

Article

# Recommending Links to Control Elections via Social Influence

Federico Corò <sup>1,\*</sup>, Gianlorenzo D'Angelo <sup>1</sup> and Yllka Velaj <sup>2</sup>

<sup>1</sup> Gran Sasso Science Institute, 67100 L'Aquila, Italy

<sup>2</sup> ISI Foundation, 10126 Turin, Italy

\* Correspondence: federico.coro@gssi.it

Received: 15 July 2019; Accepted: 23 September 2019; Published: 1 October 2019



**Abstract:** Political parties recently learned that they must use social media campaigns along with advertising on traditional media to defeat their opponents. Before the campaign starts, it is important for a political party to establish and ensure its media presence, for example by enlarging their number of connections in the social network in order to assure a larger portion of users. Indeed, adding new connections between users increases the capabilities of a social network of spreading information, which in turn can increase the retention rate and the number of new voters. In this work, we address the problem of selecting a fixed-size set of new connections to be added to a subset of voters that, with their influence, will change the opinion of the network's users about a target candidate, maximizing its chances to win the election. We provide a constant factor approximation algorithm for this problem and we experimentally show that, with few new links and small computational time, our algorithm is able to maximize the chances to make the target candidate win the elections.

**Keywords:** computational social choice; election control; influence maximization; social networks; graph augmentation; approximation algorithms

## 1. Introduction

Over the past few years, political parties learned that, along with advertising on traditional mediums such as television and newspapers, they must use social media campaigns to defeat their opponents. Many real-life examples exist that show how the political intervention effect in social media [1–4]. One of the most effective examples of political digital marketing is the recent presidential US election where a study shows that 92% percent of people remembered the pro-Trump fake news, and 23% percent of them remembered the pro-Clinton fake news [5]. Another example, in 2017, where automated accounts in social networks spread a considerable portion of political-related content, mostly fake news, trying to influence the French election [2].

Before the campaign starts, it is important for a political party to establish and ensure its media presence. For example, and this is the case we study in this paper, they can enlarge their number of connections in the social network in order to assure a larger portion of users. Indeed, adding new connections between users increases the capabilities of a social network of spreading information which in turn can increase the retention rate and the number of new voters.

There is an extensive literature on manipulating elections without considering the underlying social network structure; we point the reader to a recent survey [6]. Although there have been few studies that exploits opinion diffusion on social networks in order to manipulate elections, we review them in Section 2.

In this work, we focus on a variant of the election control problem via social influence, introduced in [7]. We consider the scenario in which users of a social network have to vote for elections, i.e., each voter has a preference list over the set of candidates. Based on the model by Wilder et al. [7],

there is a subset of users who want to propagate a campaign, i.e., a message, among the users of the network with the aim of changing voters' opinions about a target candidate. Our goal is to study the case in which an external manipulator can decide to add new links in the network in order to change the outcome and make a specific candidate win.

### 1.1. Original Contribution

We study the problem of selecting a fixed-size set of new connections to a subset of voters who, with their influence, will change the opinion of the network's users about a target candidate, maximizing its chances to win the election. We provide a  $(1 - 1/e)$ -approximation algorithm for the problem of maximizing the score of a target candidate by showing submodularity, where  $e$  is the base of the natural logarithm. We exploit such property to provide a  $\frac{1}{3}(1 - 1/e)$ -approximation algorithm for the *constructive* election control problem. Finally, we present an experimental study on heterogeneous real-world networks to validate the model and assess the capability of the algorithm.

## 2. Related Work

In the literature, there are few studies that exploit social influence in order to solve the problem of constructive/destructive election control considering the social connections between users. This problem consists in changing voters' opinions with the aim of maximizing/minimizing the margin of victory of target candidate with respect to its most voted opponent.

Wilder et al. [7] considered the Independent Cascade Model [8] as diffusion process in both the constructive and destructive control scenarios providing algorithmic and hardness results, respectively, for maximizing the margin and the probability of victory.

Corò et al. studied the same problem under the Linear Threshold Model [8] and arbitrary scoring rules [9,10]. 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. Moreover, Faliszewski et al. [11] studied a variant of the Linear Threshold Model with weighted vertices; in their scenario, each node of the graph is a group 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. Bredereck et al. [12] instead focused 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  $1/2$ ; in this scenario, they studied how to manipulate, by means of bribing nodes or adding/deleting edges, the network in order to have control on the majority opinion, i.e., having more than 50% of nodes with the same opinion.

