



HAL
open science

Election Control Through Social Influence with Unknown Preferences

Mohammad Abouei Mehrizi, Federico Corò, Emilio Cruciani, Gianlorenzo
d'Angelo

► **To cite this version:**

Mohammad Abouei Mehrizi, Federico Corò, Emilio Cruciani, Gianlorenzo d'Angelo. Election Control Through Social Influence with Unknown Preferences. COCOON 2020 - 26th International Conference on Computing and Combinatorics, Aug 2020, Atlanta / Online, United States. pp.397-410, 10.1007/978-3-030-58150-3_32 . hal-03007175

HAL Id: hal-03007175

<https://hal.science/hal-03007175>

Submitted on 11 Dec 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Election Control through Social Influence with Unknown Preferences

Mohammad Abouei Mehrizi

Gran Sasso Science Institute

L'Aquila, Italy

mohammad.aboueimehrizi@gssi.it

Federico Corò

Sapienza University of Rome

Rome, Italy

federico.coro@uniroma1.it

Emilio Cruciani

Inria, I3S Lab, UCA, CNRS

Sophia Antipolis, France

emilio.cruciani@inria.fr

Gianlorenzo D'Angelo

Gran Sasso Science Institute

L'Aquila, Italy

gianlorenzo.dangelo@gssi.it

Abstract

The election control problem through social influence asks to find a set of nodes in a social network of voters to be the starters of a political campaign aiming at supporting a given target candidate. Voters reached by the campaign change their opinions on the candidates. The goal is to shape the diffusion of the campaign in such a way that the chances of victory of the target candidate are maximized. Previous work shows that the problem can be approximated within a constant factor in several models of information diffusion and voting systems, assuming that the controller, i.e., the external agent that starts the campaign, has full knowledge of the preferences of voters. However this information is not always available since some voters might not reveal it. Herein we relax this assumption by considering that each voter is associated with a probability distribution over the candidates. We propose two models in which, when an electoral campaign reaches a voter, this latter 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 through social influence on the new models: In the first model, under the Gap-ETH, 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, which is a slight relaxation of the first one, the problem admits a constant factor approximation algorithm.

1 Introduction

Social media play a fundamental role in everyone’s life providing information, entertainment, and learning. Many social media users prefer to access social network platforms such as Facebook or Twitter before news websites as they provide faster means for information diffusion [MS18]. As a consequence, online social networks are also exploited as a tool to alter users’ opinions. The extent to which the opinions of an individual are conditioned by social interactions is called social influence. It has been observed that social influence starting from a small set of individuals may generate a cascade effect that allows to reach a large part of the network. Recently, this capability has been used to affect the outcome of political elections. There exists evidence of political intervention which shows the effect of social media manipulation on the elections outcome, e.g., by spreading fake news [PCR18]. A real-life example is in the 2016 US election where a study showed that on average 92% of people remembered pro-Trump fake news and 23% of them remembered pro-Clinton fake news [AG17]. Several other cases have been studied [BFJ⁺12, Fer17, Kre16, SBLS18].

There exists a wide literature about manipulation of voting systems; we point the reader to a recent survey [FR16]. Despite that, only few studies focus on the problem of controlling the outcome of political elections through the spread of information in social networks. The *election control* problem [WV18] consists in selecting a set of nodes of a network to be the starters of a diffusion with the aim of maximizing the chances for a target candidate to win an election. In particular, in the *constructive* election control problem, the goal is to maximize the *Margin of Victory* (MoV) of the target candidate on its most critical opponent, i.e., the difference of votes (or score, depending on the voting system) between the two candidates after the effect of social influence. A variation of the problem, known as *destructive* election control, aims at making a target candidate lose. Both problems have been originally analyzed under the Independent Cascade Model (ICM) [KKT15], and considering *plurality voting*; approximation and hardness-of-approximation results are provided [WV18]. Corò et al. [CCDP19a, CCDP19b] analyzed the problem in arbitrary scoring rules voting systems under the Linear Threshold Model (LTM) [KKT15], providing constant factor approximation algorithms. It has been later shown that it is *NP*-hard to find any constant factor approximation in the multi-winner scenario [AD20].

Faliszewski et al. [FGKT18] examine bribery in an opinion diffusion process with voter clusters: each node is a cluster of voters, represented as a weight, with a specific list of candidates; there is an edge between two nodes if they differ by the ordering of a single pair of adjacent candidates. The authors show that making a specific candidate win in their model is *NP*-hard and fixed-parameter tractable with respect to the number of candidates. Bredereck et al. [BE17] studied the problem of manipulating diffusion on social networks, though not specifically in the context of elections. They show that identifying successful manipulation via bribing, adding/deleting edges, or controlling the order of asynchronous updates are all computationally hard problems. A similar approach is taken by Apt et al. [AM14], where the authors introduce a threshold model for social networks in order to characterize the role of social influence in the global adoption of a commercial product.

Contribution. In all previous works it is assumed that the controller knows the preference list of each voter. However, this assumption is not always satisfied in realistic scenarios as voters may not reveal their preferences to the controller. Herein, in Section 2, we introduce two new models, *Probabilistic Linear Threshold Ranking* (PLTR) and *Relaxed-PLTR* (R-PLTR), that encompass scenarios where the preference lists of the voters are not fully revealed. Specifically, we use an uncertain model in which the controller only knows, for each voter, a probability

distribution over the candidates. In fact, in applied scenarios, the probability distribution could be inferred by analyzing previous social activity of the voters, e.g., re-tweets or likes of politically oriented posts. We envision that some given focused news about a target candidate spread through the network as a message. We model such a diffusion via the LTM [KKT15]. The message will have an impact on the opinions of voters who received it from their neighbors, leading to a potential change of their vote if the neighbors exercise a strong influence on them. With this intuition in mind, in our models, the probability distribution of the voters reached by the message is updated as a function of the degree of influence that the senders of the message have on them. The rationale is that the controller, without knowing the exact preference list which is kept hidden, can just update its estimation on it by considering the mutual degree of influence among voters. We acknowledge that our models do not cover all scenarios that can arise in election control, e.g., messages about multiple candidates. However they represent a first step towards modeling uncertainty.

