

Election Control in Social Networks via Edge Addition or Removal

Matteo Castiglioni,¹ Diodato Ferraioli,² Nicola Gatti¹

¹Politecnico di Milano, Piazza Leonardo da Vinci 32, Milano, Italy

²Università degli Studi di Salerno, Via Giovanni Paolo II, Fisciano, Italy
{matteo.castiglioni, nicola.gatti}@polimi.it, dferraioli@unisa.it

Abstract

We focus on the scenario in which messages *pro* and/or *against* one or multiple candidates are spread through a *social network* in order to affect the *votes* of the receivers. Several results are known in the literature when the manipulator can make seeding by buying influencers. In this paper, instead, we assume the set of influencers and their messages to be given, and we ask whether a manipulator (*e.g.*, the platform) can alter the outcome of the election by *adding* or *removing edges* in the social network. We study a wide range of cases distinguishing for the number of candidates or for the kind of messages spread over the network. We provide a *positive* result, showing that, except for trivial cases, manipulation is *not* affordable, the optimization problem being hard even if the manipulator has an unlimited budget (*i.e.*, he can add or remove as many edges as desired). Furthermore, we prove that our hardness results still hold in a *reoptimization* variant, where the manipulator already knows an optimal solution to the problem and needs to compute a new solution once a local modification occurs (*e.g.*, in bandit scenarios where estimations related to random variables change over time).

Introduction

Nowadays, social network media are the most used, if not the unique, sources of information. This indisputable fact turned out to influence most of our daily actions, and also to have severe effects on the political life of our countries. Indeed, in many of the recent political elections around the world, there has been evidence of the impact that false or incomplete news spread through these media influenced the electoral outcome. For example, in the recent US presidential election, Allcott and Gentzkow (2017) and Guess, Nyhan, and Reifler (2018) show that, on average, 92% of Americans remembered pro-Trump false news, while 23% of them remembered the pro-Clinton fake news. As another example, Ferrara (2017) shows that automated accounts in Twitter spread a considerable amount of political news in order to alter the outcome of 2017 French elections. Furthermore, Alaphilippe et al. (2018) and Giglietto et al. (2018) show that the fake news spread over the major social media

during the 2018 political campaign in Italy is linked with the content of the populist parties that won the elections.

In this scenario, a natural question is to understand at which extent the spread of (mis)information on social network media may alter the result of a political election. This topic has recently received large interest in the community: *e.g.*, Auletta et al. (2015; 2017a; 2017b) show that, in the case of two only candidates, a manipulator may be able to lead the minority to become a majority by influencing the order in which voters change their mind. Moreover, Auletta, Ferraioli, and Greco (2018) show that, in the same previous setting, a manipulator can lead a bare majority to consensus. However, as showed by Auletta et al. (2019), these results do not extend to the case with more than two candidates.

Arguably, the form of manipulation most studied in the literature is *seeding*, where the manipulator has to find a set of *seed* nodes from which the information (either positive towards the desired candidate or negative towards the opponents) spreads over the network. This form of manipulation has been largely studied in the setting with only two candidates, according to various diffusion models (Kempe, Kleinberg, and Tardos 2003; Bredereck and Elkind 2017). Only recently, the setting with multiple candidates has been investigated. In particular, Wilder and Vorobeychik (2018) study the manipulation of a plurality voting based election when the information spread in the network according to an Independent Cascade model; Corò et al. (2019) extend this result to a linear threshold diffusion model and arbitrary scoring rules; and Aboueimehrizi et al. (2019) allow a manipulator to have imperfect information on the network. However, all of these works still assume the manipulator only sends messages in favour or against a *single* candidate. This assumption is very restrictive and unrealistic when there are multiple candidates: for a manipulator, it may be advantageous to send messages not only in favour of her desired candidate, but also in favour of a third weaker candidate to make that the latter “steals” votes from the stronger opponents. However, Castiglioni et al. (2019) show that, when messages about multiple candidates are allowed, then the optimal seeding is usually hard even to approximate.

In all the above results, the network is given and the manipulator has no chance to alter the social relationships be-

tween voters. The question we pose in this paper is “what happens if the manipulator is the owner of the social network media?”. She may indefinitely conceal a message exchanged among two “friend” voters, or she may reveal information spread by unknown sources (*e.g.*, as sponsored content or through mechanisms as friend suggestions). That is, such a manipulator can remove or add edges in the network in order to push or obstruct the diffusion of information. Manipulations of this form have been already considered in literature: *e.g.*, Brederick and Elkind (2017) look at the case of two candidates and simple information diffusion dynamics, while Sina et al. (2015) and Auletta, Ferraioli, and Savarese (2019) study the effect of this form of manipulation in a setting where no information spread, but voters update their votes in an iterative voting process by effect of selfish voting.

