Election Control with Voters’ Uncertainty: Hardness and Approximation Results

Mohammad Aboueimehrizi, Federico Corò, Emilio Cruciani, Gianlorenzo D’Angelo

GSSI – Gran Sasso Science Institute
Viale Francesco Crispi 7, 67100, L’Aquila, Italy

mohammad.aboueimehrizi@gssi.it
federico.coro@gssi.it
emilio.cruciani@gssi.it
gianlorenzo.dangelo@gssi.it

Abstract

The election control problem asks to find a set of nodes in a social network of voters to be the starters of a political campaign aimed at supporting a given target candidate. When a voter is reached by the campaign it changes its opinion 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 works on this problem show that it can be approximated within a constant ratio in several models for information diffusion and voting systems. These works assume the 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. Herein 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.
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 (Facebook, Twitter, etc.) before news sites as they provide a faster means of information diffusion. 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 started from a small set of individuals may generate a cascade effect that allows the initial influence to be diffused to 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, for example by spreading fake news, on the elections outcome [PCR18]. A real-life example is in recent US election where a study on the effect of social media on people during the presidential election shows that in average ninety-two percent of American remembered pro-Trump false news, and twenty-three percent of them remembered the pro-Clinton fake news [AG17]. We point the reader to many other real-life examples that have been recorded and studied [SBLS18, Kre16, BFJ+12, Fer17].

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 election control problem has been 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 [WV18]. They use the Independent Cascade Model [KKT15] 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. Corò et al. studied the same problem by using the Linear Threshold Model (LTM) [KKT15] and an arbitrary scoring rule [CCDP19]. In both cases, the election control problem can be solved with an approximation factor of $\frac{1}{3}(1 - 1/e)$ in the constructive scenario and $\frac{1}{2}(1 - 1/e)$ in the destructive one. Faliszewski et al. studied a variant of the Linear Threshold Model with weights on the vertices 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 [FGKT18]. 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. Bredereck et al. studied the problem of manipulating the network by bribing or adding/deleting edges in order to control the majority opinion on a simple Linear Threshold Model where each node holds a binary opinion, each edge has the same fixed weight, and all vertices have a threshold fixed to 1/2 [BE17]. In their work they studied the hardness and the parameterized complexity of the manipulations problem they proposed.

In all these previous works, it is assumed that the preference list of each voter on
the candidates is given in input. However, this assumption is not always satisfied in realistic scenarios. In fact, voters can be undecided on their preferences or they may not reveal them to the manipulator. In this paper we consider a more realistic model that takes into account the uncertainty of a voter and where the preference list of a voter is not fully revealed. Specifically, we use an uncertain model in which the manipulator only knows the probability distribution of the most preferred candidate of each voter, instead of a deterministic preference list. In this model we study the election control problem introduced in [WV18, CCDP19], that is, we look for a bounded set of nodes to start the information diffusion in such a way that the chances of victory of a given target candidate are maximized.

1.1 Related Work

There is a wide literature about manipulating a voting system without considering the underlying social network of the voters, e.g., swap bribery [EFS09], shift bribery [BFNT16]; we point the reader to a recent survey [FRM16].

The study of opinion diffusion modeled as a majority dynamics has been widely considered [ACF+15, BEEG16, BGP17]. In these models each agent has an initial preference list and at each time step a subset of agents updates their opinions, i.e., their preference lists, according to some majority-based rule that depends on their neighbors’ opinions in the network.

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 [KL05, BHN09, XC11], or models in which candidates may change during the election campaign [CLMM10, BRR11], or the vote of a bribed voter may or may not be counted [CXX+18].

1.2 Roadmap

In Section 3, we introduce two new models called Probabilistic Linear Threshold Ranking (PLTR) and Relaxed Probabilistic Linear Threshold Ranking (R-PLTR) which use LTM for information diffusion and plurality voting as a voting system. In both models each voter has an associated probability distribution over the candidates, which is changed during the diffusion process according to the amount of influence received from the neighbors. In PLTR only active voters, i.e., nodes that propagate the influence, can change their probability distribution; in R-PLTR all voters might change it if they have active neighbors.

In Section 4, we show that the election control problem in PLTR is at least as hard to approximate as the well-known 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 5, we give an algorithm for the election control problem (both constructive and destructive) in R-PLTR that guarantees constant factor approximations. This result along with the hardness results show a separation between the two new models: the election control problem is hard to be approximated to within any reasonable bound in PLTR but it can be approximated to within a constant by slightly relaxing the model, i.e., in R-PLTR.

2 Background

In this section we present some notions and concepts about voting systems and influence maximization on social networks that will be used in the design and analysis of the algorithm.

2.1 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. Social choice theory formally defines and analyzes voting systems, studying how the combination of individual opinions reaches a collective decision; computational social choice, instead, is the study of the computational complexity of outcomes of voting rules to pose a barrier against strategic manipulation in elections [CELM07, FP10, BCE+16, End17].