We study on our models both the constructive and destructive election control problems. We show in Section 3 that the election control problem in PLTR is at least as hard to approximate as the Densest- k -Subgraph problem [Man17]. This result implies several conditional hardness of approximation bounds for our problem, for example it cannot be approximated within any constant factor, unless the Unique Game Conjecture holds and it cannot be approximated to within any polynomial factor if the Exponential Time Hypothesis holds. However, these hardness of approximation bounds do not hold for the election control problem in R-PLTR, for which we can show that the problem remains *NP*-hard.

In Section 4 we provide an algorithm that guarantees a constant factor approximation to the constructive and destructive election control problems in R-PLTR. In the relaxed model, R-PLTR, also “partially-influenced” nodes change their probability distribution. Although this simple modification is enough to make the problem substantially easier, preliminary experimental results show that the hardness of approximation for PLTR is purely theoretical and is due to hard instances in the reduction.

In Section 5 we present the simulation of our models and algorithm on two real-world datasets.

2 Influence Models and Problem Statement

Background. Influence Maximization is the problem of finding a subset of the most influential users in a social network with the aim of maximizing the spread of information given a particular diffusion model. In this work, we focus on the diffusion model known as *Linear Threshold Model* (LTM) [KKT15]. Given a graph $G = (V, E)$, each edge $(u, v) \in E$ has a weight $b_{uv} \in [0, 1]$, each node $v \in V$ has a threshold $t_v \in [0, 1]$ sampled uniformly at random and independently from the others, and the sum of the weights of the incoming edges of v is $\sum_{(u,v) \in E} b_{uv} \leq 1$. Each node can be either *active* or *inactive*. Let A_0 be a set of initially active nodes and A_t be the set of nodes active at time t . A node v becomes active if the sum of the incoming active weights at time $t - 1$ 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 $\sum_{u \in A_{t-1}: (u,v) \in E} b_{uv} \geq t_v$.

The process terminates at the first time \tilde{t} in which the set of active nodes would not change in the next round, i.e., $A_{\tilde{t}} = A_{\tilde{t}+1}$. We define the eventual set of active nodes as $A := A_{\tilde{t}}$ and the expected size of A as $\sigma(A_0)$. Given a budget B , the influence maximization problem consists in finding a set of nodes A_0 of size B , called *seeds*, in such a way that $\sigma(A_0)$ is maximum.

Kempe et al. [KKT15] showed that the distribution of active nodes A , for any set A_0 , is equal to the distribution of the sets of nodes that are *reachable* from A_0 in the set of random graphs called *live-edge graphs*. A live-edge graph is a subgraph in which each node has at most

one incoming edge. Even if the number of live-edge graphs is exponential, by using standard Chernoff-Hoeffding bounds, it is possible to compute a $(1 \pm \epsilon)$ -approximation of $\sigma(A_0)$, for a given A_0 , with high probability by sampling a polynomial number of live-edge graphs. Moreover, $\sigma(A_0)$ is monotone and submodular w.r.t. to the initial set A_0 ; hence, an optimal solution can be approximated to a factor of $1 - 1/e$ using a simple greedy algorithm [NWF78]. There has been intensive research on the problem in the last decade. We point the reader to a recent survey on the topic [LFWT18].

Notation. Let $G = (V, E)$ be a directed graph representing a social network of voters and their interactions. We denote the set of m candidates running for the election as $C = \{c_1, c_2, \dots, c_m\}$ and the target candidate as $c_\star \in C$. 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 $\sum_{i=1}^m \pi_v(c_i) = 1$. Moreover, we denote by N_v^- and N_v^+ , respectively, the sets of incoming and outgoing neighbors for each node $v \in V$. 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 $\gamma > 0$.¹ Let $X_v(c_i)$ be an indicator random variable, where $X_v(c_i) = 1$ if v votes for c_i , with probability $\pi_v(c_i)$, and $X_v(c_i) = 0$ otherwise. We define the expected *score* of a candidate c_i as the expected number of votes that c_i obtains from the voters $F(c_i, \emptyset) := \mathbf{E} \left[\sum_{v \in V} X_v(c_i) \right] = \sum_{v \in V} \pi_v(c_i)$.

PLTR Model. 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} , that models the influence of node u on v , with the constraint that, for each node v , $\sum_{u: (u,v) \in E} b_{uv} \leq 1$. We assume the weight of each existing edge (u, v) not to be too small, i.e., $b_{uv} = \Omega(1/|V|^\gamma)$ for some constant $\gamma > 0$.⁴

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. Mainly, we are modeling the spread of some ads/news about the target candidate: Active nodes receive the message and spread it to their neighbors. Moreover, in PLTR, active nodes are influenced by the message, increasing their probability of voting for the target candidate. In particular, an active node v increases the probability of voting for c_\star by an amount equal to the sum of the weights of its edges incoming from other active nodes, i.e., it adds $\sum_{u \in A \cap N_v^-} b_{uv}$ to the initial probability $\pi_v(c_\star)$. Then it normalizes to maintain π_v as 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 of v is denoted as $\tilde{\pi}_v$ and it is equal to:

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

for each $c_i \neq c_\star$. 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 . As for the expected score before the process, we can compute the expected final score of a candidate c_i as

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

where $X_v(c_i, S)$ is the indicator random variable after the process, i.e., $X_v(c_i, S) = 1$ if v votes for c_i , with probability $\tilde{\pi}_v(c_i)$, and $X_v(c_i, S) = 0$ otherwise.

¹The assumption is used in the approximation results, since Influence Maximization problem with exponential (or exponentially small) weights on nodes is an open problem. However, the assumption is realistic: Current techniques to estimate such parameters generate values linear in the number of messages shared by a node.

Let us denote by \mathcal{G} the set of all possible live-edge graphs sampled from G . We can also compute $F(c_i, S)$ by means of live-edge graphs used in the LTM model as

$$F(c_i, S) = \sum_{G' \in \mathcal{G}} F_{G'}(c_i, S) \cdot \mathbf{P}(G'), \quad (2)$$

where $F_{G'}(c_i, S)$ is the score of c_i in $G' \in \mathcal{G}$ and $\mathbf{P}(G')$ is the probability of sampling live-edge G' . More precisely, for the target candidate 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 $c_i \neq c_\star$.