Original Contributions. We focus on a manipulator able to add and/or remove edges in the social network when the information spread according to (a generalization of) the Independent Cascade model, arguably the most studied diffusion dynamics in the literature. We study both the case with only two candidates and the one with multiple candidates, and, in the latter case, we study both the simpler subcase in which the information on the network is all about a single candidate, and the more realistic subcase in which messages about multiple candidates can be spread. Basically, we show that, in any of these cases, the problem of deciding whether a set of edges to add and/or remove in the network exists so to make the desired candidate to win is hard. Surprisingly, these results hold even if the manipulator has an unlimited budget of edges to add or remove, except for the trivial setting of two candidates and a single message spreading over the network, in which the optimal solution with unlimited budget is to remove all edges, if the message is against the desired candidate, or to add all possible edges, otherwise. Actually, our results are even stronger. Indeed, we prove that it is hard to find a set of edges to add or remove that cause an increment in the margin of victory of the desired candidate that is a constant (and, in some case, even polynomial) approximation of the best possible increment that would be achieved by adding or removing edges.¹ Tables 1 and 2, reported in the following sections, summarize our results. Importantly, we remark that all our results still hold if one restricts the network to be acyclic.

Incidentally, in order to establish these results, we also provide new results for the basic Influence Optimization problem, that consists in maximizing or minimizing the number of nodes that receive the information spread over the network. We, indeed, originally prove that the minimization (maximization, respectively) variant of the problem cannot be approximated within any constant factor by removing

¹The proof of the hardness that the problem of deciding whether there is a strategy s.t. the probability that the desired candidate wins the election is larger than a given threshold follows from our inapproximability results. Indeed, we can pad each reduction with an opportune number of isolated nodes sending messages against the desired candidate, in order to make the latter to win the election only if a constant approximation exists before the padding.

(adding, respectively) a limited number of edges.

The hardness results presented in this work are a starting point for shaping the landscape of manipulability of election through social networks. This task is fundamental to understand when and how one must design interventions to reduce the severe effects of the spread of misinformation. Although our results are positive, showing that manipulation is not affordable in the worst case, we believe that the border of manipulability can be further sharpened. We here present a first study along this direction, looking at manipulators that face a *reoptimization problem* (Ausiello et al. 2012, Chap. 4) and thus answering the question “is manipulation easier if a solution to the problem for a given instance is already available and a local modification occurs?”. Note that this is very common in the real world, in which the social relationships among voters remain essentially stable between an election and the next one or when one is using bandit algorithms to learn the influence probabilities and estimations slightly change during time. Surprisingly, we show that all our hardness results are *robust* to the knowledge of solutions in similar settings, since they still hold in this reoptimization setting.

Model

Election Model. We study an election scenario modeled as follows. We have a set of candidates $C = \{c_0, c_1, \dots, c_\ell\}$ and a network of voters, represented as a weighted directed graph $G = (V, E, p)$, where V is the set of voters, E is the set of direct edges, and $p: V \times V \rightarrow [0, 1]$ denotes the strength of the potential influence among voters. In particular, for each edge (u, v) where $u, v \in V$, $p(u, v)$ returns the strength of the influence of u on v (more details on the influence model are discussed below). Each voter v assigns a *score*, by function $\pi_v: C \rightarrow \mathbb{N}$, to every candidate c_i . We assume function π_v to be injective, thus returning a different score to every candidate, formally, $\pi_v(i) \neq \pi_v(j) \forall c_i, c_j \in C$. The score $\pi_v(i)$ models how much voter v likes candidate c_i and induces, for voter v , a strict preference ordering over the candidates. Thus, $\pi_v(i) > \pi_v(j)$ models that voter v (strictly) prefers c_i to c_j . The election is based on *plurality voting*, where every voter casts a single vote for a single candidate, and the candidate that received the largest number of votes wins the election. We assume voters to be *myopic*, thus casting a vote for the candidate with highest score in their preference ordering. For each candidate $c \in C$, we denote with V_c the set of voters that rank c as first, formally, $V_c = \{v \in V \mid c = \operatorname{argmax}_{c_i \in C} \pi_v(i)\}$. Let $S \subseteq V$ be a subset of voters said *seeds*. Every seed s sends, for the sake of simplicity, at most one message per candidate (our proofs can be extended to the general case). We denote with $m_s = (q_0, \dots, q_\ell)$ the *message profile* of $s \in S$, where $q_i \in \{+, \cdot, -\}$, with $q_i = +$ ($q_i = -$, respectively) representing that s sends a *positive* (*negative*, respectively) message on c_i , and $q_i = \cdot$ representing that s does not send any message on c_i . A positive message on c_i increases the scores that voters assign to it, while a negative message does the reverse. Let $m_s(i) \in \{+, \cdot, -\}$ denote the message sent by s on c_i and $M = \cup_{s \in S} m_s$ denote the set of all the messages.