In this paper we focus on a single-winner voting system, plurality rule, that is the most common used for national legislatures and presidential elections. In this voting system each voter submits a single candidates among the candidates and a candidate wins if it has the majority of votes (the absolute majority, i.e., 50%+1 of votes, is not required).

2.2 Influence Maximization

Influence Maximization problem studies a social network with the goal of finding a fixed sized subset of users that are the most influential and that can be used to maximize the spread of information given a particular diffusion model [KKT15]. In this work we focus on the diffusion model known as Linear Threshold Model (LTM) [KKT15]. Given a graph $G = (V, E)$, in LTM 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 additional constraint that, for each node $v \in V$, the sum of the weights of the incoming edges of $v$ is less or equal to 1, i.e., $\sum_{(u, v) \in E} b_{uv} \leq 1$. 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 and activate them. Let $A_0$ be the set of nodes that are active at the beginning of the process and let $A_t$ be the set of nodes active at time $t$. In LTM,
an inactive node $v$ becomes active if the sum of the weights of the edges coming from nodes that are active at the previous round is greater than or equal to its threshold $t_v$, i.e., $v \in A_t$ if and only if $v \in A_{t-1} \lor \sum_{u \in A_{t-1} \cap (u,v) \in E} b_{uv} \geq t_v$. 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 $A := A_{\tilde{t}}$ and the expected size of $A$ as $\sigma(A_0)$, where $A_0$ is the initial set of active nodes. Given a budget $B$, the influence maximization problem consists in computing an initial set $A_0$ of $B$ active nodes, called seeds, in such a way that $\sigma(A_0)$ is maximum.

Given $A_0$, computing $\sigma(A_0)$ is $\#P$-hard [CYZ10], however, Kempe et al. showed a way to compute it by using a random graph model called live-edge graphs [KKT15]. Given a graph $G = (V,E)$, a live-edge graph is generated from $G$ as follows: each node $v \in V$, independently, selects at most one of its incoming edges with probability proportional to the weight of that edge, that is edge $(u,v)$ is selected with probability $b_{uv}$, and no edge is selected with probability $1 - \sum_{(u,v) \in E} b_{uv}$. We denote by $\mathcal{G}$ the set of all possible live-edge graphs that can be generated from $G$. Kempe et al. showed that for any initial set of nodes $A_0$, the distribution of the sets of active nodes in $G$ after 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 $\mathcal{G}$. Therefore, we can compute $\sigma(A_0)$ by solving a graph reachability problem in all the live-edge graphs in $\mathcal{G}$. Moreover, even if the number of graphs in $\mathcal{G}$ is exponential in the size of $G$, by using standard Chernoff-Hoeffding bounds, they showed that a polynomial number of Monte-Carlo simulations are enough to compute a $1 \pm \epsilon$ approximation of $\sigma(A_0)$ with high probability. They have also shown that $\sigma(A_0)$ is monotone and submodular wrt the set of seed nodes, therefore the classical 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 guarantees a $(1 - 1/e)$-approximation for the influence maximization problem [NWF78].

2.3 Election Control

The use of diffusion processes to manipulate elections captures the scenario in which a social network of voters is targeted by political campaigns. Election control can be used to either ensure that a target candidate wins (constructive mode), or loses (destructive mode). Wilder et al. [WV18] studied this problem under Independent Cascade Model. They provide positive and negative results for the maximization of, respectively, margin of victory (MoV) and probability of victory of a target candidate w.r.t. its most voted opponent. Corò et al. [CCDP19] introduced a new model called Linear Threshold Ranking (LTR), based on LTM, that takes into account the degree of influence that voters exercise on each other. They show that in LTR it is possible to achieve a constant factor approximation for MoV for any scoring rule.
3 Election Control with Voters’ Uncertainty

We consider a non-deterministic scenario in which a set of candidates are running for the elections and a social network of voters will decide the winner. In particular, we take into account the inherent uncertainty of a voter and we model its decision as a probabilistic function over the list of candidates.

3.1 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 $C = \{c_1, c_2, \ldots, c_m\}$ and the target candidate as $c^* \in C$. Each node $v \in V$ has a probability distribution over the candidates $\pi_v$, 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 $N^i_v$ and $N^o_v$, 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 $\gamma > 0$. We define the score of a candidate $c_i$ as the expected number of votes that $c_i$ obtains from the voters, that is,

$$F(c_i) := \sum_{v \in V} \pi_v(c_i).$$

3.2 PLTR Model

We introduce a variation of the Linear Threshold Model (LTM) [KKT15] called Probabilistic Linear Threshold Ranking (PLTR). Similarly to the Linear Threshold Ranking [CCDP19], 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.