R-PLTR Model. In the next section we prove that the election control problem in PLTR is hard to approximate to within a polynomial fraction of the optimum (Theorem 1). However, we 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). More formally, every node $v \in V$ (and not just every node $v \in A$ as in PLTR) changes its preference by updating its probability distribution via Eq. (1); thus also nodes that have at least an active incoming neighbor can change. 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 the next section, we show that election control in R-PLTR is still *NP*-hard, and then we give an algorithm that guarantees a constant approximation ratio in this setting.

Problem Statement. In the *constructive election control* problem we maximize the expected Margin of Victory (MoV) of the target candidate w.r.t. its most voted opponent, akin to [CCDP19a, WV18]. We define the $\text{MoV}(S)$ obtained starting from S as the expected increase, w.r.t. the value before the process, of the difference between the score of c_\star and that of the most voted opponent.² Formally, if c and \hat{c} are respectively the candidates different from c_\star with the highest score before and after the diffusion process

$$\text{MoV}(S) := F(c, \emptyset) - F(c_\star, \emptyset) - (F(\hat{c}, S) - F(c_\star, S)). \quad (3)$$

Given a budget B , the *constructive election control problem* asks to find a set of seed nodes S , of size at most B , that maximizes $\text{MoV}(S)$. It is worth noting that MoV can also be expressed as a function of the score gained by candidate c_\star and the score lost by its most voted opponent \hat{c} at the end of the process. We define the score gained and lost by a candidate c_i as

$$g^+(c_i, S) := F(c_i, S) - F(c_i, \emptyset), g^-(c_i, S) := F(c_i, \emptyset) - F(c_i, S).$$

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

$$\text{MoV}(S) = g^+(c_\star, S) + g^-(\hat{c}, S) - F(\hat{c}, \emptyset) + F(c, \emptyset). \quad (4)$$

²The increment in margin of victory, instead of just the margin, cannot be negative and gives well defined approximation ratios.

The *destructive election control problem*, instead, aims at making the target candidate lose by minimizing its MoV. In this dual scenario, the probability distributions of the voters are updated slightly differently in our models, i.e., influenced voters have a lower probability of voting for the target candidate c_\star mimicking the spread of “negative” news about c_\star .

Influencing Voters About Other Candidates. In our model the controller can send to the seed nodes a message in support of only one single candidate, e.g., latest news about the candidate. We prove that the best strategy is that of sending messages in support of the target candidate c_\star , i.e., if the controller wants c_\star to win, then, according to our models, the direct strategy of targeting voters with news about c_\star is more effective than the alternative strategy of distracting the same voters with news about other candidates.

Indeed, it is not always sufficient to maximize the score of the target candidate to ensure his victory or to maximize the margin of victory, and it is easy to find counter-examples of this strategy. Moreover, in the models of Wilder et al. [WV18] and Corò et al. [CCDP19a] it could be 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 .

However, as previously claimed, in our models this does not hold. In fact, we can distinguish between the three possible strategies:

- MoV₁: Influencing voters about c_\star .
- MoV₂: Influencing voters about \hat{c} , i.e., the most voted opponent w.r.t. c_\star at the end of the process.
- MoV₃: 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 (4), a general formulation for MoV is the following

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

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:

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

We just need to observe that $\text{MoV}_1(S) \geq \text{MoV}_2(S)$ and that $\text{MoV}_1(S) \geq \text{MoV}_3(S)$ to conclude that 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 on changing the score of the target candidate c_\star . Note that the observations above hold both for PLTR and R-PLTR.

3 Hardness Results

In this section we provide two hardness results related to election control in PLTR and R-PLTR. In Theorem 1 we show that maximizing the MoV in PLTR is at least as hard to approximate as the Densest- k -subgraph problem. This 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 [RS10]; to within $n^{-1/(\log \log n)^c}$, for some constant c , under the exponential time hypothesis (ETH) [Man17]; to $n^{-f(n)}$ for any function $f \in o(1)$, under the Gap-ETH assumption [Man17]. Then, in Theorem 2, we show that maximizing the MoV in R-PLTR is still NP -hard.

Theorem 1. *An α -approximation algorithm 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. Given an undirected graph $G = (V, E)$ and an integer k , Densest k -Subgraph (DkS) is the problem of finding the subgraph induced by a subset of V of size k with the highest number of edges given that k is fixed.

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 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, the score of \hat{c} after PLTR starting from any initial set S is

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

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

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

To compute the expected final score of the target candidate we average its score in all live live-edge graph in \mathcal{G} , according to Formula (3). In our reduction, the empty live-edge graph $G'_\emptyset = (V, \emptyset)$ is sampled *with high probability*, i.e., with probability at least $1 - \frac{1}{n^{\gamma-2}}$:

$$\mathbf{P}(G'_\emptyset) = \prod_{v \in V} \left(1 - \sum_{u \in N_v^-} b_{uv} \right) = \prod_{v \in V} \left(1 - \frac{|N_v^-|}{n^\gamma} \right)$$

$$\begin{aligned}
&\geq \prod_{v \in V} \left(1 - \frac{1}{n^{\gamma-1}}\right) = \left(1 - \frac{1}{n^{\gamma-1}}\right)^n \\
&\stackrel{(a)}{=} \sum_{i=0}^n \binom{n}{i} (1)^{n-i} \left(\frac{-1}{n^{\gamma-1}}\right)^i = \sum_{i=0}^n \binom{n}{i} \frac{(-1)^i}{n^{i(\gamma-1)}} \\
&\stackrel{(b)}{\geq} \binom{n}{0} - \binom{n}{1} \frac{1}{n^{\gamma-1}} + \sum_{i=2}^{\lfloor n/2 \rfloor} \left(\binom{n}{i} \frac{1}{n^{2i(\gamma-1)}} - \binom{n}{i+1} \frac{1}{n^{(2i+1)(\gamma-1)}} \right) \\
&\stackrel{(c)}{\geq} 1 - \frac{1}{n^{\gamma-2}}
\end{aligned}$$

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

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

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

since $1 \leq 1 + \sum_{u \in R_{G'}(S) \cap N_v^-} \frac{1}{n^\gamma} \leq 2$ for each $v \in R_{G'}(S)$. Also, 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

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

where $\text{SOL}_{DkS}(S) := \sum_{v \in S} |S \cap N_v^-|$ 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

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

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

$$\begin{aligned} \sum_{G' \neq G_0} F_{G'}(c_\star, S) \cdot \mathbf{P}(G') &= \mathcal{O}\left(\frac{1}{n^{\gamma-2}}\right) \sum_{G' \neq G_0} \mathbf{P}(G') \\ &= \mathcal{O}\left(\frac{1}{n^{2(\gamma-2)}}\right) = \mathcal{O}\left(\frac{\text{SOL}_{DkS}(S)}{n^\gamma}\right), \end{aligned}$$

for any $\gamma \geq 4$. Thus

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

which means that $F(c_\star, S) = \Theta\left(\frac{\text{SOL}_{DkS}(S)}{n^\gamma}\right)$. 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 $n > n_0$,

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

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 $\beta_1 \frac{\text{OPT}_{DkS}}{n^\gamma} \leq \text{OPT} \leq \beta_2 \frac{\text{OPT}_{DkS}}{n^\gamma}$, where OPT is the value of an optimal solution for PLTR and OPT_{DkS} is the value of an optimal solution for DkS.