Diffusion Model. Given a pair seeds/messages (S, M) , messages are supposed to spread over the network according to a *Multi-Issue Independent Cascade* (MI-IC) model (Bharathi, Kempe, and Salek 2007).² Roughly speaking, in this model, each seed s propagates all messages in m_s to her neighbors. Then, a voter v with $v \notin S$, receiving messages from s , accepts the information that these messages carry with probability $p(s, v)$. If voter v accepts the messages, we say that v is *activated* by s . In her turn, each just activated voter v sends the received messages to her neighbors u , that will be activated with probability $p(v, u)$ if not activated in the past. After that, voter v becomes *inactive*. The process continues as long as there is some active voter and it is repeated for every different m_s independently. Formally, given graph $G = (V, E, p)$, we define the *live-graph* $H = (V, E')$, where each edge $(u, v) \in E$ is included in H with probability $p(u, v)$. Moreover, for every $s \in S$, we introduce a set $A_{m_s}^t \subseteq V$ composed of the *active* voters at time t due to message m_s . Every set $A_{m_s}^t$ is initialized with the seed sending the corresponding message for $t = 0$, i.e., $A_{m_s}^0 = \{s\}$, and the empty set for $t > 0$. At every time $t \geq 1$, set $A_{m_s}^t$ is defined as follows: for every edge $(u, v) \in E'$, we consider the set $\mathcal{M}(u, v) \subseteq M$ of messages m_s such that $u \in A_{m_s}^{t-1}$ —and thus u has just been activated by m_s —and $v \notin \bigcup_{i < t} A_{m_s}^i$ —and thus v has never been activated by m_s ; then for each (u, v) such that $\mathcal{M}(u, v)$ is not empty, we add v to $A_{m_s}^t$ for every $m_s \in \mathcal{M}(u, v)$. The diffusion process of message m_s terminates at time T_{m_s} when $A_{m_s}^{T_{m_s}} = \emptyset$. Finally, the cascade terminates when the diffusion of every message m_s terminates. A voter that activates at some t is said *influenced*.

Preference Revision. When a voter v accepts the messages received by a neighbor, her preferences can change. Denote with $R \subseteq M$ a set of received messages. A *ranking revision function* ϕ associates each pair (π, R) with a new ranking π' obtained by revising ranking π according to the set of received messages R . We use a *score-based* ranking revision function in which a positive (negative) message on a candidate c increases (decreases) her score by $\sigma(c) \geq 0$. Formally, each voter updates every candidate's score as follows:³

$$\pi_v(i) \leftarrow \pi_v(i) + \sigma(i) \cdot \left(\left| \{m_s \in R \mid m_s(i) = +\} \right| - \left| \{m_s \in R \mid m_s(i) = -\} \right| \right).$$

²Actually, the model by Bharathi, Kempe, and Salek (2007) focuses on a slightly different setting, namely viral marketing, in which alternative products, in place of messages, diffuse competitively over the network. Hence, our model differs from the one by Bharathi, Kempe, and Salek (2007) since a node can send multiple messages. Apart from that, we keep all the main features of the model by Bharathi, Kempe, and Salek (2007): it is based on independent cascade, and once a node is activated it ceases to receive messages in next rounds.

³It is easy to see that every hardness result related to this model keeps to hold even when we allow the same message to cause a different score increment (decrement) if received by different voters, or if forwarded by different neighbors, or if sent by different seeds.

According to our assumption on π_v , we require that, at the end of the diffusion process, no pair of candidates c_i, c_j has the same value of π_v . This can be obtained, e.g., by breaking ties according to some rule and slightly tilting scores so that they satisfy the tie-break outcome. In the rest of the paper, we assume, for simplicity, that $\sigma(i) = 1$ and we break ties in favor of the candidate ranked last before the diffusion process. This is equivalent to slightly perturbate the initial score with a multiplicative factor $(1 - \epsilon)$, where ϵ is a sufficiently small positive constant, e.g., $\epsilon = \frac{1}{1 + \max_{v,i} \pi_v(i)}$, and then apply the update rule. That is,

$$\pi_v(i) \leftarrow (1 - \epsilon)\pi_v(i) + \left(\left| \{m_s \in R \mid m_s(i) = +\} \right| - \left| \{m_s \in R \mid m_s(i) = -\} \right| \right).$$

Given a seed set S , a set M of messages, a set E of edges, and a live graph H , we denote as $\pi_v^*(i, S, M, E, H)$, for every voter $v \in V$, the score of candidate c_i at the end of the MI-IC diffusion (i.e., after the preference revision). Moreover, for each candidate $c \in C$, we denote with V_c^* the set of voters for which c is ranked as first after the preference revision, i.e., $V_c^*(S, M, E, H) = \{v \mid \operatorname{argmax}_{c_i} \pi_v^*(i, S, M, E, H) = c\}$. Finally, we suppose that the manipulator wants c_0 to win and we define the *margin of victory* of (S, M, H) as

$$\operatorname{MoV}(S, M, E, H) = \left| V_{c_0}^*(S, M, E, H) \right| - \max_{c \neq c_0} \left| V_c^*(S, M, E, H) \right|.$$

Election Control Problems. We study the problem of maximizing the *increase* of MoV when the manipulator can either *remove* or *add edges*. It is not hard to see that our hardness results extend also to the case where the manipulator can *both* remove and add edges.⁴

We next formally state the problems.

Definition 1 (ELECTION-CONTROL-BY-EDGE-REMOVAL (ECER)). Given an election scenario $(C, G, \{\pi_v\}, S, M)$ and budget $B \in \mathbb{N} \cup \{\infty\}$, the goal is finding $E' \subseteq E$ with $|E'| \leq B$ to remove from graph G to maximize $\mathbb{E}_H[\Delta_{\operatorname{MoV}}^-(E', H)]$, where $\Delta_{\operatorname{MoV}}^-(E', H) = \operatorname{MoV}(S, M, E \setminus E', H) - \operatorname{MoV}(S, M, E, H)$ is the increase of MoV due to the removal of edges E' .