Recently, few studies considered the problem of modifying the network structure in order to minimize/maximize the information diffusion. D'Angelo et al. [13] introduced the *Influence Maximization with Augmentation* problem (IMA) that consists in adding a limited number of edges incident to a given set of nodes in order to maximize their capability of spreading information under the Independent Cascade Model. They proved that such problem is *NP*-hard to approximate within a factor greater than  $1 - 1/2e$  and provide an approximation algorithm that almost matches such upper bound. Furthermore, D'Angelo et al. [14] studied a similar problem for which the budget could be spent to buy either seeds or edges according to a cost function, providing two approximation algorithms that depend on the cost of the edges. Sheldon et al. [15] studied the problem of adding nodes in a network to maximize the diffusion in a network. They showed that their problem is not submodular and proposed exact integer programming formulations. Recently, Amelkin et al. [16] studied the problem of disabling opinion control in a social network by strategically altering the network's users' Eigen centralities by recommending new connections between the users. Zhang et al. [17] considered the problem of removing either edges or nodes from the network with the aim of minimizing the information diffusion and developed algorithms with rigorous performance guarantees and good empirical performance. Finally, experimental studies show that increasing the connectivity or the

centrality of a node, by adding edges to the graph, lead also to an increase in the expected number of nodes that the diffusion process is able to reach [18–20].

### 3. Preliminaries

#### 3.1. The Influence Maximization Problem

The *Influence Maximization Problem*, introduced by Kempe et al. [21] in 2003, studies a social network, represented by a weighted directed graph  $G = (V, E, p)$ , where the set of nodes  $V$  represent the users, the set of edges  $E$  represent social links between users, and the weight function  $p : V \times V \rightarrow [0, 1]$  represents the influence between users. The goal is then to find a set of influential nodes of size  $B$  that can maximize the spread of information. The influence is defined based on the *information diffusion process* among users.

There are many different models to study these processes on graphs, one of the most used being the *Independent Cascade Model* (ICM) [21]. Given a graph  $G = (V, E, p)$  and an initial set of nodes  $A_0$ , usually referred as *seeds*, the diffusion process proceeds iteratively in a synchronous way along a discrete time-axis, starting from  $A_0$ . Such set can be seen, for example, as the initial supporters of a product (or an opinion) that start to propagate it in a social network. Let  $A_t \subseteq V$  be the set of nodes active at time  $t$ , then, for each node  $u \in A_t \setminus A_{t-1}$ , a neighbor node  $v$  of  $u$  is activated at time  $t + 1$  with probability  $p_{uv}$ , independently, for each edge  $(u, v) \in E$ . Then,  $v \in A_{t+1}$  if and only if  $v \in A_t$  or one of the above independent events is realized. We say that 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  $A := A_{\tilde{t}}$ . In this case, the goal is to maximize the set  $A$ , i.e., the set of active nodes at the end of the process.

Kempe et al. showed that the distribution of the set of active nodes  $A$  in the graph starting with  $A_0$  under the ICM process is equivalent to the distribution of reachable nodes from the same set  $A_0$  in the set of random graphs called *live-edge graphs* (see Proof of Theorem 4.5 in [8]). A live-edge graph  $G' = (V, E')$  is built as follows: Given a graph  $G = (V, E, p)$ , every edge  $(u, v) \in E$  is selected to be inserted to  $E'$  independently at random with a probability proportional to its weight  $p_{uv}$ . This equivalent model allow them to provide two important results.

First, they conjectured that evaluating the expected number of active nodes under ICM is  $\#P$ -complete, which was later proved by Chen et al. [22]. However, it can be efficiently approximated by using a polynomial number of live-edge graphs ([8], Proposition 4.1). Moreover, under the live-edge model, the problem of selecting the initial seed set of nodes in order to maximize the diffusion is *submodular* (for a ground set  $N$ , a function  $z : 2^N \rightarrow \mathbb{R}$  is *submodular* if for any two sets  $S, T$  such that  $S \subseteq T \subseteq N$  and for any element  $e \in N \setminus T$  it holds that  $z(S \cup \{e\}) - z(S) \geq z(T \cup \{e\}) - z(T)$ ) [8]. Hence, exploiting a classical result [23], the influence maximization problem admits a  $1 - 1/e$  approximation using a simple greedy hill-climbing approach that, starting with an empty solution, for  $B$  iterations, selects the node that gives the maximal marginal gain on the objective function with respect the solution computed so far.

### 4. Problem Statement

Consider the scenario in which users of a social network have to vote for elections. In this work, we study the case in which an external manipulator can decide to add new links in the network in order to change the outcome of the elections and make a specific candidate win.