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

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

Thus $\text{SOL}_{DkS}(S) \geq \frac{\alpha \beta_1}{2 \beta_2} \text{OPT}_{DkS}$, i.e., it is an $\frac{\alpha \beta_1}{2 \beta_2}$ -approximation to DkS. \square

As a corollary of Theorem 1 we get the conditional hardness of approximation bounds stated at the beginning of this section.

Theorem 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 [KKT15].

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

1. Duplicate each vertex in the graph, i.e., we define the new set of nodes as $V' := V \cup \{v_{|V|+1}, \dots, v_{2|V|}\}$.
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., $w'(e) = w(e)$ for each $e \in E$ and $w'(e) = 1$ for each $e \in E' \setminus E$. Note that the constraint on incoming weights required by LTM is not violated by w' .
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}_{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 $\text{MoV}(S) = |V| - |V \setminus A|$. Let us assume that S does not maximize \mathcal{I}_{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., $F(c_\star, S) = |V| + \frac{1}{2}|A|$, is exactly equivalent to maximize the cardinality of the active nodes in LTM. \square

4 Approximation Results

In this section, we first show that we can approximate the optimal MoV to within a constant factor by optimizing the increment in the score of c_\star . In detail we show that, given two solutions S^* and S^{**} such that $g^+(c_\star, S^*)$ and $\text{MoV}(S^{**})$ are maximum, then $\text{MoV}(S^*) \geq \frac{1}{3}\text{MoV}(S^{**})$. Indeed, we show a more general statement that is: If a solution S approximates $g^+(c_\star, S^*)$ within a factor α , then $\text{MoV}(S) \geq \frac{\alpha}{3}\text{MoV}(S^{**})$.

Then we show that a simple greedy hill-climbing approach (Algorithm 1) 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 the two results, we get a $\frac{1}{6}(1 - \frac{1}{e})$ -approximation algorithm for the election control problem in R-PLTR.

The next theorem generalizes [WV18, 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 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.

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

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

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

where the last inequality holds because $g^+(c_\star, S) \geq g^-(c_i, S)$ for any solution S and candidate c_i since S modifies only the score of c_\star , increasing it, while the score of all the other candidates

is decreased, and the increment in score to c^* is equal to the sum of the decrement in score of all the other candidates. Since $F(\hat{c}) \leq F(c)$, we have that

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

where \bar{c} is the candidate with the highest score after the solution S^* is applied. By definition of \bar{c} we have that $F(\bar{c}, S^*) \geq F(\hat{c}, S^*)$, which implies that

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

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

Constructive Election Control in R-PLTR. Next theorem shows how to get a constant factor approximation to the problem of maximizing the MoV in R-PLTR by reducing the problem to an instance of the weighted version of the influence maximization problem with LTM [KKT15].

This extension of the LTM, associates to each node a non-negative weight ($w : V \rightarrow \mathbb{R}^+$) that captures the importance of activating that node. The goal is 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 $\arg \max_S \sigma_w(S) = \mathbf{E} [\sum_{v \in A} w(v)]$, where w is a weight function over the node set.

Algorithm 1 GREEDYSCORE

Require: Social graph $G = (V, E)$; Budget B

- 1: $S = \emptyset$; $\hat{G} = (G, w)$ \triangleright Weighted graph \hat{G}
 - 2: **while** $|S| \leq B$ **do**
 - 3: $v = \arg \max_{u \in V \setminus S} \sigma_w(S \cup \{u\}) - \sigma_w(S)$
 - 4: $S = S \cup \{v\}$
 - 5: **return** S
-

A simple hill-climbing greedy algorithm achieves a $(1 - 1/e)$ -approximation if the weights are polynomial (or polynomially small) in the number of nodes of the graph and the number of live-edge graph samples is polynomially large in the weights [KKT15].³ 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 via Algorithm 1, 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 $w(v) := \sum_{u \in N_v^+} b_{vu}(1 - \pi_u(c_*))$. Note that we are able to correctly approximate the value of $\sigma_w(S)$ using such weights since by hypothesis on the model $b_{uv} \geq \frac{1}{|V|^{\gamma_1}}$, for each $(u, v) \in E$ and for some constant $\gamma_1 > 0$, and since $\pi_v(c_i) \geq \frac{1}{|V|^{\gamma_2}}$, for each $v \in V$ for some constant $\gamma_2 > 0$. By applying a multiplicative form of the Chernoff bound we can get a $1 \pm \epsilon$ approximation of $\sigma_w(S)$, with high probability [KKT15, Proposition 4.1].

Thus, we can use Algorithm 1 to maximize the influence on \hat{G} . The algorithm starts with an empty set S and adds to it, in each of B rounds, the node v with maximal marginal gain w.r.t. the solution computed so far.

³It is still an open question how well the value of $\sigma_w(S)$ can be approximated for an influence model with arbitrary node weights.