Definition 2 (ELECTION-CONTROL-BY-EDGE-ADDITION (ECEA)). Given an election scenario $(C, G, \{\pi_v\}, S, M)$ and budget $B \in \mathbb{N} \cup \{\infty\}$, the goal is finding E' with $E' \cap E = \emptyset$ and $|E'| \leq B$ to add to graph G to maximize $\mathbb{E}_H[\Delta_{\operatorname{MoV}}^+(E', H)]$, where $\Delta_{\operatorname{MoV}}^+(E', H) = \operatorname{MoV}(S, M, E \cup E', H) - \operatorname{MoV}(S, M, E, H)$ is the increase of MoV due to the addition of edges E' .

In the following, when we refer to the *single-message case*, we are considering the setting in which there is a $c_i \in C$ and a $q_i \in \{+, -\}$ such that $m_s(j) \neq \cdot$ implies

⁴It reduces to the case where the manipulator can only remove edges setting $p(\cdot) = 0$ for all the added edges.

$j = i$ and $m_s(i) = q_i$, i.e., all the messages are referred to the same candidate and are either all positive or all negative. We use the term *multi-message case*, otherwise.

Influence Optimization. Incidentally, our analysis allows us to provide results also on the (more general) influence maximization/minimization problems when the manipulator can either *remove* or *add edges*. To formally describe these problems, we need to define function $\chi : S \times E \times H \rightarrow \mathbb{R}_+$ returning the number of influenced nodes with seeds S , edges E and live graph H . With abuse of notation, we also define $\chi(S, E) = \mathbb{E}_H[\chi(S, E, H)]$. Then we have the following two problems.

Definition 3 (INFLUENCE-MINIMIZATION-BY-EDGE-REMOVAL (IMER)). Given a setting (G, S, M) and budget $B \in \mathbb{N} \cup \{\infty\}$, the goal is finding a set $E' \subseteq E$ with $|E'| \leq B$ to remove from graph G to maximize $\Delta I^-(E') = \chi(S, E) - \chi(S, E \setminus E')$.

Definition 4 (INFLUENCE-MAXIMIZATION-BY-EDGE-ADDITION (IMEA)). Given a setting (G, S, M) and budget $B \in \mathbb{N} \cup \{\infty\}$, the goal is finding a set E' with $E' \cap E = \emptyset$ and $|E'| \leq B$ to add to graph G to maximize $\Delta I^+(E') = \chi(S, E \cup E') - \chi(S, E)$.

Example. Let us introduce a simple example to make the reader familiar with the above problems. Consider Figure 1, depicting the connections among five voters.

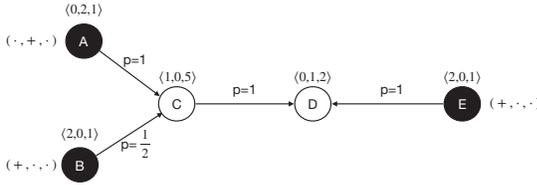


Figure 1: Example of an election with three candidates (c_0, c_1, c_2) . Black nodes represent seeds: node A sends a positive message on c_1 , while nodes B and E send a positive message on c_0 . The tuples $\langle \pi_v(0), \pi_v(1), \pi_v(2) \rangle$ above the nodes are the voters' preferences.

Two different live-graphs H_1 and H_2 are possible depending on whether or not B influences C. This happens with probability $\frac{1}{2}$.

In H_1 , B does not influence C and C receives only a positive message on c_1 , thus increasing the score of c_1 by 1. However, C has a very high evaluation of candidate c_2 and keeps to prefer c_2 over c_0 and c_1 . Instead, D updates her score to $\langle 1, 2 - \epsilon, 2 - 2\epsilon \rangle$ and will vote for c_1 . Thus, if B does not diffuse the message, at election time c_0 has 2 votes (B and E), c_1 has 2 votes (A and D) and c_2 has one votes (C). Hence, $\text{MoV}(S, M, E, H_1) = 2 - \max\{2, 1\} = 0$.

In H_2 , B influences C and C receives a positive message on c_0 and a positive message on c_1 . However, C keeps to prefer c_2 over c_0 and c_1 . Voter D receives a positive message on c_1 and two positive messages on c_0 , thus updating the scores to $\langle 2, 2 - \epsilon, 2 - 2\epsilon \rangle$ and then voting for c_0 . Hence, $\text{MoV}(S, M, E, H_2) = 3 - \max\{1, 1\} = 2$.

Results with Edge Removal

We study, in this section, the ECER and IMER problems. Our results on the ECER problem are summarized in Table 1 and show that, except for the trivial case with two candidates, unlimited budget, and a single message, the problem is hard even to approximate unless $P = NP$. In all our reductions, the edges have probability 1, unless we specify differently.

	single message		multiple messages,
	two candidates	three candidates	two or more candidates
limited budget	\notin APX (Cor 1)	\notin APX (Thm 2)	\notin Exp-APX (Thm 3)
unlimited budget	P (Obs 1)	\notin APX (Thm 2)	\notin Exp-APX (Thm 3)

Table 1: Complexity of election control by edge removal.

Initially, we focus on the IMER problem when one can only remove edges, as its characterization is useful for the characterization of the ECER problem with two candidates and limited budget. We show that the IMER problem is hard. Our proof reduces from the MAXIMUM-SUBSET-INTERSECTION problem that does not admit any constant-factor approximation polynomial-time algorithm unless $P = NP$, as showed by Shieh, Tsai, and Yang (2012).