We represent the underlying connections between users as a directed graph  $G = (V, E)$ . Let  $C = \{c_1, \dots, c_m\}$  be the set of  $m$  candidates; we refer to our *target candidate*, i.e., the one that we want to make win the elections, as  $c_*$ . Each voter  $v \in V$  has a list of preferences for the elections represented as a function  $\pi_v : C \rightarrow \{1, \dots, m\}$  that represents the position of a given candidate in the preference list of  $v$ , e.g.,  $\pi_v(c_i) = 3$  for a given candidate  $c_i$  means that in the preference list of voter  $v$  such candidate is ranked as third.

In this work, we use the model presented by Wilder et al. [7]: We use the Independent Cascade Mode as a model for influence diffusion, i.e., each edge  $(u, v) \in E$  has an associated probability  $p_{uv}$ , that is the probability to influence the neighbor. Then, the process works as follow: Starting from the seed set, when a voter  $v$  is reached by the social influence, i.e., there exists a path from a seed to  $v$ , it shifts the ranking of the target candidate by one position up. If  $c_*$  is already in first position, i.e.,  $\pi_v(c_*) = 1$ , the message has no effect but the node still shares the message with its neighbors. At the end of the process, each voter  $v \in V$  casts a vote for their first ranked candidate, i.e., we consider plurality as voting system. Under *plurality rule*, each voter can only express a single preference among the candidates and that with the *plurality* of the votes wins, i.e., it is sufficient to have the highest number of votes and there is no need of an absolute majority (50% + 1 of votes).

Recall that in the problem of *election control* we want to maximize the chances of the target candidate to win the elections. To achieve that, we maximize its expected *Margin of Victory* (MOV) with respect to the most voted opponent, akin to that defined in [7]. Let us first define the set of voters that rank candidate  $c_i$  in the  $j$ th position as  $V_{c_i}^j = \{v \in V : \pi_v(j) = c_i\}$ . Note that initially candidate  $c_i$  has  $|V_{c_i}^1|$  votes. After the ICM process has quiesced, the position of  $c_*$  in the preference list of each influenced node is changed by one position up, except for the voters that already have  $c_*$  in the first position from the beginning.

Given the set of seed nodes  $A_0$  and the set  $S$  of edges added to  $G$ , we define the *score* of candidate  $c_i$  as  $F_{A_0}(c_i, S)$ . Initially, i.e., before the process starts, the score of a candidate  $c_i$  is equivalent to the number of voters that have  $c_i$  in first position, i.e.,  $F_{\emptyset}(c_i, \emptyset) = |V_{c_i}^1|$ . Note that, under this model, we can write the new score of  $c_*$ , i.e., the score after the process has quiesced, by the use of live-edge graphs. In fact, the score can be expressed by the number of voters that had  $c_*$  in first position before the process plus the expected number of reachable nodes in live-edge graphs that had  $c_*$  in second position, starting from the seed set  $A_0$ . Recall that a live-edge graph  $X = (V, E')$  is a subset of graph  $G$  where each edge  $(u, v) \in E$  is selected to be part of  $E'$  independently at random with a probability  $p_{uv}$ . Hence,  $X$  has probability to be sampled equal to  $\mathbf{P}(X) = \prod_{e \in E'} p_{uv} \prod_{e \in E \setminus E'} (1 - p_{uv})$ . Let us define  $R_{A_0}(X, V_{c_*}^2)$  as the set of reachable nodes in set  $V_{c_*}^2$  from the seed set  $A_0$  in live-edge graph  $X$  and  $\mathcal{G}(S)$  define the set of all possible live-edge graphs. Finally,

$$F_{A_0}(c_*, S) = \sum_{X \in \mathcal{G}(S)} \mathbf{P}(X) |R_{A_0}(X, V_{c_*}^2)| + |V_{c_*}^1|.$$

Finally, we define  $\text{MOV}_{A_0}(S)$  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 the candidates, different from  $c_*$ , with the highest score before and after the process has quiesced, respectively, then the MOV is defined as

$$\begin{aligned} \text{MOV}_{A_0}(S) &= F_{A_0}(c_*, S) - F_{A_0}(\hat{c}, S) - (F_{\emptyset}(c_*, \emptyset) - F_{\emptyset}(c, \emptyset)) \\ &= |V_c^1| - |V_{c_*}^1| - (F_{A_0}(\hat{c}, S) - F_{A_0}(c_*, S)). \end{aligned}$$

Let  $B \in \mathbb{N}$  be the initial budget, i.e., the maximum size of the set edges that the manipulator can add to  $G$ ; the *election control problem* asks to find a set of edges  $S \subseteq (A_0 \times V) \setminus E$ , of size at most  $B$ , that maximizes the MOV, i.e.,

$$\arg \max_{S \subseteq (A_0 \times V) \setminus E : |S| \leq B} \text{MOV}_{A_0}(S)$$

Roughly, we aim to maximize the expected number of votes by which  $c_*$  wins the election; however, note that not only candidate  $c_*$  gain a vote, but another candidate loses a vote and we need to keep track of the number of votes lost by each other candidate. The definition of MOV given above take into consideration this fact by comparing the votes gained by  $c_*$  and the number of votes of the candidate with the largest starting and smallest loss in votes.