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 $\pi_v$ changes as follows:

$$\tilde{\pi}_v(c^*) = \frac{\pi_v(c^*) + \sum_{u \in A \cap N^i_v} b_{uv}}{1 + \sum_{u \in A \cap N^o_v} b_{uv}},$$

(1)
while for any other candidate \( c_i \neq c^* \) it changes to

\[
\tilde{\pi}_v(c_i) = \frac{\pi_v(c_i)}{1 + \sum_{u \in A \cap N_v} b_{uv}}.
\] (2)

All inactive nodes \( v \in V \setminus A \) will have \( \tilde{\pi}_v(c_i) = \pi_v(c_i) \) for all candidates, including \( c^* \). Then, we denote the score of a candidate \( c_i \) at the end of the process as

\[
F(c_i, S) := \sum_{v \in V} \tilde{\pi}_v(c_i).
\]

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

\[
F(c_i, S) := \sum_{G' \in \mathcal{G}} F_{G'}(c_i, S) \cdot \mathbf{P}(G'),
\] (3)

where \( F_{G'}(c_i, S) \) is the score of \( c_i \) in \( G' \in \mathcal{G} \). More precisely, for \( c_i = c^* \) we have

\[
F_{G'}(c^*, S) = \sum_{v \in R_{G'}(S)} \frac{\pi_v(c^*) + \sum_{u \in R_{G'}(S) \cap N_v} b_{uv}}{1 + \sum_{u \in R_{G'}(S) \cap N_v} b_{uv}} \sum_{v \in V \setminus R_{G'}(S)} \pi_v(c^*),
\]

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

\[
F_{G'}(c_i, S) = \sum_{v \in R_{G'}(S)} \frac{\pi_v(c_i)}{1 + \sum_{u \in R_{G'}(S) \cap N_v} b_{uv}}.
\]

### 3.3 R-PLTR Model

In Section 4, we show that the election control problem in PLTR is hard to approximate to within a polynomial fraction of the optimum (Theorem 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 Relaxing 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 (1) and (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 4 we show that the election control problem in R-PLTR is still NP-hard, and in Section 5 we give an algorithm that guarantees a constant approximation ratio in this setting.
3.4 Objective Function

In the election control problem we maximize the expected Margin of Victory (MoV) of the target candidate w.r.t. its most voted opponent, akin to [WV18, CCDP19].

We define the MoV($S$) obtained starting from $S$ as the expected increase of the difference between the score of $c_*$ and that of the most voted opponent. Formally, if $c$ and $\hat{c}$ are respectively the candidates different from $c_*$ with the highest score before and after the LTM process, the MoV is

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

(4)

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.,

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

It is worth noting that MoV can also be expressed as a function of the score gained by candidate $c_*$ and the score lost by its most voted opponent $\hat{c}$ at the end of PLTR. 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 MoV($S$) as

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

(5)

3.5 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 in [WV18, CCDP19] it is sometimes convenient to increase the score of a third candidate in order to make the most voted opponent w.r.t. $c_*$ lose score and favor $c_*$. In the following we show that in our model the best strategy is the one that changes only the score of $c_*$. We distinguish between three possible strategies:

- **MoV$_1$**: Influencing voters about $c_*$. 
- **MoV$_2$**: Influencing voters about $\hat{c}$, i.e., the most voted opponent w.r.t. $c_*$ at the end of PLTR. 
- **MoV$_3$**: Influencing voters about any other candidate $c$.

\footnote{Like in previous papers, we prefer to use the increment in margin of victory rather than the margin itself to have well defined approximation ratios.}
Let us now analyze the MoV of $c_\star$ in these three different cases. As described in Equation (5), a general formulation for MoV is the following

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

$$= F(c_\star, S) - F(c_\star, \emptyset) + F(\hat{c}, \emptyset) - F(\hat{c}, S) + \Delta,$$

where $S$ is the initial set of seed nodes and $\Delta$ 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:

- **MoV$_1(S)$**
  $$\text{MoV}_1(S) = \sum_{v \in A} \frac{\pi(c_\star)}{1 + \sum_{u \in A \cap N_i^c} b_{uv}} - \sum_{v \in A} \pi(c_\star) + \sum_{v \in A} \pi(\hat{c}) - \sum_{v \in A} \frac{\pi(\hat{c})}{1 + \sum_{u \in A \cap N_i^c} b_{uv}} + \Delta$$

- **MoV$_2(S)$**
  $$\text{MoV}_2(S) = \sum_{v \in A} \frac{\pi(c_\star)}{1 + \sum_{u \in A \cap N_i^c} b_{uv}} - \sum_{v \in A} \pi(c_\star) + \sum_{v \in A} \pi(\hat{c}) - \sum_{v \in A} \frac{\pi(\hat{c})}{1 + \sum_{u \in A \cap N_i^c} b_{uv}} + \Delta$$

- **MoV$_3(S)$**
  $$\text{MoV}_3(S) = \sum_{v \in A} \frac{\pi(c_\star)}{1 + \sum_{u \in A \cap N_i^c} b_{uv}} - \sum_{v \in A} \pi(c_\star) + \sum_{v \in A} \pi(\hat{c}) - \sum_{v \in A} \frac{\pi(\hat{c})}{1 + \sum_{u \in A \cap N_i^c} b_{uv}} + \Delta$$

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 in PLTR it is always convenient to influence the voters about the target candidate $c_\star$ whenever you want to maximize the MoV of $c_\star$ and you can influence them only about one single candidate. Therefore, in the remainder of the paper we only focus at changing the score of the target candidate $c_\star$.

