β_1, and β_2 such that, for all $n > n_0$,

$$\beta_1 \frac{\mathrm{SOL}_{DkS}(S)}{n^\gamma} \leq F\left(c_\star, S\right) \leq \beta_2 \frac{\mathrm{SOL}_{DkS}(S)}{n^\gamma}.$$

Note that, in this case, the constants n_0, β_1, and β_2 do not depend on the specific instance.

Since the previous bounds hold for any set S we also have that $\beta_1 \frac{\mathrm{OPT}_{DkS}}{n^\gamma} \leq \mathrm{OPT}$, where OPT is the value of an optimal solution for PLTR and OPT_{DkS} is the value of an optimal solution for DkS.

Suppose that there exists an α-approximation algorithm for PLTR, i.e., an algorithm that finds a set S such that the value of its solution is $\mathrm{MoV}(S) = 2F\left(c_\star, S\right) \geq \alpha \cdot \mathrm{OPT}$. Then,

$$\frac{\alpha}{2} \cdot \beta_1 \frac{\mathrm{OPT}_{DkS}}{n^\gamma} \leq \frac{\alpha}{2} \cdot \mathrm{OPT} \leq F\left(c_\star, S\right) \leq \beta_2 \frac{\mathrm{SOL}_{DkS}(S)}{n^\gamma}.$$

Thus $\mathrm{SOL}_{DkS}(S) \geq \frac{\alpha}{2} \frac{\beta_1}{\beta_2} \mathrm{OPT}_{DkS}$, i.e., the solution is an $\alpha\beta$-approximation to DkS, with $\beta := \frac{\beta_1}{2\beta_2}$. □

Appendix C

Supplemental Material for Chapter 6

C.1 Deferred Proofs for Section 6.1

Approximating $\Phi^{\geq \ell}(\mathcal{S})$ and Ψ.

We use the following algorithm (Algorithm 7) for approximating $f \in \{\Psi, \Phi^{\geq 0}, \ldots, \Phi^{\geq \nu}\}$.

Algorithm 7 approx$(f, \mathcal{S}, \mathcal{I}, \nu, \epsilon, \delta)$

```
                    ▷ Note that, if $f = \Psi$, then $\mathcal{S} \subseteq V \times \{0\}$, otherwise $\mathcal{S} \subseteq \hat{V}$.
$T \leftarrow |V|^2 \ln(1/\delta)/\epsilon^2$
for $t = 1, \ldots, T$ do
    Sample outcome profile $\mathcal{X}$
    if $f = \Psi$ then
        Compute $R \leftarrow \rho^{(0)}_{X_0}(\mathcal{S})$ and $n_t \leftarrow |(R \cap \bigcup_{j=1}^{\nu-1} V^j_{\mathcal{X}}) \cup \bigcup_{j=\nu}^{\mu} V^j_{\mathcal{X}}|$
    else
        Compute $\mathcal{R} \leftarrow (\rho^{(i)}_{\mathcal{X}_i}(I_i \cup S_i))_{i \in [\mu]}$ and $n_t \leftarrow \mathrm{NoSM}_{\mu,\nu}(\mathcal{R} \setminus (\cup_{j=0}^{\ell-1} V^j_{\mathcal{X}}))$
return $\frac{1}{T} \sum_{t=1}^{T} n_t$
```

We show the following lemma. As a condition for the lemma, we have the requirement that $f(\mathcal{S}) \geq 1$ for the set $\mathcal{S}$ that we evaluate f on. We argue at the end of this section, see Lemma C.1 that we can assume $\Phi^{\geq \ell}(\mathcal{S}) \geq 1$ for any $\ell \in [0, \nu]$ and $\mathcal{S}$ as well as $\Psi(\mathcal{T}) \geq 1$ for any $\mathcal{T}$ at the cost of an arbitrarily small ϵ in the approximation guarantee.

Lemma 6.2. *Let $f \in \{\Psi, \Phi^{\geq 0}, \ldots, \Phi^{\geq \nu}\}$ and let $\mathcal{S}$ be such that $f(\mathcal{S}) \geq 1$. Let $\tilde{f}(\mathcal{S}) :=$ $\mathrm{approx}(f, \mathcal{S}, \mathcal{I}, \nu, \epsilon, \delta)$ for some $0 < \delta \leq 1/2$ and $0 < \epsilon < 1$, then $\tilde{f}(\mathcal{S})$ is a $(1 \pm \epsilon)$-approximation of $f(\mathcal{S})$ with probability at least $1 - \delta$.*

Proof. The proof is very similar to the proof of Proposition 4.1 in [11], it is a straightforward application of a Chernoff bound, we use Theorem 2.3 from [108] here. Let us define T random variables, one for each iteration of the algorithm, $Y_1, \ldots, Y_T$ by $Y_t := n_t/|V|$. Note that the Y_t are independent and $Y_t \in [0,1]$. Let $S_T := \sum_{t=1}^{T} Y_t$ and $\mu := \mathrm{E}[S_T]$, then $S_T = T \cdot \tilde{f}(\mathcal{S})/|V|$ and $\mu = T \cdot f(\mathcal{S})/|V|$. Thus, setting $\gamma := \epsilon f(\mathcal{S})/|V|$, the Chernoff bound yields

$$\mathbf{P}[|f(\mathcal{S}) - \tilde{f}(\mathcal{S})| \geq \epsilon f(\mathcal{S})] = \mathbf{P}[|S_T - \mu| \geq T\gamma] \leq 2e^{-2T\gamma^2} = 2e^{-\frac{2T\epsilon^2 f(\mathcal{S})^2}{|V|^2}} \leq \delta,$$

since $T = |V|^2 \ln(1/\delta)/\epsilon^2$ and $f(\mathcal{S}) \geq 1$. □

Lemma 6.3. *Let f and $\tilde{f}$ be as above for some $0 < \epsilon \leq 1$. Let $U := \{\tau \subseteq D_f, |\tau| = \lambda(f)\}$, $\mathcal{S} \subseteq D_f$ with $|\mathcal{S}| \leq B - \lambda(f)$, and let $\mathcal{S}^*$ denote a set maximizing f of size B. Then, either*

$$f(\mathcal{S}) \geq \left(1 - \frac{1}{e}\right) \cdot f(\mathcal{S}^*) \quad \text{or} \quad f(\mathcal{S} \cup \tilde{\tau}) - f(\mathcal{S}) \geq (1 - \epsilon) \cdot (f(\mathcal{S} \cup \tau^*) - f(\mathcal{S})),$$

where $\tau^* := \arg\max\{f(S \cup \tau) : \tau \in U\}$, and $\tilde{\tau} := \arg\max\{\tilde{f}(S \cup \tau) : \tau \in U\}$.

Proof. We distinguish two cases. First, assume that $f(\mathcal{S} \cup \tau^*) - f(\mathcal{S}) \leq f(\mathcal{S}^*)/(e \cdot \binom{B}{\nu - \ell})$. If $f = \Phi^{\ell}$ for some ℓ, Lemma 6.1 yields that τ^* satisfies

$$f(\mathcal{S} \cup \tau^*) - f(\mathcal{S}) \geq \frac{1}{\binom{B}{\lambda(f)}} \cdot (f(\mathcal{S}^*) - f(\mathcal{S})). \tag{C.1}$$

For $f = \Psi$, we note that $\lambda(f) = 1$ and thus $\binom{B}{\lambda(f)} = B$, so we get inequality (C.1) by the submodularity of Ψ. Thus, in both case by combining the two inequalities, we get $f(\mathcal{S}) \geq (1 - 1/e) \cdot f(\mathcal{S}^*)$, which concludes this case.

On the other hand, assume $f(\mathcal{S} \cup \tau^*) - f(\mathcal{S}) > f(\mathcal{S}^*)/(e \cdot \binom{B}{\lambda(f)})$. Using the approximation guarantee of $\tilde{f}$, the definition of $\tilde{\tau}$, and again the approximation guarantee, we get

$$
\begin{aligned}
f(\mathcal{S} \tilde{\cup} \tau) - f(\mathcal{S}) &\geq \frac{1-\epsilon'}{1+\epsilon'} f(\mathcal{S} \cup \tau^*) - f(\mathcal{S}) \\
&= f(\mathcal{S} \cup \tau^*) - f(\mathcal{S}) - \frac{2\epsilon'}{1+\epsilon'} f(\mathcal{S} \cup \tau^*) \\
&= (1-\epsilon) \cdot (f(\mathcal{S} \cup \tau^*) - f(\mathcal{S})) + \frac{\epsilon(1+\epsilon') - 2\epsilon'}{1+\epsilon'} \cdot f(\mathcal{S} \cup \tau^*) - \epsilon \cdot f(\mathcal{S}).
\end{aligned}
$$

Thus, it remains to argue that the latter two summands are non-negative. From the case assumption and the optimality of $\mathcal{S}^*$, we have $f(\mathcal{S}) \leq f(\mathcal{S} \cup \tau^*)/(1 + 1/(e \cdot \binom{B}{\lambda(f)}))$ and thus the above latter two summands can be lower bounded by

$$
f(S \cup \tau^*) \cdot \Big(\frac{\epsilon(1+\epsilon') - 2\epsilon'}{1+\epsilon'} - \frac{\epsilon}{(1 + 1/(e \cdot \binom{B}{\lambda(f)}))} \Big).
$$

The latter is non-negative, since

$$
(\epsilon(1+\epsilon') - 2\epsilon')\Big(1 + \frac{1}{e \cdot \binom{B}{\lambda(f)}}\Big) - \epsilon(1+\epsilon') = \epsilon'(1+\epsilon') - \frac{2\epsilon'}{e \cdot \binom{B}{\lambda(f)}} \geq 0
$$

by the choice of $\epsilon' := \epsilon/(e \cdot \binom{B}{\lambda(f)})$ and $0 < \epsilon \leq 1$. This concludes the proof. □

Lower Bound on $\Phi^{\geq \ell}$ and Ψ.