As a running example, let us consider five voters  $v_1, \dots, v_5$  and three candidates  $C = \{c_*, c_2, c_3\}$ . Let three voters have  $p_{v_i}(c_2) = 1, p_{v_i}(c_3) = 2, p_{v_i}(c_*) = 3$ , for  $i \in 1, \dots, 3$  and two have  $p_{v_i}(c_2) = 1, p_{v_i}(c_*) = 2, p_{v_i}(c_3) = 3$ , for  $i = 4, 5$ . Since we are considering the plurality rule, we have that the scores of  $c_*, c_2, c_3$  following the elections are  $0, 5, 0$ .

Now, consider being able to influence voters 3 and 4 using the process presented by Wilder et al., which is the one we consider in this paper. Then,  $c_*$  is able to move one position up in  $\pi_{v_4}$  and  $\pi_{v_5}$  and the new scores are  $2, 3, 0$  and the value of is  $MOV = 5 - 0 - (3 - 2) = 4$ . We can make the following observations in this simple example: (1) even if we failed to make the target candidate win the elections, the value of is at its maximum value in this instance, i.e., even the optimal solution, would not have been able to make  $c_*$  win; and (2) since, under this model, we can only move the target candidate up to one position, influencing the first three voters would not have changed the outcome of the elections and the scores would have still been  $0, 5, 0$ .

### 5. Approximation Result

In this section, we first show that the change in score of the candidate  $c_*$  is a monotone and submodular function. This allows us to give a constant factor approximation using the result given in [24]. The authors proved that, if a solution approximates the change in score of the target candidate within a factor  $\alpha$ , then this solution provides a  $\alpha/3$  approximation to the MOV.

Let us first denote the change in score of candidate  $c_*$  as

$$g_{A_0}^+(c_*, S) = F_{A_0}(c_*, S) - F_{A_0}(c_*, \emptyset) = \sum_{X \in \mathcal{G}(S)} \mathbf{P}(X) |R_{A_0}(X, V_{c_*}^2)|$$

It directly follows that the function  $g_{A_0}^+(c_*, S)$  is a non-negative linear combination of functions  $R_{A_0}(X, V_{c_*}^2)$ . In the next lemma, we show that  $R_{A_0}(X, V_{c_*}^2)$  in live-edge  $X$  is a monotone submodular function of the initial set of nodes  $A_0$ . This allows us to prove that also  $g_{A_0}^+(c_*, S)$  is monotone and submodular with respect to  $A_0$  (Theorem 1).

**Lemma 1.** *Given a set of nodes  $A_0 \subseteq V$ , two live-edge graphs  $X, Y$  s.t.  $E_X \subseteq E_Y$  and  $E_Y \setminus E_X \subseteq A_0 \times V$ , and an edge  $(u, v) \in (A_0 \times V) \setminus (E \cup E_Y)$ . Then,*

$$|R_{A_0}(Y^+, V_{c_*}^2) \setminus R_{A_0}(Y, V_{c_*}^2)| \leq |R_{A_0}(X^+, V_{c_*}^2) \setminus R_{A_0}(X, V_{c_*}^2)|,$$

where  $X^+ = X \cup \{(u, v)\}$  and  $Y^+ = Y \cup \{(u, v)\}$ .

**Proof.** Note that we can decompose  $R_{A_0}(X^+, V_{c_*}^2) = |R_{A_0}(X, V_{c_*}^2)| + |R_{A_0}(\{(u, v)\}, V_{c_*}^2)| - |R_{A_0}(X \cap \{(u, v)\}, V_{c_*}^2)|$  and  $R_{A_0}(Y^+, V_{c_*}^2) = |R_{A_0}(Y, V_{c_*}^2)| + |R_{A_0}(\{(u, v)\}, V_{c_*}^2)| - |R_{A_0}(Y \cap \{(u, v)\}, V_{c_*}^2)|$ . Then, since  $E_X \subseteq E_Y$  and  $E_Y \setminus E_X \subseteq A \times V$  we have that  $R_{A_0}(X, V_{c_*}^2) \subseteq R_{A_0}(Y, V_{c_*}^2)$  and  $R_{A_0}(X \cap \{(u, v)\}, V_{c_*}^2) \subseteq R_{A_0}(Y \cap \{(u, v)\}, V_{c_*}^2)$ . Thus,