Note also that the observations above hold for both PLTR and R-PLTR models.
4 Hardness Results

In this section we provide two hardness results related to PLTR and R-PLTR. In Theorem 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). 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 within $n^{-f(n)}$ for any function $f \in o(1)$, under the Gap-ETH assumption [Man17].

In Theorem 2 we show that maximizing the MoV in R-PLTR is still $NP$-hard.

**Theorem 1.** An $\alpha\beta$-approximation to the election control problem in PLTR gives an $\alpha$-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 (D$k$S) 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 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, the score of $\hat{c}$ after PLTR starting from any initial set $S$ is

$$F(\hat{c}, S) = \sum_{v \in V} \tilde{\pi}_v(\hat{c}) = \sum_{v \in V \setminus A} \pi_v(\hat{c}) + \sum_{v \in A} \tilde{\pi}_v(\hat{c})$$

$$= |V| - |A| + \sum_{v \in A} \frac{1}{1 + \sum_{u \in A \cap N_v} \frac{1}{n\gamma}}$$

$$= |V| - \sum_{v \in A} \left( 1 - \frac{1}{1 + \sum_{u \in A \cap N_v} \frac{1}{n\gamma}} \right)$$

$$= |V| - \sum_{v \in A} \left( \frac{\sum_{u \in A \cap N_v} \frac{1}{n\gamma}}{1 + \sum_{u \in A \cap N_v} \frac{1}{n\gamma}} \right)$$

$$= |V| - F(c^\star, S),$$

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

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

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

$$P(G'_\emptyset) = \prod_{v \in V} \left(1 - \sum_{u \in N_v^i} b_{uv}\right) = \prod_{v \in V} \left(1 - \frac{|N_v^i|}{n^{\gamma}}\right) \geq \prod_{v \in V} \left(1 - \frac{1}{n^{\gamma - 1}}\right) = \left(1 - \frac{1}{n^{\gamma - 1}}\right)^n \geq \left(\frac{n}{1}\right) - \left(\frac{n}{1}\right) \frac{1}{n^{2(\gamma - 1)}} + \sum_{i=2}^{[n/2]} \left(\left(\frac{n}{i}\right) - \left(\frac{n}{i}\right) \frac{1}{n^{2i(\gamma - 1)}}\right) \geq 1 - \frac{1}{n^{\gamma - 2}}$$

where $(a)$ follows from the binomial expansion, $(b)$ is due to last negative term in the left hand side that does not appear in the right hand side when $n$ is even, and $(c)$ is due to

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

for any $\gamma \geq 2$. Since $P(G'_\emptyset) \leq 1$, then $P(G'_\emptyset) = \Theta(1)$. Moreover, $\sum_{G' \not= G'_\emptyset} P(G') = O\left(\frac{1}{n^{\gamma-2}}\right)$.

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

$$F_{G'}(c_*, S) = \sum_{v \in R_{G'}(S)} \pi_v(c_*) + \sum_{u \in R_{G'}(S) \cap N^i_v} \frac{1}{n^{\gamma}} = \Theta\left(\frac{1}{n^{\gamma}} \sum_{v \in R_{G'}(S)} |R_{G'}(S) \cap N^i_v|\right),$$

since $1 \leq \sum_{u \in R_{G'}(S) \cap N^i_v} \frac{1}{n^{\gamma}} \leq 2$ for each $v \in R_{G'}(S)$. Note that $\sum_{v \in R_{G'}(S)} |R_{G'}(S) \cap N^i_v|$ 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 $F_{G'}(c_*, S) = O\left(\frac{1}{n^{\gamma-2}}\right)$.

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

$$F_{G'_\emptyset}(c_*, S) = \frac{1}{n^{\gamma}} \sum_{v \in S} \frac{|S \cap N^i_v|}{1 + \sum_{u \in S \cap N^i_v} \frac{1}{n^{\gamma}}}$$
and since the denominator is, again, bounded by two constants we have that

\[ F_{G_0'}(c_\star, S) = \Theta \left( \sum_{v \in S} |S \cap N_v| \right) = \Theta \left( \frac{\text{SOL}_{DkS}(S)}{n^\gamma} \right), \]

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

\[ F(c_\star, S) = \sum_{G' \in \mathcal{G}} F_{G'}(c_\star, S) \cdot P(G') = F_{G_0'}(c_\star, S) \cdot P(G_0') + \sum_{G' \neq G_0'} F_{G'}(c_\star, S) \cdot P(G'). \]

Since \( F_{G'}(c_\star, S) \) and \( \sum_{G' \neq G_0'} P(G') \) are in \( O \left( \frac{1}{n^{\gamma - 2}} \right) \), then

\[ \sum_{G' \neq G_0'} F_{G'}(c_\star, S) \cdot P(G') = O \left( \frac{1}{n^{\gamma - 2}} \right) \sum_{G' \neq G_0'} P(G') = O \left( \frac{1}{n^{2(\gamma - 2)}} \right) = O \left( \frac{\text{SOL}_{DkS}(S)}{n^\gamma} \right), \]

for any \( \gamma \geq 4 \). Thus

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

which means that \( F(c_\star, S) = \Theta \left( \frac{\text{SOL}_{DkS}(S)}{n^\gamma} \right) \). We apply the Bachmann-Landau definition of \( \Theta \) notation: There exist three positive constants \( n_0, \beta_1, \) and \( \beta_2 \) 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 \( n_0, \beta_1, \) and \( \beta_2 \) 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 \( \text{OPT}_{DkS} \) is the value of an optimal solution for DkS.