Our goal in this section is to argue that there is a transform τ that takes an instance $(G, \mathcal{P}, \mathcal{I}, B)$ of the μ-ν-Balance problem and outputs a (slightly modified) instance $(G', \mathcal{P}, \mathcal{I}', B) := \tau(G, \mathcal{P}, \mathcal{I}, B)$ such that the function $f'(\mathcal{S})$ is at least 1 for every argument $\mathcal{S}$ in the new instance for any $f \in \{\Phi^{\geq 0}, \ldots, \Phi^{\geq \nu}, \Psi\}$. Moreover, given an approximation algorithm for Φ with approximation ratio α, we will show that applying this algorithm on the transformed instance $\tau(G, \mathcal{P}, \mathcal{I}, B)$ leads to a solution of approximation ratio at least $\alpha - \epsilon$ for the original instance, for any $\epsilon > 0$.

The transform τ is defined as follows. Obtain G' by adding an isolated node v to G and extend $\mathcal{I}$ to $\mathcal{I}'$ by adding v to I_i for every $i \in [\nu]$. Now clearly, for every solution $\mathcal{S}$, it holds that $f'(\mathcal{S}) = f(\mathcal{S}) + 1 \geq 1$, where the 1 originates from the additional node v that is initially covered by $\nu \geq \ell$ campaigns. Moreover, we get the following lemma.

Lemma C.1. *Let $\epsilon > 0$. Then, for instances $(G, \mathcal{P}, \mathcal{I}, B)$ with $B \geq 2\nu/\epsilon$, the following holds: Let $\mathcal{S}'$ be a solution in $(G', \mathcal{P}, \mathcal{I}', B) := \tau(G, \mathcal{P}, \mathcal{I}, B)$ such that $\Phi'(\mathcal{S}') \geq \alpha \cdot \Phi'(\mathcal{S}'^*)$, where $\mathcal{S}'^*$ denotes an optimal solution for maximizing Φ' in the new instance $(G', \mathcal{P}, \mathcal{I}', B)$. Then $\mathcal{S} := \mathcal{S}' \setminus \{v\}$ satisfies $\Phi(\mathcal{S}) \geq (\alpha - \epsilon) \cdot \Phi(\mathcal{S}^*)$, where $\mathcal{S}^*$ denotes an optimal solution for maximizing Φ in the original instance $(G, \mathcal{P}, \mathcal{I}, B)$.*

Proof. First note that $\Phi(\mathcal{S}^*) \geq \lfloor B/\nu \rfloor \geq B/\nu - 1 \geq 1/\epsilon$ or equivalently $1 \leq \epsilon\Phi(\mathcal{S}^*)$. This yields the claim, since

$$\Phi(\mathcal{S}) = \Phi'(\mathcal{S}) - 1 \geq \alpha \cdot \Phi'(\mathcal{S}'^*) - 1 \geq \alpha \cdot \Phi(\mathcal{S}^*) - 1 \geq (\alpha - \epsilon) \cdot \Phi^{\geq \ell}(\mathcal{S}^*).$$

□

Maximizing $\Phi^{\geq \nu-1}$ and Ψ.

Our goal here is to show that the standard greedy hill climbing algorithm, we refer to it as Greedy$(f, \epsilon, \delta, \mathcal{I}, \nu, B)$ (Algorithm 8), can be applied in order to approximate both $\Phi^{\geq \nu-1}$ and Ψ to within a factor of $1 - 1/e - \epsilon$ for any $0 < \epsilon < 1$ with probability at least $1 - \delta$ for any $0 < \delta \leq 1/2$. We first formally prove that these functions are submodular.

Lemma C.2. *The functions Ψ and $\Phi^{\geq \nu-1}$ are monotone and submodular.*

Proof. The monotonicity of Ψ and $\Phi^{\geq \nu-1}$ is straightforward. We argue the submodularity of Ψ ($\Phi^{\geq \nu-1}$) in a similar way as we argued in the proof of Lemma 6.1. To this end, let $D(\Psi) = V \times \{0\}$ and $D(\Phi^{\geq \nu-1}) = \hat{V} = V \times [\mu]$ denote the domain of Ψ and $\Phi^{\geq \nu-1}$, respectively, and let $\mathcal{S}$ and $\mathcal{S}'$ be subsets of $D(\Psi)$ ($D(\Phi^{\geq \nu-1})$) such that $\mathcal{S} \subseteq \mathcal{S}'$, and let τ be an element of the domain $D(\Psi)$ ($D(\Phi^{\geq \nu-1})$). Furthermore, let $\mathcal{X}$ be an outcome profile w.r.t. the correlated (heterogeneous) probability distributions. Lastly, let $v \in V \setminus V_{\mathcal{X}}^0$ ($v \in V \setminus \bigcup_{j=0}^{\nu-2} V_{\mathcal{X}}^j$) be a node that can contribute to the value of Ψ ($\Phi^{\geq \nu-1}$).[1] We denote by $\mathbf{1}_{\mathcal{X}}^{\mathcal{S}}(v)$ the indicator function that is 1 if v contributes to Ψ ($\Phi^{\geq \nu-1}$) in outcome profile $\mathcal{X}$ with initial seed sets $\mathcal{I}$ and additional seed sets $\mathcal{S}$ and 0 otherwise. We now argue that the following inequality holds:.

$$\mathbf{1}_{\mathcal{X}}^{\mathcal{S}' \cup \tau}(v) - \mathbf{1}_{\mathcal{X}}^{\mathcal{S}'}(v) \leq \mathbf{1}_{\mathcal{X}}^{\mathcal{S} \cup \tau}(v) - \mathbf{1}_{\mathcal{X}}^{\mathcal{S}}(v) \tag{C.2}$$

[1] Recall that $V_{\mathcal{X}}^j$ is the set of nodes that was reached by j campaigns from seed sets $\mathcal{I}$.

Algorithm 8 Greedy$(f, \epsilon, \delta, \mathcal{I}, \nu, B)$

```
$\delta' \leftarrow \delta/(B|\hat{V}|)$
$\epsilon' \leftarrow \epsilon/(eB)$
$\mathcal{S} \leftarrow \emptyset$
while $|\mathcal{S}| \leq B$ do
    Compute $v \leftarrow \arg\max\{\text{approx}(f, \mathcal{S} \cup \{v\}, \mathcal{I}, \nu, \epsilon', \delta') : v \in D_f\}$, set $\mathcal{S} \leftarrow \mathcal{S} \cup \{v\}$
return $\mathcal{S}$
```

Note that the right-hand side cannot be negative by monotonicity and that, if the left-hand side is positive for Ψ ($\Phi^{\geq \nu-1}$), then it must hold that the node v is reached by a subset $M \subseteq [\mu]$ of campaigns from $\mathcal{I}$ with $|M| \in [1, \nu - 1]$ ($|M| = \nu - 1$). Furthermore, the node v is not reached by campaign 0 (is not reached by a campaign $j \in [\mu] \setminus M$) from $\mathcal{S}'$, but it is reached by campaign 0 (campaign j) from τ. Now, observe that $\mathcal{S} \subseteq \mathcal{S}'$ and thus the node v is not reached by campaign 0 (by campaign j) from $\mathcal{S}$ neither. Hence it follows that the right-hand side is also positive. Taking the expected value on both sides of Eq. (C.2) yields that $\Psi(\mathcal{S}' \cup \tau) - \Psi(\mathcal{S}') \leq \Psi(\mathcal{S} \cup \tau) - \Psi(\mathcal{S})$ ($\Phi^{\geq \nu-1}(\mathcal{S}' \cup \tau) - \Phi^{\geq \nu-1}(\mathcal{S}') \leq \Phi^{\geq \nu-1}(\mathcal{S} \cup \tau) - \Phi^{\geq \nu-1}(\mathcal{S})$) due to linearity of expectation. This establishes submodularity and concludes the proof. □

We now recall the following classical result concerning the greedy algorithm for maximizing a submodular function:

Lemma C.3 (Theorem 3.9 in [88])**.** *The greedy hill-climbing algorithm, that at each step picks an element that leads to an increment being within factor β of the optimal increment possible, achieves an approximation ratio of at least* $1 - (1 - \beta/B)^B > 1 - {}^1\!/_{e^\beta}$.

We have seen that both $\Phi^{\geq \nu-1}$ and Ψ can be approximated within a $(1 \pm \epsilon)$-factor using the approx-routine. In Lemma 6.3 we argued that using the approximations we can find an element v (or a set τ of cardinality $\lambda(f) = 1$) that when added to $\mathcal{S}$ leads to a progress of at least a factor of $(1 - \epsilon)$ of the maximal progress possible. We prove Lemma 6.4.