$$\begin{aligned} |R_{A_0}(Y^+, V_{c_*}^2)| - |R_{A_0}(Y, V_{c_*}^2)| &= |R_{A_0}(\{(u, v)\}, V_{c_*}^2)| - |R_{A_0}(Y \cap \{(u, v)\}, V_{c_*}^2)| \\ &\leq |R_{A_0}(\{(u, v)\}, V_{c_*}^2)| - |R_{A_0}(X \cap \{(u, v)\}, V_{c_*}^2)| = |R_{A_0}(X^+, V_{c_*}^2)| - |R_{A_0}(X, V_{c_*}^2)|. \quad \square \end{aligned}$$

We can now prove the following theorem.

**Theorem 1.** *Given a graph  $G = (V, E)$ , a seed set  $A_0$ ,  $g_{A_0}^+(c_*, S)$  is monotonically increasing submodular function of sets  $S \subseteq (A_0 \times V) \setminus E$ .*

**Proof.** First, we need to prove that  $g_{A_0}^+(c_*, S)$  is a monotonically increasing function, formally  $g_{A_0}^+(c_*, S \cup \{(u, v)\}) \geq g_{A_0}^+(c_*, S)$ , for each edge  $(u, v) \in (A_0 \times V) \setminus (E \cup S)$ .

$$\begin{aligned}
 g_{A_0}^+(c_*, S \cup \{(u, v)\}) - g_{A_0}^+(c_*, S) &= \\
 \sum_{X \in \mathcal{G}(S \cup e)} \mathbf{P}(X) |R_{A_0}(X, V_{c_*}^2)| - \sum_{X \in \mathcal{G}(S)} \mathbf{P}(X) |R_{A_0}(X, V_{c_*}^2)| &= \\
 \sum_{X \in \mathcal{G}(S)} \left( \mathbf{P}(X) p_{uv} |R_{A_0}(X^+, V_{c_*}^2)| + \mathbf{P}(X) (1 - p_{uv}) |R_{A_0}(X, V_{c_*}^2)| - \mathbf{P}(X) |R_{A_0}(X, V_{c_*}^2)| \right) &= \\
 \sum_{X \in \mathcal{G}(S)} \left( \mathbf{P}(X) p_{uv} |R_{A_0}(X^+, V_{c_*}^2)| - \mathbf{P}(X) p_{uv} |R_{A_0}(X, V_{c_*}^2)| \right) &= \\
 \sum_{X \in \mathcal{G}(S)} \mathbf{P}(X) p_{uv} \left( |R_{A_0}(X^+, V_{c_*}^2)| - |R_{A_0}(X, V_{c_*}^2)| \right) \geq 0. &
 \end{aligned}$$

The last equation holds since  $R_{A_0}(X, V_{c_*}^2) \subseteq R_{A_0}(X^+, V_{c_*}^2)$ .

To prove submodularity, we show that for each pair of sets  $S, T$  such that  $S \subseteq T \subset (A_0 \times V) \setminus E$  and, for each edge  $(u, v) \in (A_0 \times V) \setminus (E \cup T)$ , the increment in the expected number of influenced nodes that edge  $(u, v)$  causes in  $S \cup \{(u, v)\}$  is larger than the increment it produces in  $T \cup \{(u, v)\}$ , that is  $g_{A_0}^+(c_*, S \cup \{(u, v)\}) - g_{A_0}^+(c_*, S) \geq g_{A_0}^+(c_*, T \cup \{(u, v)\}) - g_{A_0}^+(c_*, T)$ .

Given two sets of edges  $S, T$ , such that  $S \subseteq T$ , for each live-edge graph  $X$  in  $\mathcal{G}(S)$  we denote by  $\mathcal{G}(T, X)$  the set of live-edge graphs in  $\mathcal{G}(T)$  that have  $X$  as a subgraph and possibly contain other edges in  $T \setminus S$ . In other words, a live-edge graph in  $\mathcal{G}(T, X)$  has been generated with the same outcomes as  $X$  on the coin flips in the edges of  $E \cup S$  and it has other outcomes for edges in  $T \setminus S$ .

$$\begin{aligned}
 g_{A_0}^+(c_*, T \cup \{(u, v)\}) - g_{A_0}^+(c_*, T) &= \\
 \sum_{Y \in \mathcal{G}(T)} \mathbf{P}(Y) p_{uv} \left( |R_{A_0}(Y^+, V_{c_*}^2)| - |R_{A_0}(Y, V_{c_*}^2)| \right) &= \\
 \sum_{X \in \mathcal{G}(S)} \sum_{Y \in \mathcal{G}(T, X)} \mathbf{P}(Y) p_{uv} \left( |R_{A_0}(Y^+, V_{c_*}^2)| - |R_{A_0}(Y, V_{c_*}^2)| \right). &
 \end{aligned}$$