Definition 5 (MAXIMUM-SUBSET-INTERSECTION (MSI)). Given a finite set $N = \{z_1, \dots, z_n\}$ of elements, a collection $X = \{x_1, \dots, x_m\}$ of sets with $x_i \subset N$, and a positive integer h , the goal is to find exactly h subsets x_{j_1}, \dots, x_{j_h} whose intersection size $|x_{j_1} \cap \dots \cap x_{j_h}|$ is maximum.

Theorem 1. For any constant $\rho > 0$, there is no polynomial-time algorithm returning a ρ -approximation to IMER problem when the budget B is finite, unless $P = NP$.

Proof. We reduce from MSI, showing that a constant-factor approximation algorithm for IMER implies the existence of a constant-factor approximation for MSI, thus having a contradiction unless $P = NP$. Given an instance (X, N) of MSI, we build an instance of IMER as follows. For each element z_i , we add $n^2 m^2$ nodes $v_{z_i, j}$ with $j \in \{1, \dots, n^2 m^2\}$. For each set $x_i \in X$, we add two nodes $v_{x_i, 1}, v_{x_i, 2}$ and an edge from $v_{x_i, 1}$ to $v_{x_i, 2}$. All $v_{x_i, 1}$ are seeds, while each $v_{x_i, 2}$ has an edge to each node $v_{z_i, j}, z_i \in N \setminus x_i$ with $j \in \{1, \dots, n^2 m^2\}$, i.e., all the nodes of all the elements not in the set x_i . Figure 2 depicts an example of network built with the above mapping. The budget is set equal to $m - h$.

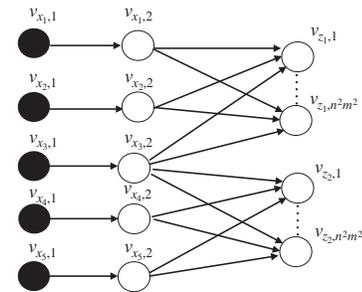


Figure 2: Graph used in the reduction of Theorem 1.

only edges from L_1 to v_{x_i} since, if it removes edges between nodes in L_1 , we would have $\Delta_{\text{MoV}}^-(E') \leq 0$. Moreover, all lines L_{z_i} must be active. Hence, the active vertexes v_{x_i} are a vertex cover and the inactive vertexes in v_{x_i} are an independent set. We remark that the value of $\Delta_{\text{MoV}}^-(E')$ is exactly the number of inactive vertexes, *i.e.*, the vertexes in the independent set. Thus, if there exists a ρ -approximation algorithm for ECER, there exists a ρ approximation algorithm for INDEPENDENT-SET, leading to a contradiction. \square

We now focus on the case in which there are multiple messages spreading in the network and only two candidates. We introduce the SET-COVER problem, that is well known to be hard, to prove the hardness of the ECER problem even with two candidates and multiple messages.

Definition 7 (SET-COVER). Given a set $N = \{z_1, \dots, z_n\}$ of n elements, a collection $X = x_1, \dots, x_m$ of sets with $x_i \subset N$, and a positive integer h , the objective is to select a collection $X^* \subset X$, $|X^*| \leq h$ with $\cup_{x_i \in X^*} x_i = N$.

Theorem 3. For any $\rho > 0$, there is no polynomial time algorithm returning a ρ -approximation to the ECER even with two candidates and unlimited budget, unless $\text{P} = \text{NP}$.

Proof. Consider an instance of SET-COVER. We suppose, w.l.o.g., $n > m$ and build a graph as follows. We add a node v_1 with preferences $\langle 1, 0 \rangle$ and seeded with messages $q_0 = +$ and $q_1 = -$, a node v_2 with preferences $\langle 1, 0 \rangle$ and seeded with message $q_0 = -$, and an edge between v_1 and v_2 . We add a line L_1 of $n^2 - h - 1$ nodes with preferences $\langle 1, 0 \rangle$ and an edge of probability $\frac{1}{2}$ from v_1 to the first node of the line and an edge from v_2 to the first node of the line. Moreover we seed the first node of the line with message $q_1 = +$. We add a node v_{x_i} for each set $x_i \in X$ with preferences $\langle 1, 0 \rangle$ and an edge from the last element of the line L_1 to v_{x_i} . For each element $z_i \in N$, we add a line L_{z_i} of n nodes with preference $\langle 0, 1 \rangle$ and an edge from each $x_j \in z_i$ to the first node of L_{z_i} . Moreover, we add $m - h + 1$ isolated nodes with preferences $\langle 0, 1 \rangle$. Figure 4 depicts an example of network produced with the above mapping. Note that, if no edge is removed, all the voters do not change their preferences and $\text{MoV} = 0$. We prove that Δ_{MoV}^- is larger than 0 if and only if SET-COVER is satisfiable.

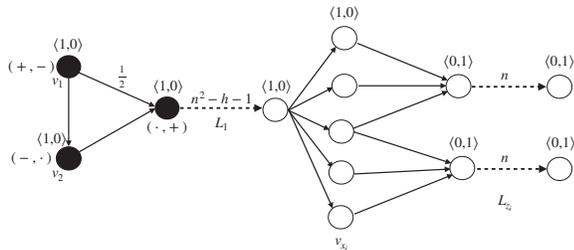


Figure 4: Graph used in the reduction of Theorem 3.