Lemma 6.4. *Let $f \in \{\Phi^{\geq \nu-1}, \Psi\}$ and let $0 < \epsilon < 1$ and $0 < \delta \leq 1/2$. With probability at least $1 - \delta$,* Greedy$(f, \epsilon, \delta, \mathcal{I}, \nu, B)$ *returns $\mathcal{S}$ satisfying $f(\mathcal{S}) \geq (1 - {}^1\!/_e - \epsilon) \cdot f(\mathcal{S}^*)$, where $\mathcal{S}^*$ is an optimal solution of size B to maximizing f.*

Proof. The union bound over all at most $B|\hat{V}|$ calls to approx, yields that, with probability at least $1-\delta$, each call resulted in a $1 \pm \epsilon'$-approximation. Then Lemma 6.3 applied to f guarantees that after each iteration i either an element v is picked such that the increment using v is at least a $(1-\epsilon)$-fraction of the optimal increment possible in this iteration or the current set $\mathcal{S}^i$ is already a $(1-1/e)$-approximation of the optimum set $\mathcal{S}^*_{\nu-1}$. In the latter case the lemma is fulfilled by the monotonicity of f. In the former case we get an $\mathcal{S}$ having an approximation ratio of at least $1-(1-(1-\epsilon)/B)^B \geq 1-1/e^{1-\epsilon}$ according to Lemma C.3. Since $1-\frac{1}{e^{1-\epsilon}} \geq (1-\epsilon)\cdot\left(1-\frac{1}{e}\right) \geq 1-\frac{1}{e}-\epsilon$, this concludes the proof. □

C.2 Deferred Proofs for Section Reducing Densest-k-Sub-d-hypergraph to μ-ν-Balance.

We first define the Multicld-Edge Densest-Sub-d-hypergraph problem which is closely related to the Densest-k-Sub-d-hypergraph problem.

Given a d-Regular Hypergraph $G=(V,E)$, an integer k representing the budget, the Multicld-Edge Densest-Sub-d-hypergraph problem aims at finding a set $S\subseteq V$ with $|S|\leq k$ and a coloring function $\varphi : S\to[d]$, s.t. $|E_\varphi(S)|$ is maximal, where $E_\varphi(S) := \{e\in E : e=(v_1,\dots,v_d)\subseteq S \ \wedge \ \varphi(v_i)\neq\varphi(v_j), \forall i\neq j\}$.

Problem Multicld-Edge Densest-Sub-d-hypergraph will be of interest to us due to the following results. We first prove a lemma showing the existence of an assignment φ' such that at least a fraction p of the hyperedges in the induced sub-hypergraph of a set S have differently colored endpoints.

Lemma C.4. *Let $G=(V,E)$ be a d-regular hypergraph. For any set $S\subseteq V$, there is an assignment $\varphi' : S\to[d]$ s.t. $|E_{\varphi'}(S)|\geq p\cdot|E(S)|$ where $p=d!/d^d$.*

Proof. Let $S\subseteq V$. Consider the probabilistic procedure in which, for each node, we assign a color from $[d]$ uniformly at random and independently of the other nodes. This procedures yields a coloring φ. For any $e=(v_1,\dots,v_d)$ in S, the probability that $\varphi(v_i)\neq\varphi(v_j)$ for all $i\neq j$ is p. This property is guaranteed if and only if $(\varphi(v_1),\dots,\varphi(v_d))$ corresponds to one of the $d!$ permutations of $[d]$. In total, there are d^d ways of coloring e. Hence, the expected value of $|E_\varphi(S)|$ is $p\cdot|E(S)|$. Consequently, the function φ' that maximizes $|E_{\varphi'}(S)|$ satisfies $|E_{\varphi'}(S)|\geq p\cdot|E(S)|$. □

This leads to the following corollary.

Corollary C.5. *Denoting with* $\mathrm{DKSH}^*_d(G,k)$ *and* $\mathrm{MCD}^*_d(G,k)$ *the value of the optimal solution for* Densest-k-Sub-d-hypergraph *and* Multicld-Edge Densest-Sub-d-hypergraph *on* (G,k)*, respectively, we have that* $\mathrm{DKSH}^*_d(G,k) \leq \mathrm{MCD}^*_p(G,k)/p$*, where* $p = d!/d^d$*.*

Proof. Let S be a set that achieves $\mathrm{DKSH}^*_d(G,k) = |E(S)|$, then $\mathrm{DKSH}^*_d(G,k) = |E(S)| \leq |E_{\varphi'}(S)|/p \leq \mathrm{MCD}^*_d(G,k)/p$, where φ' is as in Lemma C.4. □

Recall that we fixed a μ-ν-Balance instance $P = (\overline{G} = (\overline{V}, \overline{A}), \mathcal{P}, \mathcal{I}, B)$ resulting from the transform τ as image of an Densest-k-Sub-d-hypergraph instance $Q = (G = (V,E), k)$. In what follows nodes in $V_\square$ (resp. $V_\circledcirc$) are called rectangle-nodes (resp. circle-nodes).

Lemma 6.5. *The following statements hold:*

1. *An optimal solution to* Φ *also maximizes* $\Phi_\circledcirc$*, i.e.,* $\Phi_\circledcirc(\mathcal{S}^*_\circledcirc) = \Phi_\circledcirc(\mathcal{S}^*)$*.*
2. *It holds that* $\Phi_\circledcirc(\mathcal{S}^*_\circledcirc) \geq l \cdot p \cdot \mathrm{DKSH}^*_d$*, where* DKSH^*_d *is the optimal value of* Densest-k-Sub-d-hypergraph *in* Q *and* $p = d!/d^d$*.*
3. *Given* $\mathcal{S} \in \Sigma$*, we can, in polynomial time, build a feasible solution* S *of* Q *such that* $|E(S)| \geq \Phi_\circledcirc(\mathcal{S})/(\lambda l)$*.*

Proof.

1. We can w.l.o.g. assume that $\mathcal{S}^* \cap V_\circledcirc = \emptyset$ and $\mathcal{S}^*_\circledcirc \cap V_\circledcirc = \emptyset$. Then, it follows that both $\Phi_\circledcirc(\mathcal{S}^*)$ and $\Phi_\circledcirc(\mathcal{S}^*_\circledcirc)$ are multiples of l. Now, assume for the purpose of contradiction that $\Phi_\circledcirc(\mathcal{S}^*_\circledcirc) > \Phi_\circledcirc(\mathcal{S}^*)$. Then, $\Phi_\circledcirc(\mathcal{S}^*_\circledcirc) \geq \Phi_\circledcirc(\mathcal{S}^*) + l$ which leads to

$$\Phi(\mathcal{S}^*) = \Phi_\circledcirc(\mathcal{S}^*) + \Phi_\square(\mathcal{S}^*) \leq \Phi_\circledcirc(\mathcal{S}^*_\circledcirc) - l + |V| < \Phi(\mathcal{S}^*_\circledcirc),$$

   using that $l > |V|$. This is a contradiction to $\mathcal{S}^*$ being optimal.

2. Let (S^*, φ^*) be an optimal solution to the Multicld-Edge Densest-Sub-d-hypergraph problem induced by Q. Construct a solution $\mathcal{S}$ for μ-ν-Balance by letting $S_i := \{v \in V : \varphi(v) = i\}, \forall i \in [d]$. Clearly $\Phi_\circledcirc(\mathcal{S}) = l|E_{\varphi^*}(S^*)|$. Thus, using Corollary C.5: $\Phi_\circledcirc(\mathcal{S}^*_\circledcirc) \geq l \cdot \mathrm{MCD}^*_d \geq l \cdot p \cdot \mathrm{DKSH}^*_d$.

3. Let $S = \{v \in V_\square : v \in S_i$ for some $i \in [\mu - \nu + d]\} \subseteq V_\square = V$ be the set of rectangle-nodes where $\mathcal{S}$ propagates at least one campaign in $[\mu - \nu + d]$. Clearly, $|S| \leq k$. Let $q = |E(S)|$ be the number of edges in the sub-graph of G induced by S. Then, $\Phi_\circledcirc(\mathcal{S}) \leq \lambda l q$, since each edge in G can count for λl circle-nodes if the d corresponding rectangle-nodes propagate all campaigns in $[\mu - \nu + d]$. It follows that $|E(S)| \geq \Phi_\circledcirc(\mathcal{S})/(\lambda l)$.

□

C.3 Deferred Proofs for Section 6.3

Deferred Proofs for the Analysis of Algorithm GreedyTuple

The aim of this section is to prove the following Lemma.

Lemma 6.8. *Let $0 < \epsilon < 1$, $\delta \leq 1/2$, and $\ell \in [1, \nu - 1]$. With probability at least $1 - \delta$, after each iteration i of Algorithm 4, it either holds that*

$$\Phi^{\geq \ell}(\mathcal{S}^i) \geq \left(1 - \left(1 - \frac{1 - \frac{\epsilon}{2}}{\binom{B}{\nu - \ell}}\right)^i\right) \cdot \Phi^{\geq \ell}(\mathcal{S}^*_{\geq \ell}) \quad \textit{or} \quad \Phi^{\geq \ell}(\mathcal{S}^i) \geq \left(1 - \frac{1}{e}\right) \cdot \Phi^{\geq \ell}(\mathcal{S}^*_{\geq \ell}).$$

For this purpose, we will first prove the following lemma.