Theorem 4. *Algorithm 1 guarantees a $\frac{1}{6}(1 - \frac{1}{e})$ -approximation factor to constructive election control in R-PLTR.*

Proof. We first prove that Algorithm 1 gives an $\frac{1}{2}(1 - \frac{1}{e})$ -approximation 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

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

and, since the denominator is at most 2, that

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

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. [KKT15] 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 $g^+(c_\star, S) \geq \frac{1}{2}(1 - 1/e) g^+(c_\star, S^\star)$ since

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

where the inequality holds since all the denominators in $g^+(c_\star, S^\star)$ are at least 1. Thus, Algorithm 1 achieves a $\frac{1}{2}(1 - \frac{1}{e})$ -approximation to the maximum increment in score. Using Theorem 3 we get a $\frac{1}{6}(1 - \frac{1}{e})$ -approximation for the MoV. \square

Destructive Election Control in R-PLTR. The *destructive election control* problem is similar to the constructive problem, but in this scenario, in our models, the probability that a voter v votes for c_\star *decreases* depending on the amount of influence received by v and the loss of probability of c_\star is evenly split over all the other candidates. In this way, we avoid negative values and values that do not sum to 1. In detail, if A is the set of active nodes at the end of LTM, then, for each $v \in V$, the preference list π_v changes as follows:

$$\tilde{\pi}_v(c_\star) = \frac{\pi_v(c_\star)}{1 + \sum_{u \in A \cap N_v^-} b_{uv}} \quad \text{and} \quad \tilde{\pi}_v(c_i) = \frac{\pi_v(c_i) + \frac{1}{m-1} \sum_{u \in A \cap N_v^-} b_{uv}}{1 + \sum_{u \in A \cap N_v^-} b_{uv}}$$

for each $c_i \neq c_\star$. We define MoV_D , i.e., what we want to maximize, as

$$\text{MoV}_D(S) := F(\hat{c}, S) - F(c_\star, S) - (F(c, \emptyset) - F(c_\star, \emptyset)) = g^-(c_\star, S) + g^+(\hat{c}, S) + \Delta,$$

where S is the initial set of seed nodes and $\Delta = F(\hat{c}, \emptyset) - F(c, \emptyset)$ is the sum of constant terms that are not modified by the process. Note that maximizing MoV_D is NP -hard (it can be proved with a similar argument to that of Theorem 2).

Similarly to the constructive case, 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 $w(v) := \sum_{u \in N_v^+} b_{vu} \pi_u(c_\star)$ and we run Algorithm 1 to find a seed set that approximates the maximum expected weight of active nodes.

Theorem 5. *Algorithm 1 guarantees a $\frac{1}{4}(1 - \frac{1}{e})$ -approximation factor to the destructive election control in R-PLTR.*

Proof. We first prove that Algorithm 1 achieves an $\frac{1}{2}(1 - \frac{1}{e})$ approximation factor to the problem of maximizing the decrease 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. Let $g_D^-(c_\star, S)$ be the decrease in score of candidate c_\star with solution S , i.e., $g_D^-(c_\star, S) = F(c_\star, \emptyset) - F(c_\star, S)$. Let A be the set of active nodes at the end of the process; then we have that

$$g_D^-(c_\star, S) = \sum_{v \in V} \frac{\pi_v(c_\star) \sum_{u \in A \cap N_v^-} b_{uv}}{1 + \sum_{u \in A \cap N_v^-} b_{uv}}$$

and, since the denominator is at most 2, that

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

Note that $\sum_{u \in A} \sum_{v \in N_u^+} \pi_v(c_\star) \cdot b_{uv}$ is exactly the objective function of the greedy Algorithm that maximizes the weighted-LTM for \hat{G} . Hence, using the result by Kempe et al. [KKT15], we know that

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

where A^\star is the optimal set of active nodes, i.e., the set of active nodes at the end process starting from S^\star (S^\star the optimal solution for the weighted-LTM).

Therefore

$$g_D^-(c_\star, S) \geq \frac{1}{2} \left(1 - \frac{1}{e}\right) g_D^-(c_\star, S^\star)$$

because

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

where the inequality is due to the fact that the denominator in all the terms of $g_D^-(c_\star, S^\star)$ is at least 1. Thus we achieve a $\frac{1}{2}(1 - \frac{1}{e})$ -approximation to the maximum increment in score.

Let us fix c and \hat{c} , respectively, as the candidates different from c_\star with the highest score before and after the solution S is applied; let \bar{c} be the most voted opponent after the optimal solution S^\star is applied. Then we have that

$$\text{MoV}(S) = g_D^-(c_\star, S) + g_D^+(\hat{c}, S) + F(\hat{c}, \emptyset) - F(c, \emptyset)$$

$$\begin{aligned}
&\geq \frac{1}{2} \left(1 - \frac{1}{e}\right) g_D^-(c_*, S^*) + g_D^+(\hat{c}, S) + F(\hat{c}, \emptyset) - F(c, \emptyset) \\
&\geq \frac{1}{4} \left(1 - \frac{1}{e}\right) [g_D^-(c_*, S^*) + g_D^+(\bar{c}, S^*) + g_D^+(\hat{c}, S) + F(\hat{c}, \emptyset) - F(c, \emptyset)] \\
&= \frac{1}{4} \left(1 - \frac{1}{e}\right) [\text{MoV}(S^*) + g_D^+(\hat{c}, S) + F(\hat{c}, \emptyset) - F(\bar{c}, \emptyset)] \\
&\geq \frac{1}{4} \left(1 - \frac{1}{e}\right) \text{MoV}(S^*),
\end{aligned}$$

where the last inequality holds since, by definition of \bar{c} and \hat{c} , we have that $g_D^+(\hat{c}, S) + F(\hat{c}, \emptyset) \geq g_D^+(\bar{c}, S) + F(\bar{c}, \emptyset)$. \square

5 Simulations

We simulate our model on two real-world social networks⁴ on which political campaigning messages could spread:

- *polbooks*: an undirected network with 105 nodes and 882 edges where nodes are political books and edges represent co-purchasing behavior; nodes are labeled as “liberal,” “conservative,” or “neutral.”
- *polblogs*: a directed network with 1,224 nodes and 19,025 edges where nodes are web blogs about US politics and edges hyperlinks connecting them; nodes are labeled as “liberal” or “conservative.”

The number of candidates in our simulations is based on the ground truth of the datasets; as mentioned earlier, *polbooks* has three clusters and *polblogs* has two clusters based on different US political parties. We set the probability of each node v to vote for, say, a “liberal” candidate proportionally to the number of neighbors labeled as “liberal,” i.e., we set $\pi_v(c) = \frac{|N_v \cap B|}{|N_v|}$ where c is the “liberal” candidate, B is the set of nodes labeled as “liberal,” and N_v is the set of neighbors of v . For each node v we sampled the “non-incoming influence weight” \bar{b}_v uniformly at random in $[0, 1]$ and assigned the remaining influence weight uniformly among its incoming neighbors, i.e., we assigned to each edge (u, v) a weight $b_{uv} = \frac{1 - \bar{b}_v}{|N_v^-|}$.