Since, for each  $X \in \mathcal{G}(S)$ ,  $\sum_{Y \in \mathcal{G}(T, X)} \mathbf{P}(Y) = \mathbf{P}(X) \sum_{Y \in \mathcal{G}(T \setminus S, X)} \mathbf{P}(Y) = \mathbf{P}(X)$  and by Lemma 1, then

$$\begin{aligned}
 g_{A_0}^+(c_*, T \cup \{(u, v)\}) - g_{A_0}^+(c_*, T) &= \\
 \sum_{X \in \mathcal{G}(S)} \sum_{Y \in \mathcal{G}(T, X)} \mathbf{P}(Y) p_{uv} \left( |R_{A_0}(Y^+, V_{c_*}^2)| - |R_{A_0}(Y, V_{c_*}^2)| \right) &\leq \\
 \sum_{X \in \mathcal{G}(S)} \mathbf{P}(X) p_{uv} \left( |R_{A_0}(X^+, V_{c_*}^2)| - |R_{A_0}(X, V_{c_*}^2)| \right) &= \\
 g_{A_0}^+(c_*, S \cup \{(u, v)\}) - g_{A_0}^+(c_*, S). &\quad \square
 \end{aligned}$$

This result provide us with the fact that a simple greedy hill-climbing approach gives a constant factor approximation to the problem of maximizing  $g^+(c_*, A_0, S)$ , where the constant is  $\left(1 - \frac{1}{e} - \epsilon\right)$ . By combining this result and applying Theorem 3 in [24], here reported as Theorem 2, we obtain a  $\frac{1}{3} \left(1 - \frac{1}{e} - \epsilon\right)$ -approximation algorithm for the election control problem.

Theorem 2 holds for any scoring rule, i.e., plurality included, and for any model in which the manipulator has the ability to change only the position of a target candidate for a subsets of voters and the increment in score of such candidate is at least equal to the decrement in score of the other candidates. Our model satisfy all of these hypotheses since the manipulator is only able to modify the lists of the voters that are active during the ICM process, i.e., the nodes that are reachable from the seed set. Moreover, using the model from Wilder et al. [7], the manipulator can only change the score of the target candidate by increasing its position in the preference list up by one position. This also

implies that the increment in score of  $c_*$  is at least equal to the decrement in scoring of the other candidates since the new score of  $c_*$  can also be defined as the scores lost by the other candidates, i.e.,  $F_{A_0}(c_*, S) = |V_{c_*}^1| + \sum_{X \in \mathcal{G}} \mathbf{P}(X) \sum_{c \in C \setminus c_*} |R_{A_0}(X, V_{c_*}^2 \cap V_c^1)|$ .

**Theorem 2 ([24]).** *An  $\alpha$ -approximation algorithm for the problem of maximizing the increment in score of a target candidate gives an  $\frac{\alpha}{3}$ -approximation to the election control problem.*

**Corollary 1.** *There is an algorithm that outputs a solution  $S$  that satisfies*

$$\text{MOV}_{A_0}(S) \geq \frac{1}{3} \left( 1 - \frac{1}{e} - \epsilon \right) \text{MOV}_{A_0}(S^*),$$

where  $S^*$  denotes an optimal solution of size  $B$  to maximizing MOV.

## 6. Improving the Running Time

In the previous section, we proved that there exists an algorithm that approximate  $\text{MOV}_{A_0}(S)$  up to a constant factor by showing that the change in score of the target candidate  $c_*$  is monotone and submodular with respect to sets  $S \subseteq (A_0 \times V) \setminus E$  (Theorem 1). It follows that GREEDY (Algorithm 1) finds a set  $S$  of edges whose value  $g_{A_0}^+(c_*, S)$  is at least  $1 - 1/e$  times the one of an optimal solution [23].

---

### Algorithm 1: GREEDY

---

**Input** : Social graph  $G = (V, E, p)$ ; Seed set  $A$ ; Budget  $B$

**Output**: Set of edges  $S \subseteq \bar{E} \setminus E$  s.t.  $|S| = B$

1  $S := \emptyset$ ;

2 **for**  $i = 1, 2, \dots, B$  **do**

3      $\hat{e} = \arg \max \{g_{A_0}^+(c_*, S \cup \{(u, v)\}) \mid e = (a, v) \in \bar{E} \setminus (E \cup S)\}$ ;

4      $S := S \cup \{\hat{e}\}$ ;

5 **return**  $S$ ;

---

As it is stated, Algorithm 1 is clearly infeasible, in terms of running time, for very large real networks, having a computational complexity of  $\mathcal{O}(B \cdot |V| \cdot |E| \cdot R)$ , where  $R$  is the number of simulations, i.e., live-edge graphs, needed to compute an approximation for  $g_{A_0}^+(c_*, S)$ . Recall that calculating the expected number of active nodes is #P-complete but can be approximated by using a polynomial number of live-edge graphs [8,22].