Lemma C.6. *Let $0 < \epsilon < 1$, $\delta \leq 1/2$, and $\ell \in [1, \nu - 1]$. With probability at least $1 - \delta$, after each iteration i of* GreedyTuple$(\epsilon, \delta, \ell, \mathcal{I}, \nu, B)$, *it either holds that*

$$\Phi^{\geq \ell}(\mathcal{S}^i) - \Phi^{\geq \ell}(\mathcal{S}^{i-1}) \geq \frac{1 - \frac{\epsilon}{2}}{\binom{B}{\nu - \ell}} \cdot (\Phi^{\geq \ell}(\mathcal{S}^*_{\geq \ell}) - \Phi^{\geq \ell}(\mathcal{S}^{i-1}))$$

$$\textit{or} \qquad \Phi^{\geq \ell}(\mathcal{S}^i) \geq \left(1 - \frac{1}{e}\right) \cdot \Phi^{\geq \ell}(\mathcal{S}^*_{\geq \ell}).$$

Proof. Algorithm GreedyTuple$(\epsilon, \delta, \ell, \mathcal{I}, \nu, B)$ calls algorithm approx at most t times. Let us call E_i the event that the i-th call to approx "succeeds", i.e., that the call results in $1 \pm \epsilon'$-approximation $\tilde{\Phi}^{\geq \ell}(\mathcal{T})$. That is, it holds that $(1 - \epsilon')\Phi^{\geq \ell}(\mathcal{T}) \leq \tilde{\Phi}^{\geq \ell}(\mathcal{T}) \leq (1 + \epsilon')\Phi^{\geq 1}(\mathcal{T})$. This event happens with probability at least $1 - \delta' = 1 - \delta/t$. Since there are at most t many evaluations, using the union bound, we obtain that the probability that all evaluations succeed is at least $1 - \delta$. Now the statement follows with Lemma 6.3.

It states that either $\Phi^{\geq\ell}(\mathcal{S}^i) \geq \left(1-\frac{1}{e}\right)\cdot\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell})$ or, for the element τ picked by the algorithm, it holds that $\Phi^{\geq\ell}(\mathcal{S}^{i-1}\cup\tau)-\Phi^{\geq\ell}(\mathcal{S}^{i-1}) \geq (1-\frac{\epsilon}{2})/\binom{B}{\nu-\ell}\cdot(\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell})-\Phi^{\geq\ell}(\mathcal{S}^{i-1}))$ using Lemma 6.1. □

We can now prove Lemma 6.8.

Proof of Lemma 6.8. We show the statement by induction. For $i=1$, we note that by Lemma C.6, we either have $\Phi^{\geq\ell}(\mathcal{S}^1)-\Phi^{\geq\ell}(\mathcal{S}^0) \geq (1-\frac{\epsilon}{2})/\binom{B}{\nu-\ell}\cdot(\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell})-\Phi^{\geq\ell}(\mathcal{S}^0))$ or $\Phi^{\geq\ell}(\mathcal{S}^i) \geq \left(1-\frac{1}{e}\right)\cdot\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell})$. In the latter case the statement holds, in the former case, we get $\Phi^{\geq\ell}(\mathcal{S}^1) \geq (1-\frac{\epsilon}{2})/\binom{B}{\nu-\ell}\cdot\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell})$ and thus the statement follows in both cases. For $i>1$, let us assume that the statement holds after iteration $i-1$. If $\Phi^{\geq\ell}(\mathcal{S}^{i-1}) \geq (1-{}^1\!/e)\cdot\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell})$, we have $\Phi^{\geq\ell}(\mathcal{S}^i) \geq (1-{}^1\!/e)\cdot\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell})$ by monotonicity. In the other case, we have that

$$\Phi^{\geq\ell}(\mathcal{S}^{i-1}) \geq \left(1-\left(1-\frac{1-\frac{\epsilon}{2}}{\binom{B}{\nu-\ell}}\right)^{i-1}\right)\cdot\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell}). \tag{C.3}$$

Applying Lemma C.6 yields that either $\Phi^{\geq\ell}(\mathcal{S}^i) \geq (1-{}^1\!/e)\cdot\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell})$, in which case the statement holds, or we obtain

$$\begin{aligned}\Phi^{\geq\ell}(\mathcal{S}^i) &= \Phi^{\geq\ell}(\mathcal{S}^{i-1}) + (\Phi^{\geq\ell}(\mathcal{S}^i)-\Phi^{\geq\ell}(\mathcal{S}^{i-1})) \\ &\geq \left(1-\frac{1-\frac{\epsilon}{2}}{\binom{B}{\nu-1}}\right)\Phi^{\geq\ell}(\mathcal{S}^{i-1}) + \frac{1-\frac{\epsilon}{2}}{\binom{B}{\nu-1}}\Phi^{\geq\ell}(\mathcal{S}^*_{\geq\ell}).\end{aligned}$$

Applying Eq. (C.3) yields the claim. □

Figure C.1 illustrates the scheme induced by an hyperedge $e=(u,v,w)$ when $d=3$ and $\mu=\nu=4$. In this case, $J=\binom{[\mu-\nu+d]}{d}$ is only composed of set [3] and S_3 is composed of 6 permutations. We use the standard tuple notation for permutations.

Deferred Proofs for the Analysis of Algorithm GreedyIter

In this section we prove lemmata 6.10 and 6.11 which are paramount in proving the approximation ratio of Algorithm GreedyIter.

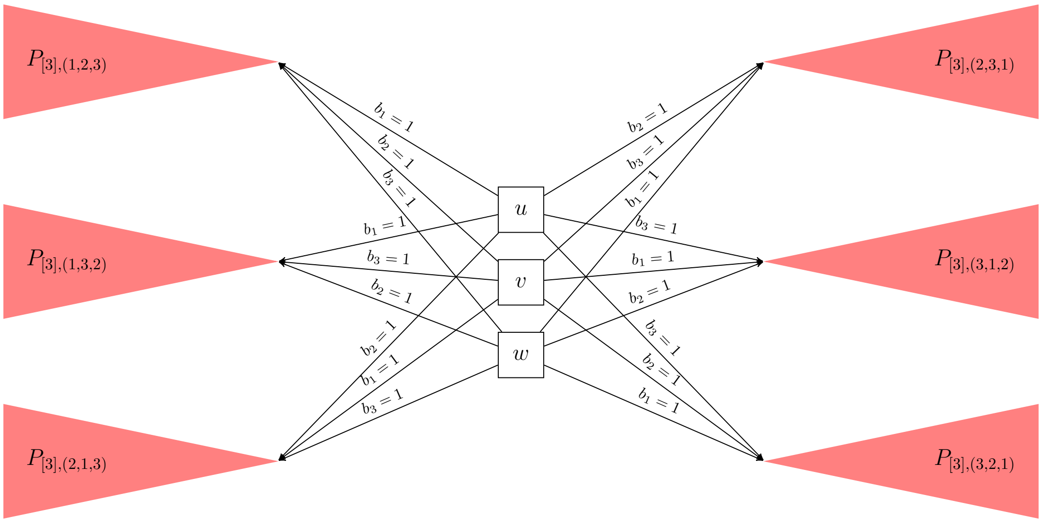


FIGURE C.1: For a set $\iota = \{i, j, k\} \in J$ of d campaigns and a permutation $\pi \in S_d$, let $P_{\iota,\pi}$ stand for the path in Figure 6.2 of nodes $e^1_{\iota,\pi}, \ldots, e^l_{\iota,\pi}$ connected by arcs $(e^t_{\iota,\pi}, e^{t+1}_{\iota,\pi})$ for $t = 1, \ldots, l-1$ with probabilities on these edges being one for $b_{\pi(i)}$, $b_{\pi(j)}$, and $b_{\pi(k)}$ and zero for all other indices. The figure illustrates the case $d = 3$ and $\mu = \nu = 4$ and the portion of the network that is generated in the transform τ of Section 6.2 for one hyper-edge $e = \{u, v, w\}$ and the only set $\iota = [3] \in J = \binom{3}{3}$. Probabilities that are not given are equal to zero.

Lemma 6.10. *Let $\epsilon > 0$ and assume that $B \geq 2(\nu - 1)/\epsilon$. If $\mathcal{S}^{[1]} \subseteq \hat{V}$ is the set of cardinality $\lfloor B/(\nu-1) \rfloor$ selected in the first iteration of* GREEDYITER$(\epsilon, \delta, \mathcal{I}, \nu, B)$, *then, with probability at least $1 - \delta/\nu$, it holds that*

$$\Phi^{\geq 1}_{2}(\mathcal{I}, \mathcal{S}^{[1]}) \geq \frac{1 - \frac{1}{e} - \epsilon}{\nu} \cdot \Phi^{\geq 1}(\mathcal{I}, \mathcal{S}^*_{\geq 1}),$$

where $\mathcal{S}^*_{\geq 1}$ is a set of cardinality B maximizing $\Phi^{\geq 1}(\mathcal{I}, \cdot)$.