In our simulations, we run *GreedyScore* (Algorithm 1) for the election control problem in R-PLTR. Then, we measure the score and the MoV of each candidate using as starting seed nodes the ones found by the algorithm both in PLTR and in R-PLTR. We run the simulation considering each different candidate as the target one to cover multiple scenarios, considering as budget values the ones in $\{0, 1, 5, 10\}$. Then, as baseline to compare, we also considered as seed nodes the most influential ones, i.e., the nodes selected by *GreedyIM*, the classical greedy algorithm for Influence Maximization [KKT15].

For the implementation, we used .Net framework 4.6.2 and C# programming language. We have implemented five different classes for managing the graph, the LTM process, the PLTR process, and a GUI. We execute the simulations on a system with the following specifications: CPU Intel Core i7-6700HQ 2.6 GHz, with 4×32 KB 8-way L1 (data and inst) cache, and 4×256 KB 4-way L2 cache, and 6 MB 12-way L3 cache, RAM 16G DDR4. Each simulation has a running time of approximately 40 seconds for *polbooks* and 140 minutes for *polblogs*.

The results relative to the scores are shown in Figures 1 and 2. As expected, the effect of our algorithm in R-PLTR is amplified compared to PLTR, since it affects a greater number of

⁴The datasets are taken from <http://networkrepository.com/>

voters. Taking as example the “liberal” candidate in *polbooks*, we need a budget $B = 5$ to make it overtake the “conservative” candidate in PLTR, while a budget $B = 1$ is enough in R-PLTR (Figure 1); in *polblogs*, instead, we are not able to make the “liberal” candidate win in PLTR with budget $B = 10$, but it is enough a budget $B = 5$ to make it overtake the “conservative” candidate in R-PLTR (Figure 2).

The results relative to MoV are presented in Figure 3. We can note that, as a general trend, candidates with lower probability of winning, are the most affected by the influence generated by the seed nodes selected by our algorithm both in PLTR and R-PLTR. The “neutral” and “liberal” candidates, respectively last and second last voted, have the higher MoV in *polbooks* (see Figure 3, on the left), while the “liberal” candidate, which was losing the elections, has the higher MoV in *polblogs* (see Figure 3, on the right).

Finally, in Figure 4 we present the difference between the MoV calculated by *GreedyScore* and *GreedyIM*. The simulations show that our algorithm outperforms *GreedyIM*, as expected. The only scenario in which our algorithm performs worse is that in which we influence, with low budget, the already winning candidate (see Figure 4, on the left, red lines). The reason why *GreedyScore* works better than *GreedyIM* is that it looks for seeds that will influence “critical” voters, i.e., voters on which the influence will have more impact on the global score of the candidates, while *GreedyIM* just looks for influential voters, independently from their initial opinion.

6 Conclusions and Future Work

Influencing elections by means of social networks is a significant issue in modern society, and understanding this phenomenon is of crucial importance in order to prevent the integrity of democracy. Our results constitute the first step towards realistic modeling of the use of social influence to control elections as our models take into account that voters might hide their preferences to a controller. In one of our models the election control problem cannot be approximated within any reasonable bound, under some computational complexity hypothesis. For the other model we provide an approximation algorithm that guarantees a constant factor approximation ratio. The results in this paper open several research directions. We plan to study the election control problem in a variant of PLTR where multiple campaigns affect voters’ opinions on different candidates. It is also worth to investigate models with uncertainty in other voting systems. Finally, it would be interesting 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.