If. Define the set of removed edges E^* as composed by the edge between v_2 and L_1 and the incoming edge of each v_{x_i} with $x_i \in X \setminus X^*$. We have two possible live graphs: H_1 if the edge between v_1 and L_1 is active, H_2 otherwise. Thus,

$$\Delta_{\text{MoV}}^-(E^*, H_1) = 2n^2 \text{ and } \Delta_{\text{MoV}}^-(E^*, H_2) = 2(-n^2 + h + 1 - h) = -2n^2 + 2. \text{ Hence, } \Delta_{\text{MoV}}^-(E^*) = 1.$$

Only if. Suppose we do not remove neither the edge from v_1 towards v_2 nor the edge from v_2 towards L_1 . In this case, no voter changes her vote from c_1 to c_0 since all the nodes that votes for c_1 receive messages $q_0 = +$, $q_0 = -$, $q_1 = +$, and $q_1 = -$. Thus, one of the two aforementioned edges should be removed. It easy to see that removing the edge from v_2 to L_1 is the best choice. Since c_0 must take some of the votes of c_1 , the message in v_1 must reach at least some lines in L_z and no edges must be removed in L_1 . We have two possible live graphs: H_1 if the edge between v_1 and L_1 is active, H_2 otherwise. Assume by contradiction that in H_1 not all lines L_z vote for c_0 . This implies that $\Delta_{\text{MoV}}^-(E^*) \leq \frac{2(n(n-1)) - 2(n^2 + h + 1)}{2} < 0$, where E^* is the set of removed edges. Hence, in H_1 , all line L_z must be active and $\Delta_{\text{MoV}}^-(E^*, H_1) = 2n^2$. In H_2 , $\Delta_{\text{MoV}}^-(E^*)$ must be larger than $-2n^2 + 1$ and at most h nodes v_{z_i} can be active. Thus, there exists a set cover of size h . \square

Results with Edge Addition

We study, in this section, the ECEA and IMEA problems. Our results on the ECEA problem are summarized in Table 2. Basically, the complexity of the ECEA problem is the same as for ECER. Some proofs in this section are omitted. We refer the interested reader to the supplementary material.

	single message		multiple messages
	two candidates	three candidates	two or more candidates
limited budget	\notin APX (Cor 2)	\notin Exp-APX (Thm 5)	\notin Exp-APX (Thm 6)
unlimited budget	P (Obs 2)	\notin Exp-APX (Thm 5)	\notin Exp-APX (Thm 6)

Table 2: Complexity of election control by edge addition.

Initially, we study the complexity of IMEA problem with a finite budget. First, we notice that the APX-hardness of the IMEA problem directly follows from the APX-hardness of the influence maximization problem by seeding. In fact, the seeding problem with network $G(V, E, p)$ and budget B is equivalent to the add-addition problem with the same graph, except for an additional isolated node v_1 , that is the only seed, in which we can add at most B edges and the probabilities p of the new (added) edges are all zero except for the edges connecting v_1 to the nodes of V , whose probabilities p are one. We improve this result, showing that the IMEA problem is harder to approximate than influence maximization by seeding. Indeed, IMEA cannot be approximated to any constant factor, unless $\text{P} = \text{NP}$, while influence maximization by seeding can be. In our proof, we reduce from the maximization version of SET-COVER, called MAX-COVER.

Definition 8 (MAX-COVER). Given a finite set $N = \{z_1, \dots, z_n\}$ of elements, a collection $X = \{x_1, \dots, x_m\}$ of sets with $x_i \subset N$, and $h \in \mathbb{N}^+$, the objective is to select $X^* \subset X$, with $|X^*| \leq h$, that maximizes $|\cup_{x_i \in X^*} x_i|$.

Feige (1998) proves that deciding whether in an instance of MAX-COVER all the elements can be covered or at most a $(1 - \frac{1}{e} + \epsilon)$ fraction of them is NP-hard for any $\epsilon > 0$.

Theorem 4. For any constant $\rho > 0$, there is no polynomial time algorithm returning a ρ -approximation to the IMEA problem when B is finite, unless $P = NP$.