Proof. Let $\mathcal{S}^*_{\lfloor B/(\nu-1) \rfloor}$ and $\mathcal{S}^*_B$ be sets of cardinality $\lfloor B/(\nu-1) \rfloor$ and B, respectively, maximizing $\Phi^{\geq 1}_2(\mathcal{I}, \cdot)$. Furthermore, let $\mathcal{T}$ be a subset of $\mathcal{S}^*_B$ of cardinality $\lfloor B/(\nu-1) \rfloor$ that maximizes $\Phi^{\geq 1}_2(\mathcal{I}, \cdot)$. Lemma 6.4 yields that for $\epsilon' = \epsilon/2$, with probability at least $1 - \delta/\nu$, we have that

$$\Phi^{\geq 1}_2(\mathcal{I}, \mathcal{S}^{[1]}) \geq \left(1 - \frac{1}{e} - \epsilon'\right) \cdot \Phi^{\geq 1}_2(\mathcal{I}, \mathcal{S}^*_{\lfloor B/(\nu-1) \rfloor}) \geq \left(1 - \frac{1}{e} - \epsilon'\right) \cdot \Phi^{\geq 1}_2(\mathcal{I}, \mathcal{T}). \tag{C.4}$$

Using the submodularity and monotonicity of $\Phi^{\geq 1}_2(\mathcal{I}, \cdot)$ and the maximum choice of $\mathcal{T}$ yields

$$\Phi^{\geq 1}_2(\mathcal{I}, \mathcal{S}^*_B) \leq \left\lceil \frac{B}{\lfloor B/(\nu-1) \rfloor} \right\rceil \cdot \Phi^{\geq 1}_2(\mathcal{I}, \mathcal{T}) \leq \frac{\nu}{1 - \epsilon'} \cdot \Phi^{\geq 1}_2(\mathcal{I}, \mathcal{T}), \tag{C.5}$$

as $B \geq (\nu - 1)/\epsilon'$ implies $B - \nu + 1 \geq (1 - \epsilon')B$ and thus

$$\left\lceil \frac{B}{\lfloor B/(\nu-1) \rfloor} \right\rceil \leq \left\lceil \frac{B(\nu-1)}{B - \nu + 1} \right\rceil \leq \left\lceil \frac{\nu - 1}{1 - \epsilon'} \right\rceil \leq \frac{\nu}{1 - \epsilon'}.$$

By combining the estimates from Eq. (C.4) and Eq. (C.5), we obtain

$$\Phi_2^{\geq 1}(\mathcal{I}, \mathcal{S}^{[1]}) \geq \frac{(1 - \frac{1}{e} - \epsilon')(1 - \epsilon')}{\nu} \cdot \Phi_2^{\geq 1}(\mathcal{I}, \mathcal{S}_B^*) \geq \frac{(1 - \frac{1}{e} - \epsilon)}{\nu} \cdot \Phi_\nu^{\geq 1}(\mathcal{I}, \mathcal{S}_{\geq 1}^*),$$

where the last step uses that $x(1 - \epsilon') \geq x - \epsilon'$ for any $x \leq 1$, the definition of $\epsilon' = \epsilon/2$, and the fact that $\mathcal{S}_B^*$ and $\mathcal{S}_{\geq 1}^*$ are both of size B and thus $\Phi_2^{\geq 1}(\mathcal{I}, \mathcal{S}_B^*) \geq \Phi_2^{\geq 1}(\mathcal{I}, \mathcal{S}_{\geq 1}^*) \geq \Phi_\nu^{\geq 1}(\mathcal{I}, \mathcal{S}_{\geq 1}^*)$. □

Lemma 6.11. *Let $\epsilon > 0$, $\ell \geq 2$ and assume that $B \geq 2(\nu - 1)/\epsilon$. If $\mathcal{S}^{[\ell]} \subseteq \hat{V}$ is the set of cardinality $\lfloor B/(\nu - 1) \rfloor$ selected in the ℓ'th iteration of* GreedyIter$(\epsilon, \delta, \mathcal{I}, \nu, B)$, *then, with probability at least $1 - \delta/\nu$, it holds that*

$$\Phi_{\ell+1}^{\geq \ell}(\mathcal{R}^{[\ell]}, \mathcal{S}^{[\ell]}) \geq \frac{(1 - \frac{1}{e} - \epsilon)B}{2(\ell + 1)(\nu - 1)|V|} \cdot \Phi_\ell^{\geq \ell - 1}(\mathcal{R}^{[\ell-1]}, \mathcal{S}^{[\ell-1]}).$$

Proof. We use the shorthand $\Phi^{[\ell]}(\cdot) := \Phi_{\ell+1}^{\geq \ell}(\mathcal{R}^{[\ell]}, \cdot)$ and similarly we use $\Phi^{[\ell-1]}(\mathcal{S}) := \Phi_\ell^{\geq \ell-1}(\mathcal{R}^{[\ell-1]}, \cdot)$. We define $U := V \times [\ell + 1]$ and partition it into sets of cardinality $\lfloor B/(\nu - 1) \rfloor$ plus a possible set of smaller size. The number of sets in the partition is $t := \lceil \frac{(\ell+1)|V|}{\lfloor B/(\nu-1) \rfloor} \rceil$. Denote these sets by $U_1, \ldots, U_t$. Now, let $\mathcal{T}$ be any set of cardinality $\lfloor B/(\nu - 1) \rfloor$ that maximizes $\Phi^{[\ell]}(\cdot)$ and assume for the purpose of contradiction that

$$\Phi^{[\ell]}(\mathcal{T}) - \Phi^{[\ell]}(\emptyset) < \frac{1}{t} \cdot (\Phi^{[\ell-1]}(\mathcal{S}^{[\ell-1]}) - \Phi^{[\ell]}(\emptyset)). \tag{C.6}$$

By definition of $\mathcal{T}$, we have $\Phi^{[\ell]}(\mathcal{T}) \geq \Phi^{[\ell]}(U_i)$ for $i \in [t]$. Hence, by submodularity we get

$$\begin{aligned} \Phi^{[\ell]}(U) - \Phi^{[\ell]}(\emptyset) &\leq \sum_{i=1}^{t} \left(\Phi^{[\ell]}(U_i) - \Phi^{[\ell]}(\emptyset)\right) \leq t\left(\Phi^{[\ell]}(\mathcal{T}) - \Phi^{[\ell]}(\emptyset)\right) \\ &< \Phi^{[\ell-1]}(\mathcal{S}^{[\ell-1]}) - \Phi^{[\ell]}(\emptyset). \end{aligned}$$

Since the maximum possible number of nodes, say N are guaranteed to be reached by $\ell + 1$ campaigns from sets U, we have however that $\Phi^{[\ell]}(V^{\ell+1}) = N$. On the other hand we have $\Phi^{[\ell-1]}(\mathcal{S}^{[\ell-1]}) \leq N$, which leads to a contradiction. From Lemma 6.4 we know that, with probability at least $1 - \delta/\nu$, it holds that $\Phi^{[\ell]}(\mathcal{S}^{[\ell]}) \geq (1 - 1/e - \epsilon') \cdot \Phi^{[\ell]}(\mathcal{T})$ with $\epsilon' = \epsilon/2$. Thus, together with the converse of Eq. (C.6), we get

$$\begin{aligned}\Phi^{[\ell]}(\mathcal{S}^{[\ell]}) &\geq \frac{1 - \frac{1}{e} - \epsilon'}{t} \cdot (\Phi^{[\ell-1]}(\mathcal{S}^{[\ell-1]}) - \Phi^{[\ell]}(\emptyset)) + \Phi^{[\ell]}(\emptyset) \\ &\geq \frac{1 - \frac{1}{e} - \epsilon'}{t} \cdot \Phi^{[\ell-1]}(\mathcal{S}^{[\ell-1]}).\end{aligned}$$

It remains to observe that $B \geq (\nu - 1)/\epsilon'$ implies $B - \nu + 1 \geq (1 - \epsilon')B$ and thus

$$t = \left\lceil \frac{(\ell+1)|V|}{\lfloor B/(\nu-1) \rfloor} \right\rceil \leq \left\lceil \frac{(\ell+1)(\nu-1)|V|}{B - \nu + 1} \right\rceil \leq \frac{2(\ell+1)(\nu-1)|V|}{(1-\epsilon')B}$$

where the last inequality follows since $B \leq \nu \cdot |V|$ and $\nu \geq 2$ yield that the argument of the ceil-function is at least 1, and thus the error due to rounding is upper bounded by a factor of 2. The choice of $\epsilon' = \epsilon/2$ leads the result. □

C.4 Deferred Proofs for Section 6.4

Lemma 6.13. *The following statements hold:*

1. *If $\mathcal{T}^B \subseteq V \times \{0\}$ is a solution of size B maximizing Ψ and $\mathcal{S}^B \subseteq \hat{V}$ is a solution of size B maximizing $\Phi^{\geq 1}$, then $\Psi(\mathcal{T}^B) \geq \Phi^{\geq 1}(\mathcal{S}^B)$.*
2. *Let $\epsilon > 0$ and $B \geq \nu/\epsilon$. If $\mathcal{T}^B \subseteq V \times \{0\}$ is a solution of size B maximizing Ψ and $\mathcal{T}^{\lfloor B/\nu \rfloor} \subseteq V \times \{0\}$ is a solution of size $\lfloor B/\nu \rfloor$ maximizing Ψ, then $\Psi(\mathcal{T}^{\lfloor B/\nu \rfloor}) \geq \frac{1-\epsilon}{\nu+1} \cdot \Psi(\mathcal{T}^B)$.*
3. *Let $\mathcal{T} \subseteq V \times \{0\}$ be of size $\lfloor B/\nu \rfloor$. Then $\mathcal{S}' := \{(v,j) | (v,0) \in \mathcal{T}, j \in [\nu]\} \subseteq \hat{V}$ is a set of size at most B such that $\Phi^{\geq 1}(\mathcal{S}') = \Psi(\mathcal{T})$.*

Proof. 1. Define $\mathcal{T} := \{(v,0) | (v,i) \in \mathcal{S}^B\}$ and observe that $|\mathcal{T}| \leq B$. For a given outcome $\mathcal{X}$ a node that contributes to $\Phi^{\geq 1}(\mathcal{S}^B)$ is either reached by at least ν campaigns in $\mathcal{I}$ or has to be reached by a node in $\mathcal{S}^B$. In this case, for the same $\mathcal{X}$ this node will also contribute to $\Psi(\mathcal{T})$. Hence, we have $\Psi(\mathcal{T}) \geq \Phi^{\geq 1}(\mathcal{S}^B)$. The optimality of $\mathcal{T}^B$ concludes the proof.