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

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

Thus \( \text{SOL}_{DkS}(S) \geq \frac{\alpha}{\beta_2} \frac{\beta_1 \text{OPT}_{DkS}}{n^\gamma} \), i.e., the solution is an \( \alpha \beta \)-approximation to DkS, with \( \beta := \frac{\beta_1}{\beta_2} \). \( \square \)
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 \( I_{\text{LTM}} = (G, B) \) of Influence Maximization under LTM. \( I_{\text{LTM}} \) is defined by a weighted graph \( G = (V, E, w) \) with weight function \( w : E \rightarrow [0, 1] \) and by a budget \( B \). Let \( I_{R-\text{PLTR}} := (G', B) \) be the instance that corresponds to \( I_{\text{LTM}} \) 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 \( V' := V \cup \{v_{|V|+1}, \ldots, v_{2|V|}\} \).
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 \( E' := E \cup \{(v_1, v_{|V|+1}), \ldots, (v_{|V|}, v_{2|V|})\} \).
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 \( c_*, c_1, \ldots, c_{m-1} \). For each \( v \in V \) we set \( \pi_v(c_*) = 1 \) and \( \pi_v(c_i) = 0 \) for any other candidate \( i \in \{1, \ldots, m-1\} \). For each \( v \in V' \setminus V \) we set \( \pi_v(c_*) = 0, \pi_v(c_1) = 1 \) and \( \pi_v(c_i) = 0 \) for any other candidate \( i \in \{2, \ldots, m-1\} \).

Let \( S \) be the initial set of seed nodes of size \( B \) that maximizes \( I_{\text{LTM}} \) and let \( A \) be the set of active nodes at the end of the process. The value of the MoV obtained by \( S \) in \( I_{R-\text{PLTR}} \) is \( \text{MoV}(S) = |V| - |V \setminus A| \). Indeed, each node \( v \in V \) in \( G' \) has \( \tilde{\pi}_v(c_*) = \pi_v(c_*) = 1 \), 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 \( \tilde{\pi}_{v_{|V|+i}}(c_*) = (\pi_{v_{|V|+i}}(c_*) + 1)/2 = \frac{1}{2} \). Therefore, \( F(c_*, \emptyset) = F(c_1, \emptyset) = |V|, F(c_*, S) = |V| + \frac{1}{2}|A| \), 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 \( I_{R-\text{PLTR}} \). 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 \( I_{\text{LTM}} \), then, \( S \) would also not maximize \( I_{R-\text{PLTR}} \), which is a contradiction since \( S \) is an optimal solution for \( I_{R-\text{PLTR}} \).

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 \( c_* \), i.e., \( F(c_*, S) = |V| + \frac{1}{2}|A| \), is exactly equivalent to maximizing the cardinality of the active nodes in LTM. \( \square \)
5 Approximation Results

In this section we give a constant factor approximation algorithm for the election control problem in R-PLTR. We first 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 \( S^\ast \) and \( S^{\ast\ast} \) that maximize \( g^+(c^\star, S^\ast) \) and MoV(\( S^\ast \)) respectively, then MoV(\( S^\ast \)) \( \geq \frac{1}{3} \) MoV(\( S^{\ast\ast} \)). Indeed, we show a more general statement that is: If a solution \( S \) approximates \( g^+(c^\star, S^\ast) \) within a factor \( \alpha \), then MoV(\( S \)) \( \geq \alpha \frac{3}{3} \) MoV(\( S^{\ast\ast} \)).

Then we show that a simple greedy hill-climbing approach gives a constant factor approximation to the problem of maximizing \( g^+(c^\star, S^\ast) \), where the constant is \( \frac{1}{2} (1 - \frac{1}{e}) \).

By combining the two results, we obtain an \( \frac{1}{6} (1 - \frac{1}{e}) \)-approximation algorithm for the election control problem in R-PLTR.

5.1 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.

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 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.