Since our Algorithm 1 is a modified version of the algorithm GREEDY2 presented in [25], whose aim was to maximize the influence diffusion in a social network, we can exploit the same reduction to the *Limited Seed Selection problem* (LSS) that aims at finding a subset of the initial active users in order to maximize the expected number of influenced users. This reduction allows us to adapt the algorithm presented in [26] in such a way that it finds a subset of a given limited set of edges, i.e.,  $S \subseteq (A_0 \times V)$ .

## 7. Experimental Study

In this section, we present some experimental results that show how our approximation algorithm performs on real-world networks.

We chose several heterogeneous social and communication networks that are suitable for our problem, taken from KONECT [27], ArnetMiner [28] and SNAP [29] repositories. The size of the graphs are reported in Table 1.

We considered three different scenarios, each with a different number of candidates, i.e.,  $|C| = 2, 5, 10$ . For each scenario, we assigned a random preference list to each node of the networks.

Moreover, we chose 0.1% of the nodes in  $V$  as seeds and set the budget to  $B = 10 \cdot |A_0|$ . We generated the probabilities to the edges according to the weighted model presented by Kempe et al. [21], i.e., for each edge  $(u, v)$ , assign  $p_{uv} = 1/N_v^-$ , where  $N_v^-$  is the number of incoming neighbour of node  $v$ . Note that, for these experiments, the seed nodes were chosen uniformly at random.

**Table 1.** Real-world networks.

Name	V	E
Software Engineering (SE)	3141	14,787
Theoretical CS (TCS)	4172	14,272
High-Performance Comp. (HPC)	4869	35,036
Wiki-Vote (Wiki)	7115	103,689
Computer Graphic (CGM)	8336	41,925
Computer Networks (CN)	9420	53,003
Artificial Intelligence (AI)	27,617	268,460
Slashdot (Sl)	51,083	130,370
Epinions (Epi)	75,879	508,837
Slashdot-Zoo (Sl-z)	79,116	515,397
Twitter	465,017	834,797

All our experiments were performed on a computer equipped with two Intel Xeon E5-2643 CPUs (six cores clocked at 3.4 GHz) and 128 GB RAM; our programs were implemented in C++ (gcc compiler v4.8.2 with optimization level O3).

In Table 2, we report:  $\Delta(\emptyset)$  and  $\Delta(S)$  that are the differences in MOV of  $c_*$  with respect to the most voted opponent before and after the edge addition, i.e.,  $\Delta(\emptyset) = F_{A_0}(c_*, \emptyset) - F_{A_0}(c, \emptyset)$  and  $\Delta(S) = F_{A_0}(c_*, S) - F_{A_0}(\hat{c}, S)$ , where  $c$  and  $\hat{c}$  are the candidates, different from  $c_*$ , with the highest score before and after the process has quiesced;  $I$  is the relative increment of MOV after the edge addition computed as  $I = \frac{\Delta(S) - \Delta(\emptyset)}{\Delta(\emptyset)} \times 100$ ; and  $T$  is the time in seconds. Note that negative values in  $\Delta(\emptyset)$  means that the target candidate  $c_*$  was not winning before we used our algorithm.

**Table 2.** Results for real-world networks.  $\Delta(\emptyset) = F_{A_0}(c_*, \emptyset) - F_{A_0}(c, \emptyset)$  and  $\Delta(S) = F_{A_0}(c_*, S) - F_{A_0}(\hat{c}, S)$ , where  $c$  and  $\hat{c}$  are the candidates, different from  $c_*$ , with the highest score before and after the process has quiesced.