2. First observe that $\lceil \frac{B}{\lfloor B/\nu \rfloor} \rceil < \frac{B}{B/\nu - 1} + 1 \leq \frac{\nu+1}{1-\epsilon}$ by the assumption on B. Now, let $\mathcal{T}$ be a subset of $\mathcal{T}^B$ of size $\lfloor B/\nu \rfloor$ maximizing Ψ. By submodularity of Ψ,

we have $\Psi(\mathcal{T}^k) \leq \lceil \frac{B}{\lfloor B/\nu \rfloor} \rceil \Psi(\mathcal{T}) \leq \frac{\nu+1}{1-\epsilon}\Psi(\mathcal{T})$. Using the optimality of $\mathcal{T}^{\lfloor B/\nu \rfloor}$ concludes the proof.

3. Since the cascade processes are completely correlated, given an outcome $\mathcal{X}$, assume that a node contributes to $\Psi(\mathcal{T})$, then either it is reached by ν campaigns from $\mathcal{I}$ or it is reached by $\mathcal{T}$. In the former case, the same node also contributes to $\Phi^{\geq 1}(\mathcal{S}')$ as it is reached by ν campaigns from $\mathcal{I}$. In the later case, it will be reached by all campaigns in $[\nu]$ by $\mathcal{S}'$ and will therefore also contribute to $\Phi^{\geq 1}(\mathcal{S}')$.

□

Appendix D

Supplemental Material for Chapter 7

Theorem 7.1. *Given a graph $G = (V, E)$, $\sigma(A, (V, E \cup S))$ is a monotone submodular function of $S \subseteq \bar{E} \setminus E$.*

Proof. We first prove that $\sigma(A, \cdot)$ is a monotonically increasing function, formally we show that $\sigma(A, S \cup \{e\}) \geq \sigma(A, S)$ for any $S \subseteq \bar{E} \setminus E$ and $e = (u, v) \in \bar{E} \setminus E$.

We decompose $\sigma(A, S \cup \{e\})$ as the sum over all the live-edge graphs in which: an edge in E has been selected; v has no incoming edges; and edge e has been selected. Formally [1],

$$\sigma(A, S \cup \{e\}) = \sum_{\substack{G' \in \mathcal{G}(S \cup e) \\ s.t. \exists (z,v) \in E', z \neq u}} \mathbf{P}_{S \cup e}\left(G'\right) |R_A(G')| + \sum_{\substack{G' \in \mathcal{G}(S \cup e) \\ s.t. \nexists (z,v) \in E'}} \mathbf{P}_{S \cup e}\left(G'\right) |R_A(G')| + \sum_{\substack{G' \in \mathcal{G}(S \cup e) \\ s.t. e \in E'}} \mathbf{P}_{S \cup e}\left(G'\right) |R_A(G')|.$$

[1] E' and E'' denote the edges sets of graphs G' and G'', resp.

Similarly we can decompose $\sigma(A,S)$ as follows:

$$\sigma(A,S) = \sum_{\substack{G''\in\mathcal{G}(S)\\ s.t.\exists(z,v)\in E''}} \mathbf{P}_S\left(G''\right)|R_A(G'')| + \sum_{\substack{G''\in\mathcal{G}(S)\\ s.t.\nexists(z,v)\in E''}} \mathbf{P}_S\left(G''\right)|R_A(G'')|.$$

Using observation 1 and 2 we can consider pair of live-edge graphs, one from $\mathcal{G}(S\cup e)$ and one from $\mathcal{G}(S)$, and notice that the two graphs are equivalent in the case in which an edge different from e is selected or node v has no incoming edges. Although, in the latter case the probabilities to sample the live-edge graph are not equal. Thus, we have that

$$\begin{aligned}\sigma(A,S\cup\{e\})-\sigma(A,S) = &\sum_{\substack{G'\in\mathcal{G}(S\cup e)\\ s.t.\nexists(z,v)\in E'}} \left[p(V\setminus\{v\},G',S)\left(1-\sum_{z\in N_v} b_{zv} - b_{uv}\right)|R_A(G')|\right]\\ &+\sum_{\substack{G'\in\mathcal{G}(S\cup e)\\ s.t.e\in E'}} p(V\setminus\{v\},G',S)\cdot b_e\cdot|R_A(G')|\\ &-\sum_{\substack{G''\in\mathcal{G}(S)\\ s.t.\nexists(z,v)\in E''}} p(V\setminus\{v\},G'',S)\cdot\left(1-\sum_{z\in N_v} b_{zv}\right)|R_A(G'')|.\end{aligned}$$

That means $\sigma(A,S\cup\{e\})-\sigma(A,S)$ is equal to

$$\sum_{\substack{G'\in\mathcal{G}(S)\\ s.t.\nexists(z,v)\in E'}} p(V\setminus\{v\},G',S)b_e\left(|R_A(G''')|-|R_A(G')|\right),$$

where G''' is the graph G' augmented with the edge e and the number of live-edge graphs such that the edge e has been selected is the same as the number of live-edge graphs for which no incoming edge is selected for v. Note that this value is greater or equal than zero because $|R_A(G'')|\geq|R_A(G')|$.

In order to prove that the function is submodular, we show that for each pair of sets S,T such that $S\subseteq T\subset\bar{E}\setminus E$ and for each $e=(a,v)\in\bar{E}\setminus(T\cup E)$,

$$\sigma(A,S\cup\{e\})-\sigma(A,S)\geq\sigma(A,T\cup\{e\})-\sigma(A,T).$$

Let V' be the set of nodes that have an incoming edge from the set $T \setminus S$, namely, $V' = \{v : (w, v) \in T \setminus S\}$. Observe that for any live-edge graph $G' \in \mathcal{G}(S)$ for which the nodes in V' have no incoming edges there exists $G'_1, \dots, G'_\ell \in \mathcal{G}(T)$ such that $G' \subseteq G'_i$ for any $i = 1, \dots, \ell$ and $R_A(G') \subseteq R_A(G'_i)$, where $\ell = 2^{|T \setminus S|}$. While for all graphs $G' \in \mathcal{G}(S)$ that have at least an edge incoming a node in V', there exists a corresponding live-edge graph $G'' \in \mathcal{G}(T)$ that is sampled with the same probability as G'. In the former case, instead, we have that the probability for each G'_i, $i = 1, \dots, \ell$, is equal to the probability of the corresponding live-edge G' in $\mathcal{G}(S)$ in which no incoming edge is selected for the nodes in V'. Formally we have that

$$\mathbf{P}_S\left(G'\right) = p(V \setminus V', G', S) \prod_{v \in V'} (1 - \sum_{z \in N_v} b_{zv})$$

and $\mathbf{P}_T\left(G'_i\right) = p(V \setminus V', G', S) \cdot p(V', G', T \setminus S)$, where

$$p(V', G', T \setminus S) = \prod_{\substack{z \in V s.t. \\ (u,z) \in E' \cap (T \setminus S)}} b_{uz} \prod_{\substack{z \in V s.t. \\ \nexists (u,z) \in E' \cap (T \setminus S)}} \left(1 - \sum_{w : (w,z) \in E \cup T} b_{wz}\right).$$

Then,

$$\sum_{i=1}^{\ell} \mathbf{P}_T\left(G'_i\right) = p(V \setminus V', G', S) \prod_{v \in V'} (1 - \sum_{z \in N_v} b_{zv}) = \mathbf{P}_S\left(G'\right).$$

Finally we can write the difference in the increment when adding the edge $e = (a, v)$ in the set T as follow:

$$\begin{aligned}
\sigma(A, T \cup \{e\}) - \sigma(A, T) &= \sum_{\substack{G' \in \mathcal{G}(T) \\ s.t. \nexists (z,v) \in E'}} p(V \setminus \{v\}, G', T)\ b_e\ \left(|R_A(G'')| - |R_A(G')|\right) \\
&= \sum_{\substack{G' \in \mathcal{G}(S) \\ s.t. \nexists (z,v) \in E'}} \left(\sum_{i=1}^{\ell} \mathbf{P}_T\left(G'_i\right)\ b_e\ \left(|R_A(G'')| - |R_A(G')|\right)\right) \\
&\leq \sum_{\substack{G' \in \mathcal{G}(S) \\ s.t. \nexists (z,v) \in E'}} \left(p(V \setminus V', G', S) \prod_{v \in V'} (1 - \sum_{z \in N_v} b_{zv})\right)\ b_e \left(|R_A(G'')| - |R_A(G')|\right) \\
&\leq \sum_{\substack{G' \in \mathcal{G}(S) \\ s.t. \nexists (z,v) \in E'}} p(V \setminus \{v\}, G', S)\ b_e\ \left(|R_A(G'')| - |R_A(G')|\right) = \sigma(A, S \cup \{e\}) - \sigma(A, S)
\end{aligned}$$

where G'' is the graph G' augmented with the edge e. □

D.1 Diffusion process approximation

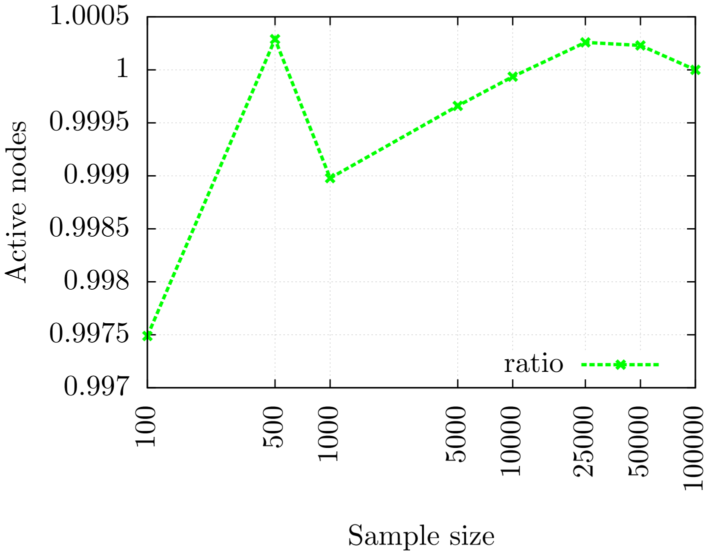


FIGURE D.1: Approximation of $\sigma(A)$ using repeated sampling.