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

**Proof.** Let us consider \( S \) and \( S^\ast \) as two solutions for the problem of maximizing the MoV for a target candidate \( c^\star \), with \( S^\ast \) 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. If we do not consider the gain given by the score lost by the most voted opponent and we assume there exists an \( \alpha \)-approximation to the problem of maximizing the increment in score of the target candidate, we have that

\[
\text{MoV}(S) = g^+(c^\star, S) + g^-(\hat{c}, S) - F(\hat{c}) + F(c) \\
\geq \alpha g^+(c^\star, S^\ast) - F(\hat{c}) + F(c) \\
\geq \frac{\alpha}{3} [g^+(c^\star, S^\ast) + g^-(\hat{c}, S^\ast) + g^-(\hat{c}, S^\ast)] - F(\hat{c}) + F(c),
\]

14
Algorithm 1 Greedy

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

where the last inequality holds because $g^+(c_*, S) \geq g^-(c_i, S)$ 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 $c_*$, 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

$$\text{MoV}(S) \geq \alpha \left[ g^+(c_*, S^*) + g^-(\hat{c}, S^*) + g^-(\hat{c}, S^*) - F(\hat{c}) + F(c) \right]$$

$$= \frac{\alpha}{3} \left[ g^+(c_*, S^*) + g^-(\hat{c}, S^*) + F(c) - F(\hat{c}) + g^-(\hat{c}, S^*) - F(\hat{c}) + F(c) \right]$$

$$= \frac{\alpha}{3} \left[ \text{MoV}(S^*) + g^-(\hat{c}, S^*) - F(\hat{c}) + F(c) \right],$$

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

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

$$\leq F(\hat{c}) - F(\hat{c}).$$

Thus, $g^-(\hat{c}, S^*) - F(\hat{c}) + F(c) \geq 0$ and we can conclude that

$$\text{MoV}(S) \geq \frac{\alpha}{3} \text{MoV}(S^*).$$

\[\square\]

5.2 Approximating the Score 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 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 [KKT15], associates to each node a non-negative weight $(w : V \rightarrow \mathbb{R}^+)$ 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

$$\arg \max_S \sigma_w(S) = \mathbb{E} \left[ \sum_{v \in A} w(v) \right].$$

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 [KKT15]. 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 such 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 $w(v) := \sum_{u \in N_v \setminus \mathbb{N}^u} b_{uv}(1 - \pi_v(c_\star))$. 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 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 [KKT15, Proposition 4.1].

Then, we can use a simple hill-climbing greedy to maximize the influence on $\hat{G}$. The hill-climbing algorithm starts with an empty set of seed nodes $S$ and adds to it, in each of $B$ rounds, the node $v$ with maximal marginal gain w.r.t. the solution computed so far.

**Theorem 4.** A greedy hill-climbing algorithm guarantees a $\frac{1}{6}(1 - \frac{1}{e})$-approximation factor to election control in R-PLTR.

**Proof.** We first prove that a simple hill-climbing greed algorithm achieves an $\frac{1}{6}(1 - \frac{1}{e})$ 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

$$g^+(c_\star, S) = F(c_\star, S) - F(c_\star)$$

$$= \sum_{v \in V} \pi_v(c_\star) + \sum_{u \in A \cap N_v^i} b_{uv} - \sum_{v \in V} \pi_v(c_\star)$$

$$= \sum_{v \in V} \frac{(1 - \pi_v(c_\star)) \sum_{u \in A \cap N_v^i} b_{uv}}{1 + \sum_{u \in A \cap N_v^i} b_{uv}}$$
and, since the denominator is at most 2, that

\[
g^+(c_\star, S) \geq \frac{1}{2} \sum_{v \in V} (1 - \pi_v(c_\star)) \sum_{u \in A \cap N_v^i} b_{uv} = \frac{1}{2} \sum_{u \in A} \sum_{v \in N_u^0} b_{uv} (1 - \pi_v(c_\star))
\]

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

Note that \(\sum_{u \in A} \sum_{v \in N_u^0} b_{uv} (1 - \pi_v(c_\star))\) is exactly the objective function that the greedy algorithm maximizes. Hence, using the result by Kempe et al. [KKT15], we know that

\[
\sum_{u \in A} \sum_{v \in N_u^0} b_{uv} (1 - \pi_v(c_\star)) \geq (1 - \frac{1}{e}) \sum_{u \in A^*} \sum_{v \in N_u^0} b_{uv} (1 - \pi_v(c_\star)),
\]

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

Therefore

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

since

\[
g^+(c_\star, S^\star) = \sum_{v \in V} \frac{(1 - \pi_v(c_\star)) \sum_{u \in A^* \cap N_v^i} b_{uv}}{1 + \sum_{u \in A^* \cap N_v^i} b_{uv}} \leq \sum_{v \in V} (1 - \pi_v(c_\star)) \sum_{u \in A^* \cap N_v^i} b_{uv} = \sum_{u \in A^*} \sum_{v \in N_u^0} b_{uv} (1 - \pi_v(c_\star)),
\]

where the inequality is due to the fact that the denominator in all the terms of \(g^+(c_\star, S^\star)\) is at least 1. Thus, the greedy algorithm 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 ratio for the MoV. \(\square\)