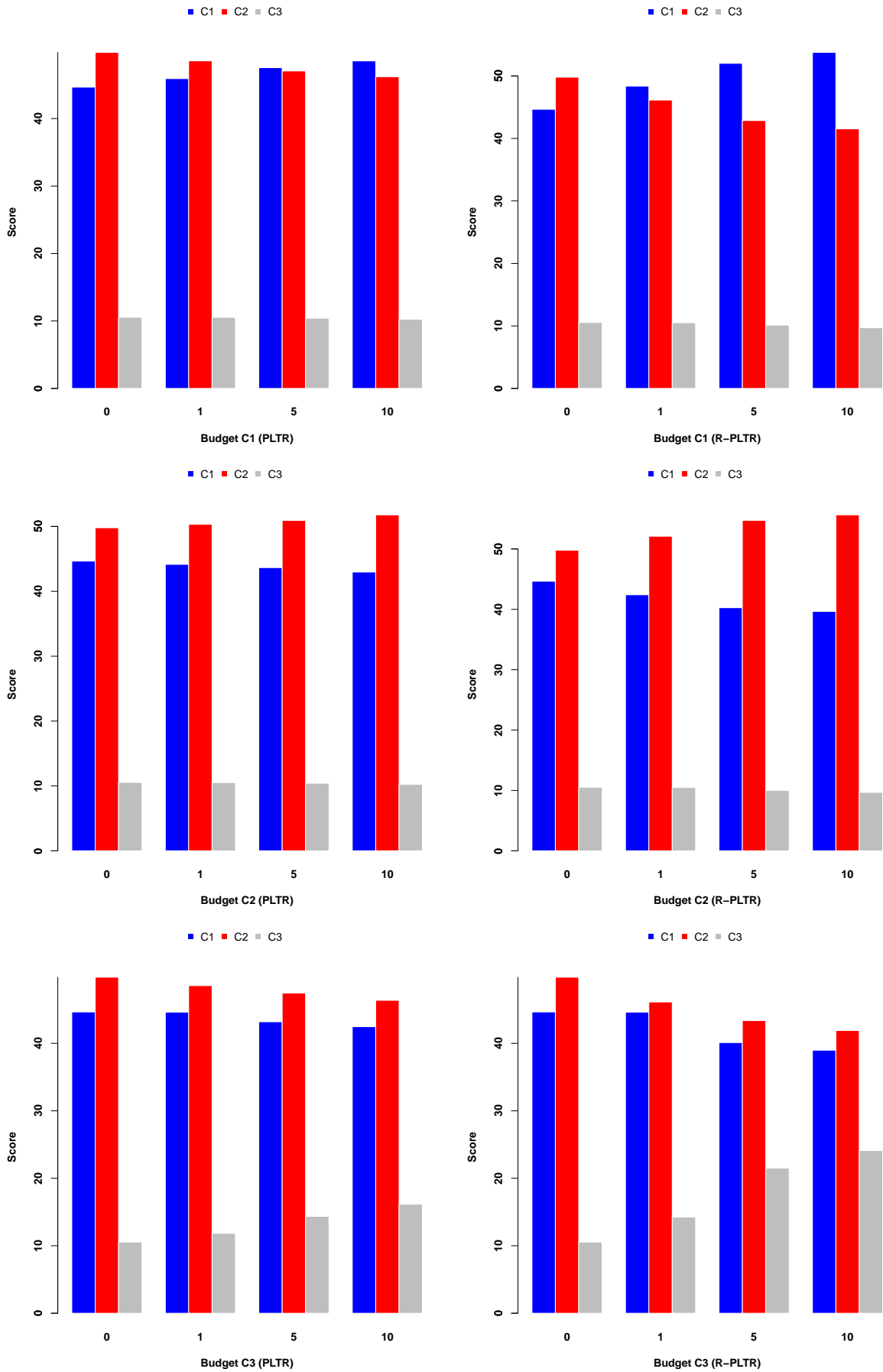


Figure 1: Candidates' scores in *polbooks* in PLTR (left column) and R-PLTR (right column), considering as target candidate the “liberal” (top), the “conservative” (center), and the “neutral” (bottom).

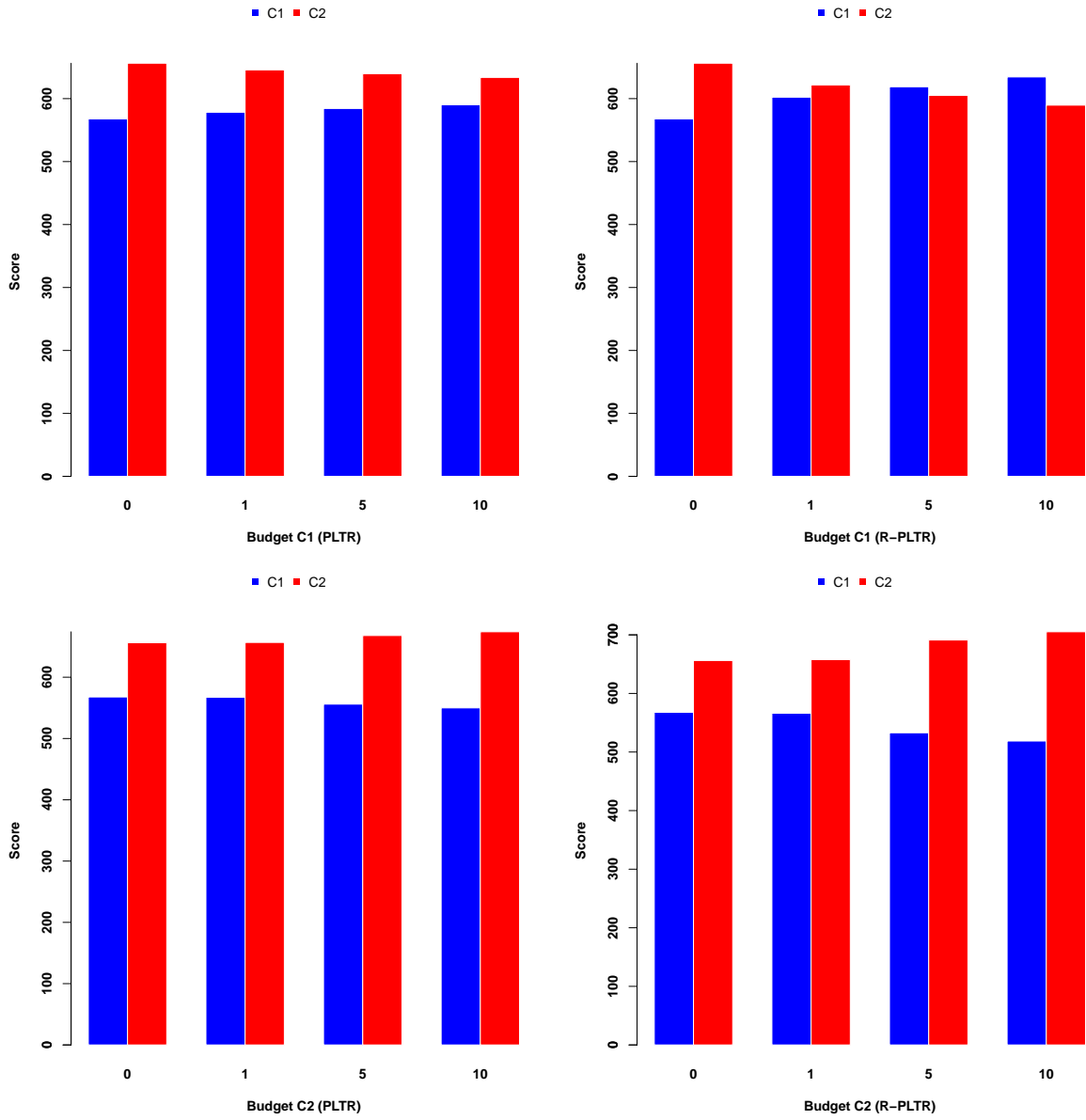


Figure 2: Candidates' scores in *polblogs* in PLTR (left column) and R-PLTR (right column), considering as target candidate the “liberal” (top) and the “conservative” (bottom).