Proof. Consider an instance of MAX-COVER. We assume, w.l.o.g., $m < n$ and we build an instance of IMEA as follows: for each i in $\{1, \dots, n^8\}$, we add a node v_i , a node v_{i,x_j} for each $x_j \in X$ and a node v_{i,z_t} for each $z_t \in N$. Moreover, we add an edge from each v_{i,x_j} to each v_{i,z_t} , $z_t \in x_j$ with probability 1 and an edge from v_{i,z_t} to v_{i+1} with probability $1 - \frac{1}{n^{\frac{8}{h}}}$. We add a node v_{n^8+1} and an edge with probability 1 towards n^{10} nodes. Call the subgraph composed by these nodes G' . The resulting graph is depicted in Figure 5. The only seed is v_1 , and the only edges that can be added are the edges between v_i and v_{i,x_j} , $x_j \in X$ with probability one. The budget is $h n^8$. If MAX-COVER is satisfiable, i.e., there exists a set X^* that covers all the elements, there exists a solution E^* to IMEA in which for each $i \in \{1, \dots, n^8\}$ we add the edges from v_i to h v_{i,x_j} such that if v_i is active then all v_{i,z_t} are active. In this case, $\Delta I^+(E^*)$ is larger than the expected influence on the subgraph G' , i.e., $\Delta I^+ > [1 - (\frac{1}{n^{\frac{8}{h}}})^n] n^8 n^{10} \geq [1 - \frac{1}{n^8}] n^8 n^{10} > (\frac{1}{e} - \epsilon) n^{10}$ for all $\epsilon > 0$ and n large enough. Suppose each cover of size at most h cover at most $\frac{3}{4}$. It implies that at least $\frac{n^8}{h+1}$ nodes v_i have at most h outwards edges and thus they leave at least $\frac{1}{4}$ vertexes v_{i,z_t} without incoming edges. Thus the probability of activating G' is smaller than $[1 - (\frac{1}{n^{\frac{8}{h}}})^n]^{\frac{h n^8}{h+1}} [1 - (\frac{1}{n^{\frac{8}{h}}})^{\frac{3n^8}{4}}]^{\frac{n^8}{h+1}} \leq \frac{1}{e^{\frac{h}{h+1}}} \frac{1}{e^{\frac{n^8}{h+1}}} = \frac{1}{e^{\frac{h+n^8}{h+1}}} \leq e^{-n}$, and $\Delta I^+(E') \leq n^8(n+m+1) + 1 + e^{-n} n^{10}$. Clearly $\frac{\Delta I^+(E')}{\Delta I^+(E^*)} < \rho$, for each $\rho > 0$ and n sufficiently large. Hence, a ρ -approximation algorithm for IMEA implies that we can distinguish between satisfiable instances of MAX-COVER and instances in which at most $\frac{3}{4}$ of the elements are covered, leading to a contradiction. \square

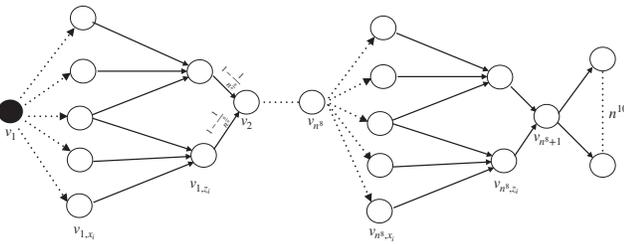


Figure 5: Graph used in the reduction of Theorem 4.

Now we can state the following result, whose proof is direct from the proof of the above theorem.

Corollary 2. For any constant $\rho > 0$, there is not any polynomial time algorithm returning a ρ -approximation to the ECEA when B is finite even with a single message and two candidates unless $P = NP$.

Thus, we focus on the case with unlimited budget. Since the maximum influence is reached when the network is fully

connected, the optimal solution to IMEA with unlimited budget adds all the non-existing edges to the network and thus can be computed in polynomial time. An argument similar to the one used for edge removal shows that ECEA with unlimited budget, two candidates, and a single message is easy. In particular, if the message is positive for c_0 , i.e., $q_0 = +$ or $q_1 = -$, we aim at maximizing the diffusion of the message and we add all the edges. If the message is negative for c_0 , i.e., $q_0 = -$ or $q_1 = +$, we aim at minimizing the diffusion and we do not remove any edge. From the previous arguments, we can directly state the following.

Observation 2. There exists a polynomial-time algorithm for the ECEA problem with unlimited budget, two candidates, and a single message.

Next, we prove that increasing the number of candidates or allowing multiple messages makes the problem hard⁵.

Theorem 5. For any $\rho > 0$, there is not any polynomial time algorithm returning a ρ -approximation to the ECEA even with three candidates, a single message, and unlimited budget, unless $P = NP$.

Theorem 6. For any $\rho > 0$, there is not any polynomial time algorithm returning a ρ -approximation to the ECEA even with two candidates and unlimited budget, unless $P = NP$.

Hardness of Reoptimization

We show that our hardness results are robust to a manipulator that knows a solution to a similar problem. Specifically, we consider the following reoptimization setting.

Definition 9. An election control reoptimization problem $ReOpt(I, E^*, e, o)$ is defined as follows.

- INPUT: (I, E^*, e, o) , where I is an instance of election control, E^* is an optimal solution to I , $e \in V \times V$ is an edge and $o \in [0, 1]$ is a probability.
- OUTPUT: the optimal solution to I_1 , where I_1 is obtained changing the probability of edge e to o in the instance I .

The following theorem shows a general result, that extends the hardness of the optimization problem to its reoptimization variant whenever a simple condition is satisfied.

Theorem 7. For the set of election control problems I with $\max_v \{\max_{c_i} \pi_v(i) - \min_{c_i} \pi_v(i)\} = O(\text{poly}(\text{size}(I)))$, reoptimization is as hard as optimization.

Proof. Consider an instance I of election control with $G = (V, E, p)$. By assumption $d = \max_v [\max_{c_i} \pi_v(i) - \min_{c_i} \pi_v(i)] = O(\text{poly}(\text{size}(I)))$, i.e., d is polynomially upper bounded in the instance size. We build a graph G_1 with $d + 1$ nodes $\{v_i\}, i \in \{0, \dots, d\}$ with seeds $q_0 = +$. We add a node v_1^* with an edge from each node in v_i to v_1^* . We add a node v_2^* with an edge from v_1^* to v_2^* . Moreover, we add an edge from v_2^* to any node of G . In edge addition instances, we set $p = 0$ for all (non-existing) edges among v_i and G . Finally, we set the preferences of v_0 and v_i s.t. they will vote for c_0 , i.e., $\pi(0) > \pi(i)$ holds for every $c_i \neq c_0$.