In Figure D.1 we show that the quality of the approximation for $\sigma(A)$ after 500 simulations of the diffusion process (both ICM and LTM) is comparable to that after 100000 iterations. We report the results for the *Twitter* network ($n = 465017$ nodes and $m = 834797$ edges), the results for the other networks are similar and therefore omitted. We run the diffusion process selecting a random set of seed nodes A such that $|A| = 1\% \cdot n$. We plot the ratio between the number of active nodes using $100, 500, 1000, 5000, 10000, 25000, 50000, 100000$ samples and the number of active nodes using 100000 samples, we notice that the ratio is almost 1. Therefore, we choose 500 as the number of samples to use for our experiments.

D.2 Adding more than one edge incident to the same node

By means of experiments, we can show that even if we do not allow addition of edges from two, or more, different seeds $a_1, \ldots, a_i$ to the same node $v_i \in V \setminus A$, we do not affect much the approximation guarantee.

We run GREEDY1 on five randomly generated set of seeds choosing $|A| = 1\% \cdot |V|$, adding $|S| = 2 \times |A|$ edges and assigning probabilities according to the weighted model. In particular, we plot the results of the relative error for the network Wiki-Vote in Figure D.2. In this graph, obtained from SNAP [97], nodes in the network represent Wikipedia users and a directed edge from node i to node j represents that user i voted on user j. It is easy to see that the two functions representing the number of influenced nodes coincide on most of the points.

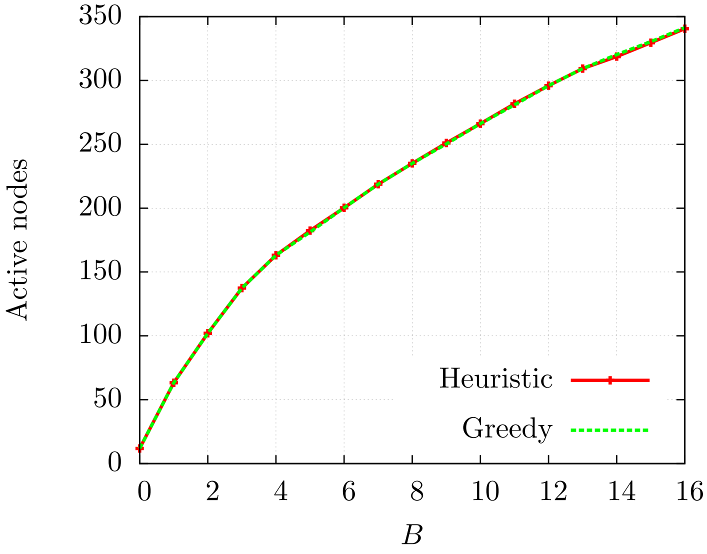


FIGURE D.2: Approximation error for Wiki-Vote network.

The results for many other networks, taken from the ArnetMiner [96] repository, are similar and reported in Table D.1: the first three columns report the name of the network, the number of nodes and the number of edges. The last column of the table reports the maximum error e_{max} among all the edge insertions and it is computed as

$$e_{max} = \left| \frac{\sigma(A, S_{all}) - \sigma(A, S)}{\sigma(A, S_{all})} \right|$$

where $\sigma(A, S_{all})$ is the expected number of active nodes computed using GREEDY1 and allowing multiple edges to the same node while $\sigma(A, S)$ is that obtained adding only one outgoing edge per seed to the same node v_i.

We can conclude that it is very unlikely that two edges towards the same node in $V \setminus A$ are added to the solution.

Name	$\|V\|$	$\|E\|$	e_{max}
Software Engineering (SE)	3141	14787	0.045
Theoretical CS (TCS)	4172	14272	0.035
High-Performance Comp. (HPC)	4869	35036	0.033
Wiki-Vote (Wiki)	7115	103689	0.033
Computer Graphic (CGM)	8336	41925	0.045
Computer Networks (CN)	9420	53003	0.035

Table D.1: Real-world networks and relative approximation error.

D.3 Experiments on random graphs

We evaluate the performance of the algorithm on four types of randomly generated directed networks which exhibit many of the structural features of complex networks, namely directed Preferential Attachment (in short, PA) [99], Erdős-Rényi (ER) [109], Copying (COPY) [110], Compressible Web (COMP) [111] and Forest Fire (FF) [112]. For each combination $(|V|, |E|)$, we generated five random directed graphs.

The size of the graphs is reported in Table D.2. We choose 0.1% of the nodes in V as seeds and we add up to $B = 2 \cdot |A|$ edges. The seed nodes are chosen uniformly at random.

The experimental results are reported in Tables D.3, D.4 (activated nodes) and D.5, D.6 (comparison with baselines).

Name	$\|V\|$	$\|E\|$
PA5	5000	6500
PA10	10000	13000
PA15	15000	20000
FF5	5000	10000
FF10	10000	20000
FF15	15000	30000
COPY5	5000	5000, 10000, 25000
COPY10	10000	20000, 50000, 100000
COPY15	15000	45000, 100000
COMP5	5000	5000, 10000, 25000
COMP10	10000	20000,50000, 100000
COMP15	15000	45000, 100000
ER5	5000	10000, 25000, 50000
ER10	10000	40000, 100000, 200000
ER15	150000	90000, 225000, 450000

Table D.2: Random networks.

G	$\sigma(A,\emptyset)$	$\sigma(A,\emptyset)\%$	GREEDY1				GREEDY2			
			$\sigma(A,S)$	$\sigma(A,S)\%$	$I\%$	time (sec.)	$\sigma(A,S)$	$\sigma(A,S)\%$	$I\%$	time (sec.)
PA5	7.30	0.17	333.37	7.67	4467.18	10.04	332.84	7.66	4448.94	0.03
PA10	15.05	0.17	688.57	7.84	4474.88	38.02	685.10	7.80	4474.17	0.07
PA15	20.51	0.16	1094.54	8.35	5237.70	86.93	1092.27	8.33	5230.44	0.12
FF5	10.07	0.20	137.94	2.76	1269.51	14.04	136.06	2.72	1246.09	0.07
FF10	19.08	0.19	276.40	2.76	1348.45	55.82	273.96	2.74	1343.34	0.15
FF15	28.34	0.19	432.76	2.89	1426.96	115.94	429.65	2.86	1420.92	0.25
COPY5-5	7.39	0.15	48.06	0.96	549.97	11.21	47.85	0.96	547.33	0.14
COPY5-10	9.12	0.18	81.71	1.63	795.56	13.61	80.43	1.61	783.16	0.09
COPY5-25	11.97	0.24	157.78	3.16	1218.38	12.45	154.85	3.10	1208.11	0.06
COPY10-20	17.99	0.18	167.47	1.67	830.95	51.26	164.63	1.65	809.46	0.21
COPY10-50	26.51	0.27	335.79	3.36	1166.53	48.53	329.07	3.29	1150.25	0.13
COPY10-100	31.98	0.32	559.43	5.59	1649.14	39.45	547.73	5.48	1608.47	0.11
COPY15-45	30.41	0.20	329.38	2.20	983.29	113.47	323.41	2.16	962.07	0.27
COPY15-100	40.40	0.27	630.23	4.20	1459.87	97.38	619.77	4.13	1446.42	0.20
COPY15-225	52.00	0.35	1128.13	7.52	2069.52	73.86	1102.94	7.35	2006.04	0.18
COMP5-5	9.39	0.19	62.43	1.25	564.61	15.14	61.92	1.24	562.24	0.11
COMP5-10	8.74	0.17	58.56	1.17	569.85	14.18	58.03	1.16	563.22	0.12
COMP5-25	7.97	0.16	55.81	1.12	600.02	12.74	54.46	1.09	583.92	0.12
COMP10-20	16.03	0.16	115.16	1.15	618.20	53.19	114.19	1.14	613.41	0.29
COMP10-50	16.85	0.17	112.70	1.13	568.72	48.59	109.95	1.10	554.42	0.29
COMP10-100	16.11	0.16	112.93	1.13	600.94	48.93	109.94	1.10	582.67	0.30
COMP15-45	24.79	0.17	170.61	1.14	588.17	114.11	167.55	1.12	573.56	0.46
COMP15-100	24.32	0.16	170.58	1.14	601.45	111.12	167.26	1.12	588.28	0.48
COMP15-225	24.82	0.17	171.82	1.15	592.36	107.07	169.09	1.13	582.21	0.50

TABLE D.3: Results for random networks (ICM).