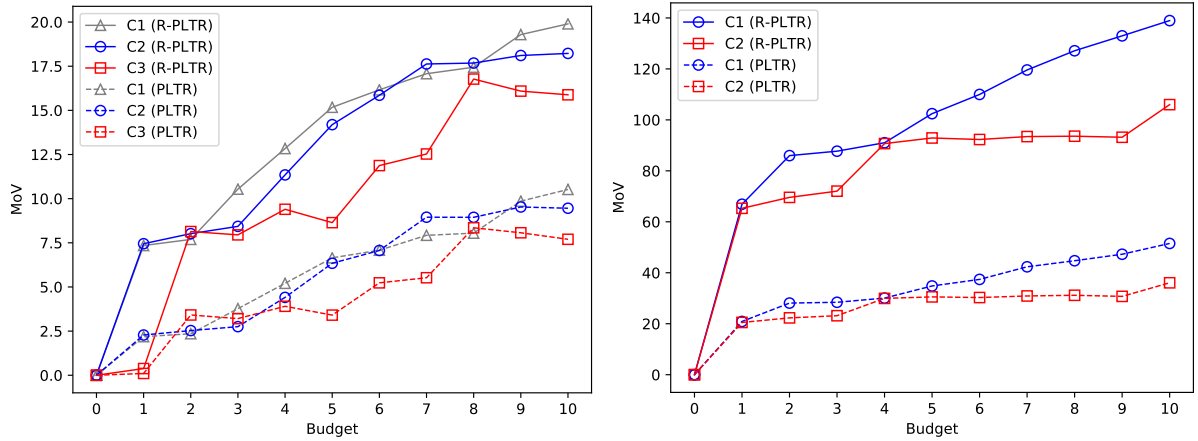


Figure 3: The MoV calculated using the presented algorithm for *polbooks* (left) and *polblogs* (right), both in PLTR (dashed line) and R-PLTR (solid line), considering as target candidate the “liberal” (blue line), the “conservative” (red line), and the “neutral” (grey line).

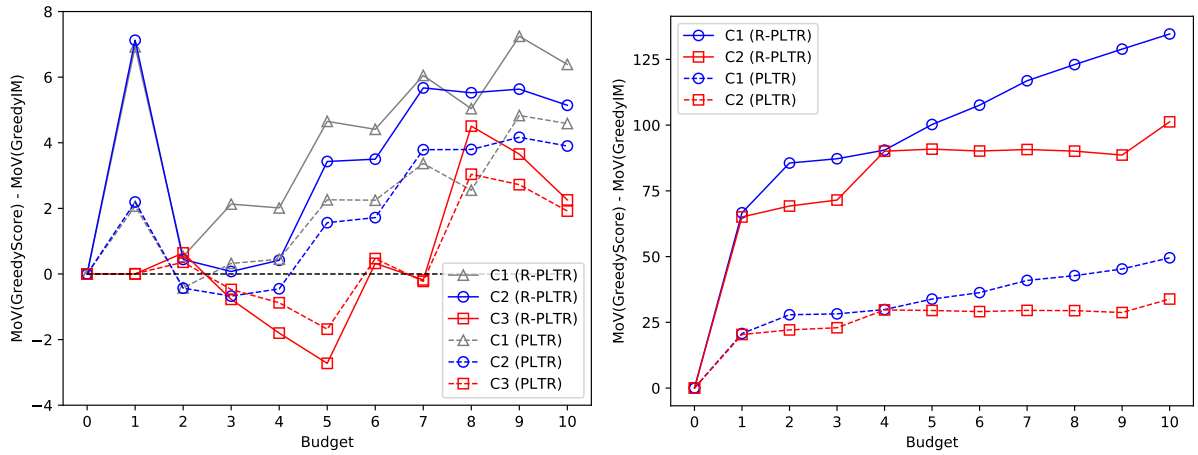


Figure 4: Difference between MoV obtained using our greedy algorithm and MoV obtained using the standard greedy algorithm for Influence Maximization problem in *polbooks* (left) and *polblogs* (right). Values greater than 0 are when our algorithm performs better than the simple Greedy for Influence Maximization.

References

- [AD20] Mohammad Abouei Mehrizi and Gianlorenzo D’Angelo. Multi-winner election control via social influence. In *Proc. of SIROCCO 2020*, 2020. To appear.
- [AG17] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *J. Economic Perspectives*, 31(2):211–36, 2017.
- [AM14] Krzysztof R. Apt and Evangelos Markakis. Social networks with competing products. *Fundam. Inform.*, 129(3):225–250, 2014.
- [BE17] Robert Brederick and Edith Elkind. Manipulating opinion diffusion in social networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, pages 894–900, 2017.
- [BFJ⁺12] 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. *Nat.*, 489(7415):295–298, 2012.
- [CCDP19a] Federico Corò, Emilio Cruciani, Gianlorenzo D’Angelo, and Stefano Ponziani. Exploiting social influence to control elections based on scoring rules. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 201–207, 2019.
- [CCDP19b] 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 ’19, Montreal, QC, Canada, May 13-17, 2019*, pages 1895–1897, 2019.
- [Fer17] Emilio Ferrara. Disinformation and social bot operations in the run up to the 2017 french presidential election. *First Monday*, 22(8), 2017.
- [FGKT18] Piotr Faliszewski, Rica Gonen, Martin Koutecký, and Nimrod Talmon. Opinion diffusion and campaigning on society graphs. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 219–225, 2018.
- [FR16] Piotr Faliszewski and Jörg Rothe. Control and bribery in voting. In *Handbook of Computational Social Choice*, pages 146–168. 2016.
- [KKT15] David Kempe, Jon M. Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory Comput.*, 11:105–147, 2015.
- [Kre16] Daniel Kreiss. Seizing the moment: The presidential campaigns’ use of twitter during the 2012 electoral cycle. *New Media & Society*, 18(8):1473–1490, 2016.
- [LFWT18] Yuchen Li, Ju Fan, Yanhao Wang, and Kian-Lee Tan. Influence maximization on social graphs: A survey. *IEEE Trans. Knowl. Data Eng.*, 30(10):1852–1872, 2018.

- [Man17] 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 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 954–961, 2017.
- [MS18] Katerina Eva Matsa and Elisa Shearer. News use across social media platforms 2018. *Pew Research Center*, 2018. Accessed Feb, 9th, 2019.
- [NWF78] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions—I. *Mathematical Programming*, 14(1):265–294, Dec 1978.
- [PCR18] Gordon Pennycook, Tyrone Cannon, and David G Rand. Prior exposure increases perceived accuracy of fake news. *Journal of Experimental Psychology: General*, 2018.
- [RS10] Prasad Raghavendra and David Steurer. Graph expansion and the unique games conjecture. In *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010*, pages 755–764, 2010.
- [SBS18] 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.
- [WV18] Bryan Wilder and Yevgeniy Vorobeychik. Controlling elections through social influence. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS 2018, Stockholm, Sweden, July 10-15, 2018*, pages 265–273, 2018.