⁵The proof of the results are in (Castiglioni, Ferraioli, and Gatti 2019)

Notice that, since all nodes in G receive $d + 1$ positive messages on c_0 and c_0 is loosing by at most d in each preferences, all nodes will vote for c_0 . Thus the optimal solution removes/adds no edges. Consider the problem $ReOpt(I, \emptyset, (v_1^*, v_2^*), 0)$, its optimal solution is the optimal solution of the optimization problem over I . \square

In the reductions used in the proofs of all the theorems provided in the previous sections, $\max_v \{\max_{c_i} \pi_v(i) - \min_{c_i} \pi_v(i)\}$ is constant. Hence, as a corollary of Theorem 7, we have that all our hardness results on optimization problems extend to their reoptimization variants.

Acknowledgments

This work has been partially supported by the Italian MIUR PRIN 2017 Project ALGADIMAR “Algorithms, Games, and Digital Market”.

References

- Aboueimehrizi, M.; Corò, F.; Cruciani, E.; and D’Angelo, G. 2019. Election control with voters’ uncertainty: Hardness and approximation results. *CoRR* abs/1905.04694.
- Alaphilippe, A.; Ceccarelli, C.; Charlet, L.; and Mycielski, M. 2018. Disinformation detection system: 2018 italian elections. *Brussels: EU Disinfo Lab, June 1*.
- Allcott, H., and Gentzkow, M. 2017. Social media and fake news in the 2016 election. *Journal of economic perspectives* 31(2):211–36.
- Auletta, V.; Caragiannis, I.; Ferraioli, D.; Galdi, C.; and Persiano, G. 2015. Minority becomes majority in social networks. In *WINE*, 74–88.
- Auletta, V.; Caragiannis, I.; Ferraioli, D.; Galdi, C.; and Persiano, G. 2017a. Information retention in heterogeneous majority dynamics. In *WINE*, 30–43.
- Auletta, V.; Caragiannis, I.; Ferraioli, D.; Galdi, C.; and Persiano, G. 2017b. Robustness in discrete preference games. In *AAMAS*, 1314–1322.
- Auletta, V.; Ferraioli, D.; Fionda, V.; and Greco, G. 2019. Maximizing the spread of an opinion when tertium datur est. In *AAMAS*, 1207–1215.
- Auletta, V.; Ferraioli, D.; and Greco, G. 2018. Reasoning about consensus when opinions diffuse through majority dynamics. In *IJCAI*, 49–55.
- Auletta, V.; Ferraioli, D.; and Savarese, V. 2019. Manipulating an election in social networks through edge addition. In *AIIA*.
- Ausiello, G.; Crescenzi, P.; Gambosi, G.; Kann, V.; Marchetti Spaccamela, A.; and Protasi, M. 2012. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media.
- Bharathi, S.; Kempe, D.; and Salek, M. 2007. Competitive influence maximization in social networks. In *WINE*, 306–311.
- Bredereck, R., and Elkind, E. 2017. Manipulating opinion diffusion in social networks. In *IJCAI*.
- Castiglioni, M.; Ferraioli, D.; Landriani, G.; and Gatti, N. 2019. Election manipulation on social networks with messages on multiple candidates. *CoRR* abs/1902.03779.
- Castiglioni, M.; Ferraioli, D.; and Gatti, N. 2019. Election control in social networks via edge addition or removal. *arXiv preprint arXiv:1911.06198*.
- Corò, F.; Cruciani, E.; D’Angelo, G.; and Ponziani, S. 2019. Exploiting social influence to control elections based on scoring rules. In *IJCAI*, 201–207.
- Feige, U. 1998. A threshold of $\ln n$ for approximating set cover. *J. ACM* 45(4):634–652.
- Ferrara, E. 2017. Disinformation and social bot operations in the run up to the 2017 french presidential election. *First Monday* 22(8).
- Giglietto, F.; Iannelli, L.; Rossi, L.; Valeriani, A.; Righetti, N.; Carabini, F.; Marino, G.; Usai, S.; and Zurovac, E. 2018. Mapping italian news media political coverage in the lead-up to 2018 general election. *Available at SSRN 3179930*.
- Guess, A.; Nyhan, B.; and Reifler, J. 2018. Selective exposure to misinformation: Evidence from the consumption of fake news during the 2016 us presidential campaign. *European Research Council* 9.
- Kempe, D.; Kleinberg, J.; and Tardos, E. 2003. Maximizing the spread of influence through a social network. In *ACM SIGKDD*, 137–146.
- Shieh, M.-Z.; Tsai, S.-C.; and Yang, M.-C. 2012. On the inapproximability of maximum intersection problems. *Information Processing Letters* 112(19):723 – 727.
- Sina, S.; Hazon, N.; Hassidim, A.; and Kraus, S. 2015. Adapting the social network to affect elections. In *AAMAS*, 705–713.
- Wilder, B., and Vorobeychik, Y. 2018. Controlling elections through social influence. In *AAMAS*, 265–273.
- Zuckerman, D. 2007. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing* 3(6):103–128.