### 5.3 Destructive Election Control in R-PLTR

In this section we focus on the destructive election control problem. The model is similar to the constructive one, defined in Section 3. For each node \(v \in A\), where \(A\) is the set of active nodes at the end of LTM, the preference list \(\pi_v\) in the destructive case changes as follows:

\[
\tilde{\pi}_v(c_\star) = \frac{\pi_v(c_\star)}{1 + \sum_{u \in A^* \cap N_v^i} b_{uv}};
\]

\[
\tilde{\pi}_v(c_i) = \frac{\pi_v(c_i) + \frac{1}{m-1} \sum_{u \in A^* \cap N_v^i} b_{uv}}{1 + \sum_{u \in A^* \cap N_v^i} b_{uv}}, \text{ for each } c_i \neq c_\star.
\]
As for the MoV of $c_\star$, it is defined as
\[
\text{MoV}_D(S) := F(\hat{c}, S) - F(c_\star, S) - (F(c, \emptyset) - F(c_\star, \emptyset))
\]
\[
= F(c_\star, \emptyset) - F(c_\star, S) + F(\hat{c}, S) - F(\hat{c}, \emptyset) + \Delta,
\]
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. Equivalently, using the preference lists modified by the process, $\text{MoV}_D(S)$ can be also written as
\[
\text{MoV}_D(S) = \sum_{v \in A} \left( \frac{\pi_v(c_\star) \sum_{u \in A \cap N_v} b_{uv}}{1 + \sum_{u \in A \cap N_v} b_{uv}} \right).
\]

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 $w(v) := \sum_{u \in N_v} b_{uv} \pi_u(c_\star)$.

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

**Proof.** We first prove that a simple hill-climbing greed algorithm 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 $S^*$ 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
\[
g_D(c_\star, S) \geq \frac{1}{2} \sum_{v \in V} \left( \pi_v(c_\star) \sum_{u \in A \cap N_v} b_{uv} \right)
\]
\[
= \frac{1}{2} \sum_{u \in A} \sum_{v \in N_u} \pi_v(c_\star) \cdot b_{uv}.
\]

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
\[
\sum_{u \in A} \sum_{v \in N_u} b_{uv} \pi_v(c_\star) \geq \left(1 - \frac{1}{e}\right) \sum_{u \in A^*} \sum_{v \in N_u} b_{uv} \pi_v(c_\star),
\]
where \( A^* \) is the optimal set of active nodes, i.e., the set of active nodes at the end process starting from \( S^* \) (\( S^* \) the optimal solution for the weighted-LTM).

Therefore

\[
g_D^-(c_*, S) \geq \frac{1}{2} \left( 1 - \frac{1}{e} \right) g_D(c_*, S^*)
\]

because

\[
g_D^-(c_*, S^*) = \sum_{v \in V} \pi_v(c_*) \frac{\sum_{u \in A^* \cap N^i_v} b_{uv}}{1 + \sum_{u \in A^* \cap N^i_v} b_{uv}} \\
\leq \sum_{v \in V} \pi_v(c_*) \sum_{u \in A^* \cap N^i_v} b_{uv} = \sum_{u \in A^*} \sum_{v \in N^i_u} b_{uv} \pi_v(c_*),
\]

where the inequality is due to the fact that the denominator in all the terms of \( g_D(c_*, S^*) \) 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 \( c_* \) with the highest score before and after the solution \( S \) is applied; let \( \bar{c} \) be the most voted opponent after the optimal solution \( S^* \) is applied. Then we have that

\[
\text{MoV}(S) = g_D(c_*, S) + g_D^+(\hat{c}, S) + F(\hat{c}, \emptyset) - F(c, \emptyset)
\]

\[
\geq \left( 1 - \frac{1}{e} \right) g_D(c_*, S*) + g_D^+(\hat{c}, S) + F(\hat{c}, \emptyset) - F(c, \emptyset)
\]

\[
\geq \frac{1}{2} \left( 1 - \frac{1}{e} \right) \left[ g_D(c_*, S*) + g_D^+(\hat{c}, S^*) + g_D^+(\hat{c}, S) + F(\hat{c}, \emptyset) - F(c, \emptyset) \right]
\]

\[
= \frac{1}{2} \left( 1 - \frac{1}{e} \right) \left[ g_D(c_*, S*) + g_D^+(\hat{c}, S^*) + g_D^+(\hat{c}, S) + F(\hat{c}, \emptyset) - F(c, \emptyset) \right]
\]

\[
\geq \frac{1}{2} \left( 1 - \frac{1}{e} \right) \text{MoV}(S^*),
\]

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

\[\square\]

6 Simulations

We simulate our model on two real-world social networks\(^2\) on which political campaigning messages could spread:

\[^2\text{The datasets are taken from } \text{http://networkrepository.com/}\]
• **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” \( b_v \) 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 \( b_{uv} = \frac{1 - b_v}{|N^+_v|} \).

In our simulations we run our Greedy algorithm for the election control problem in R-PLTR. Then, we measure the score and the Margin of Victory (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 GREEDY to solve the standard Influence Maximization problem.

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 \( \times \) 32 KB 8-way L1 (data and inst) cache, and 4 \( \times \) 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 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 (see Figure 1, left column); 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.