G	$\sigma(A,\emptyset)$	$\sigma(A,\emptyset)\%$	GREEDY1				GREEDY2			
			$\sigma(A,S)$	$\sigma(A,S)\%$	$I\%$	time (sec.)	$\sigma(A,S)$	$\sigma(A,S)\%$	$I\%$	time (sec.)
PA5	6.72	0.15	132.20	3.04	1866.23	3.33	141.42	3.25	1885.31	0.08
PA10	13.53	0.15	272.78	3.11	1916.24	6.89	297.09	3.38	1980.65	0.15
PA15	18.68	0.14	432.16	3.30	2213.32	10.31	470.22	3.59	2290.91	0.24
FF5	8.33	0.17	49.89	1.00	498.95	3.76	61.12	1.22	575.13	0.11
FF10	16.32	0.16	106.05	1.06	549.64	7.56	133.37	1.33	659.49	0.27
FF15	24.04	0.16	161.44	1.08	571.65	11.48	203.40	1.36	690.55	0.43
COPY5-5	6.70	0.13	22.01	0.44	228.31	4.78	22.69	0.45	224.33	0.24
COPY5-10	7.90	0.16	30.98	0.62	292.09	4.06	34.53	0.69	310.20	0.17
COPY5-25	9.53	0.19	49.43	0.99	418.47	3.22	61.04	1.22	498.48	0.18
COPY10-20	15.25	0.15	62.79	0.63	311.83	8.31	69.34	0.69	323.29	0.38
COPY10-50	19.85	0.20	103.34	1.03	420.53	7.06	125.60	1.26	465.98	0.36
COPY10-100	22.53	0.23	148.03	1.48	557.06	5.98	191.86	1.92	662.01	0.55
COPY15-45	25.08	0.17	114.71	0.76	357.41	11.88	128.55	0.86	376.51	0.60
COPY15-100	29.85	0.20	179.87	1.20	502.63	10.57	228.39	1.52	566.52	0.65
COPY15-225	34.89	0.23	276.52	1.84	692.65	9.22	389.49	2.60	869.25	1.21
COMP5-5	7.84	0.16	25.44	0.51	224.68	4.03	27.36	0.55	223.39	0.22
COMP5-10	7.55	0.15	27.26	0.55	261.32	4.08	29.18	0.58	263.29	0.21
COMP5-25	7.14	0.14	27.54	0.55	285.77	4.05	29.57	0.59	290.93	1.13
COMP10-20	14.26	0.14	52.24	0.52	266.39	8.28	54.73	0.55	268.17	0.45
COMP10-50	15.03	0.15	55.33	0.55	268.13	8.73	57.95	0.58	264.55	3.25
COMP10-100	14.60	0.15	56.53	0.57	287.18	8.88	59.95	0.60	291.71	20.05
COMP15-45	22.11	0.15	81.23	0.54	267.34	13.90	84.86	0.57	260.04	1.26
COMP15-100	21.96	0.15	85.23	0.57	288.15	13.41	90.21	0.60	290.18	16.70
COMP15-225	22.49	0.15	88.29	0.59	292.64	14.25	92.82	0.62	294.14	108.79

TABLE D.4: Results for random networks (LTM).

G	$B = 0$		GREEDY2	AA	PA	J	D	TopK	Prob	KKT
	$\sigma(A, \emptyset)$	$\sigma(A, \emptyset)\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$
PA5	7.30	0.17	4467.18	1996.92	2691.24	389.35	2877.88	3957.10	216.01	3900.63
PA10	15.05	0.17	4474.88	1785.58	3114.64	150.66	3183.03	3807.15	178.39	4030.14
PA15	20.51	0.16	5237.70	2587.99	3247.53	196.43	2049.33	4284.63	202.95	4079.94
FF5	10.07	0.20	1269.51	58.74	263.83	59.62	325.81	401.58	206.53	5079.94
FF10	19.08	0.19	1348.45	71.36	278.57	67.69	362.02	470.00	221.63	981.44
FF15	28.34	0.19	1426.96	63.68	322.10	68.29	425.73	515.21	225.89	929.24
COPY5-5	7.39	0.15	549.97	83.43	9.86	83.43	191.46	282.70	215.58	414.55
COPY5-10	9.12	0.18	795.56	43.29	3.70	50.57	168.26	477.72	206.25	537.71
COPY5-25	11.97	0.24	1218.38	52.57	0.72	51.10	137.23	785.68	246.65	990.36
COPY10-20	17.99	0.18	830.95	51.69	3.30	53.28	165.90	408.71	211.38	737.68
COPY10-50	26.51	0.27	1166.53	51.84	0.77	51.45	127.60	709.52	222.21	1038.17
COPY10-100	31.98	0.32	1649.14	55.77	0.29	51.15	76.79	788.38	317.94	1415.66
COPY15-45	30.41	0.20	983.29	53.59	1.84	52.69	172.67	567.75	225.81	880.44
COPY15-100	40.40	0.27	1459.87	45.39	0.54	41.39	109.73	995.94	297.03	1275.84
COPY15-225	52.00	0.35	1128.13	48.13	0.21	40.90	50.68	929.98	383.81	1073.52
COMP5-5	9.39	0.19	564.61	68.48	17	68.48	202.41	276.67	211.75	503.82
COMP5-10	8.74	0.17	569.85	60.91	2.83	60.91	186.71	268.85	217.59	530.11
COMP5-25	7.97	0.16	600.02	60.40	1.55	60.40	178.28	297.03	224.21	592.42
COMP10-20	16.03	0.16	618.20	61.62	3.86	61.62	195.86	254.23	221.14	589.87
COMP10-50	16.85	0.17	568.72	68.12	0.91	68.12	204.75	268.39	236.17	555.31
COMP10-100	16.11	0.16	600.94	68.44	1.29	68.44	185.98	272.92	227.66	568.24
COMP15-45	24.79	0.17	588.17	61.15	1.51	61.15	179.62	233.62	233.02	557.50
COMP15-100	24.32	0.16	601.45	57.70	1.35	57.70	178.75	249.61	228.20	597.35
COMP15-225	24.82	0.17	592.36	59.48	0.71	59.48	176.23	244.07	225.91	572.52

TABLE D.5: Baseline results for random networks (ICM).

G	$B = 0$		GREEDY2	AA	PA	J	D	TopK	Prob	KKT
	$\sigma(A, \emptyset)$	$\sigma(A, \emptyset)\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$	$I\%$
PA5	6.71	0.15	1885.31	987.99	1640.56	211.4	1776.95	1773.83	102.76	1795.10
PA10	13.57	0.15	1980.65	848.85	1707.63	79.77	1865.31	1687.18	91.02	1875.23
PA15	18.76	0.14	2290.91	1137.50	1944.29	102.85	2168.81	1847.77	101.28	2180.47
FF5	8.36	0.17	575.13	38.91	111.62	35.69	124.69	211.59	107.25	368.06
FF10	16.36	0.16	659.49	39.09	117.04	38.70	150.44	232.07	107.51	446.36
FF15	24.11	0.16	690.55	39.79	167.42	38.72	186.36	255.20	118.61	460.99
COPY5-5	6.70	0.13	224.33	43.91	18.98	43.91	91.27	134.12	107.79	221.25
COPY5-10	7.92	0.16	310.20	28.95	16.49	30.64	68.60	194.35	99.67	257.13
COPY5-25	9.52	0.19	498.48	30.49	12.40	30.32	58.28	291.20	118.23	384.06
COPY10-20	15.23	0.15	323.29	30.19	17.84	31.34	76.60	172.63	109.94	291.46
COPY10-50	19.91	0.20	465.98	30.69	13.60	30.78	56.53	275.64	117.36	386.69
COPY10-100	22.57	0.23	662.01	29.47	11.28	27.88	35.24	398.60	157.88	509.22
COPY15-45	25.09	0.17	376.51	32.16	13.76	31.83	74.09	218.13	110.36	334.18
COPY15-100	29.86	0.20	566.52	27.55	13.11	26.29	47.91	365.96	138.18	454.24
COPY15-225	34.95	0.23	869.25	26.81	10.56	24.24	24.86	464.83	186.77	597.01
COMP5-5	7.83	0.16	223.39	39.08	16.17	39.08	91.19	122.93	103.05	200.92
COMP5-10	7.52	0.15	263.29	36.61	15.43	36.61	83.17	124.97	107.51	249.83
COMP5-25	7.12	0.14	290.93	38.67	17.78	38.67	77.24	147.48	111.48	291.39
COMP10-20	14.25	0.14	268.17	36.86	19.53	36.86	86.20	115.92	112.93	268.57
COMP10-50	15.01	0.15	264.55	41.77	16.55	41.77	94.58	130.59	115.76	272.88
COMP10-100	14.61	0.15	291.71	39.22	16.26	39.22	83.68	142.97	117.31	291.30
COMP15-45	22.13	0.15	260.04	37.85	18.84	37.85	82.23	116.88	111.13	271.28
COMP15-100	21.94	0.15	290.18	35.71	16.97	35.71	81.55	130.94	111.85	293.32
COMP15-225	22.50	0.15	294.14	35.98	16.47	35.98	78.75	132.40	109.85	294.51

TABLE D.6: Baseline results for random networks (LTM).