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
Figure 3: The MoV calculated using the presented algorithm for polbooks (left) 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. Values greater than 0 are when our algorithm performs better than the simple Greedy for Influence Maximization.
our algorithm and the MoV calculated using the greedy algorithm for the Influence
Maximization problem. The simulations show that our algorithm outperforms the
standard Greedy algorithm, 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 our algorithm works
better than a simple Greedy 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 the simple Greedy algorithm just looks for influential voters,
independently from their initial opinion.

7 Conclusion

Influencing elections by means of social networks is a major issue in modern society
and understanding this phenomenon is of crucial importance in order to prevent
an attacker to control a large part of the votes. Our results constitute a first step
towards a realistic modeling of the use of social influence to control elections. We
proposed two new models that take into account the voter’s uncertainty or the fact
that they might partially hide their preference to a manipulator. In one model
the election control problem cannot be approximated within any reasonable bound,
under some computational complexity hypothesis. For the other model we provided
approximation algorithms that guarantee a constant approximation ratio both in the
constructive and in the destructive scenarios.

The results in this paper open several research directions. We plan to study the
approximability of the election control problem in a variant of R-PLTR in which
there are multiple campaigns that affects voters’ opinion on different candidates. It is
also worth to model voters’ uncertainty in different voting systems like, for example,
general scoring rules and study the approximation of the election control problem in
these cases. Finally, it would be interesting to consider uncertainty models also for
the diffusion process, for example, in robust influence maximization the weights on
the edges are not given in input but only their probability distribution is known.
References

[ACF+15] Vincenzo Auletta, Ioannis Caragiannis, Diodato Ferrioli, Clemente Galdi, and Giuseppe Persiano. Minority becomes majority in social networks. In 11th WINE, pages 74–88. Springer, 2015.

[AG17] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. J. Economic Perspectives, 31(2):211–36, 2017.

[BCE+16] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia. Handbook of Computational Social Choice. Cambridge University Press, New York, NY, USA, 1st edition, 2016.

[BE17] Robert Bredereck and Edith Elkind. Manipulating opinion diffusion in social networks. In 26th IJCAI, pages 894–900, 2017.

[BEEG16] Markus Brill, Edith Elkind, Ulle Endriss, and Umberto Grandi. Pairwise diffusion of preference rankings in social networks. In 25th IJCAI, pages 130–136, 2016.

[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. Nature, 489:295, Sep 2012.

[BFN16] Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Complexity of shift bribery in committee elections. In 30th AAAI, pages 2452–2458, 2016.

[BGP17] Sirin Botan, Umberto Grandi, and Laurent Perrussel. Propositionwise opinion diffusion with constraints. In 4th AAMAS Workshop (EXPLORE), 2017.

[BHN09] Nadja Betzler, Susanne Hemmann, and Rolf Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In 21st IJCAI, volume 9, pages 53–58, 2009.

[BRR11] Dorothea Baumeister, Magnus Roos, and Jörg Rothe. Computational complexity of two variants of the possible winner problem. In 10th AAMAS, pages 853–860, 2011.

[CCDP19] 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 18th AAMAS, 2019.
[CELM07] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. A short introduction to computational social choice. In *33rd SOFSEM*, pages 51–69, 2007.

[CLMM10] Yann Chevaleyre, Jérôme Lang, Nicolas Maudet, and Jérôme Monnot. Possible winners when new candidates are added: The case of scoring rules. In *24th AAAI*, 2010.

[CXX+18] Lin Chen, Lei Xu, Shouhui Xu, Zhimin Gao, and Weidong Shi. Election with bribed voter uncertainty: Hardness and approximation algorithm. *arXiv preprint arXiv:1811.03158*, 2018.

[CYZ10] Wei Chen, Yifei Yuan, and Li Zhang, Scalable influence maximization in social networks under the linear threshold model. In *10th ICDM*, pages 88–97, 2010.

[EFS09] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. Swap bribery. In *2nd SAGT*, pages 299–310, 2009.

[End17] Ulle Endriss. *Trends in Computational Social Choice*. Lulu.com, 2017.

[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 *27th IJCAI*, pages 219–225, 2018.

[FP10] Piotr Faliszewski and Ariel D Procaccia. Ai’s war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[FRM16] Piotr Faliszewski, Jörg Rothe, and Hervé Moulin. *Control and Bribery in Voting*, pages 146–168. Handbook of Computational Social Choice. Cambridge University Press, 2016.

[KKT15] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. *Theory of Computing*, 11(4):105–147, 2015.

[KL05] Kathrin Konczak and Jérôme Lang. Voting procedures with incomplete preferences. In *IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, volume 20, 2005.

[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.
[Man17] Pasin Manurangsi. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In 49th STOC, pages 954–961, 2017.

[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 42nd STOC, pages 755–764, 2010.

[SBLS18] 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 17th AAMAS, pages 265–273, 2018.

[XC11] Lirong Xia and Vincent Conitzer. Determining possible and necessary winners given partial orders. Journal of Artificial Intelligence Research, 41:25–67, 2011.