G	C  = 2				C  = 5				C  = 10			
	$\Delta(\emptyset)$	$\Delta(S)$	I	T	$\Delta(\emptyset)$	$\Delta(S)$	I	T	$\Delta(\emptyset)$	$\Delta(S)$	I	T
SE	6.65	330.88	4878.01%	0.70	23.03	106.79	363.76%	0.83	-26.31	13.02	149.48%	1.12
TCS	-57.61	305.47	630.22%	0.95	-15.87	74.41	568.99%	1.13	-22.23	34.05	253.16%	1.44
HPC	-32.31	470.10	1554.99%	1.16	-33.41	96.28	388.15%	1.26	-14.80	37.35	352.35%	1.87
Wiki	-32.39	680.64	2201.34%	3.68	-28.80	140.87	589.14%	4.03	-27.41	49.01	278.80%	5.80
CN	-38.08	1144.90	3106.47%	3.15	-57.77	243.39	521.32%	3.61	12.83	122.41	854.42%	5.04
CGM	-0.54	846.73	$157 \times 10^3\%$	2.43	-52.16	157.97	402.87%	2.57	-10.08	92.88	1021.20%	4.02
AI	221.07	3394.04	1435.27%	14.03	7.24	816.50	11,179.65%	14.62	10.78	362.16	3259.92%	21.22
Sl	150.59	2958.49	1864.54%	41.33	-16.43	750.20	4664.81%	48.02	54.87	493.46	799.25%	57.78
Sl-z	523.65	11,613.94	2117.86%	70.36	130.74	2941.96	2150.17%	81.61	12.48	1115.27	8837.80%	118.94
Epi	365.45	5700.39	1459.81%	80.61	47.68	1465.61	2974.00%	98.97	16.76	691.97	4027.51%	123.17
Twitter	1665.35	$363 \times 10^3$	$109 \times 10^3\%$	717.66	-238.7	$90 \times 10^3$	$190 \times 10^3\%$	790.26	166.5	37,739	$113 \times 10^3\%$	746.86

To prove the capability of our proposed algorithm, we compared the value of MOV in a graph augmented by using Algorithm 1 and in the same graph augmented by using several alternative baselines that connect the given seed set to a set of  $B$  nodes chosen accordingly: PrefAtt (PA) selects nodes according to the Preferential Attachment model [30,31]; Jaccard (J) selects the nodes with the highest Jaccard [32] coefficient; Degree (D) selects the nodes with the highest out-degree; Topk (TopK) selects the nodes with the highest harmonic centrality [33]; and Prob (Prob) selects nodes adding the edges with the highest probability. Note that some of these algorithms are well-known algorithms used for link recommendation. For each graph, we report in Tables 3 and 4 the relative increment of MOV, defined above as  $I$ . The experiments clearly show that our proposed algorithm outperforms all the alternative baselines with respect to the final value of MOV.

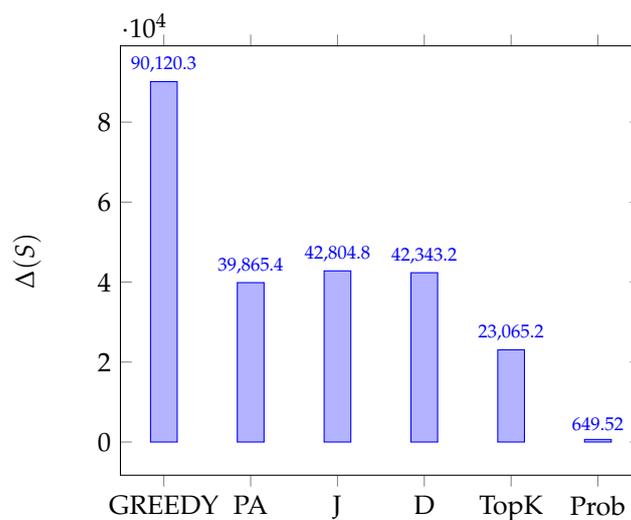
**Table 3.** Baseline results for real-world networks.

G	GREEDY			Pref. Attachment			Jaccard		
	2	5	10	2	5	10	2	5	10
C	2	5	10	2	5	10	2	5	10
SE	4878.01	363.76	149.48	895.00	62.25	25.25	757.50	42.97	24.79
TCS	630.22	568.99	253.16	154.79	136.00	51.79	110.12	90.67	23.43
HPC	1554.99	388.15	352.35	47.22	10.22	14.33	174.83	42.98	40.19
Wiki	2201.34	589.14	278.80	959.02	266.30	125.44	240.12	67.07	38.78
CN	3106.47	521.32	854.42	565.63	92.67	181.23	249.39	37.84	84.84
CGM	$157 \times 10^3$	402.87	1021.20	$18 \times 10^3$	48.84	107.16	$21 \times 10^3$	58.00	115.80
AI	1435.27	$12 \times 10^3$	3259.92	58.13	468.75	135.80	95.40	715.93	228.99
SI	1864.54	4664.81	799.25	302.27	680.22	91.69	292.41	673.30	87.54
SI-z	2117.86	2150.17	8837.80	758.46	760.91	3495.03	159.30	172.30	776.79
Epi	1459.81	2974.00	4027.51	371.26	707.52	905.31	222.93	447.97	541.20
Twitter	$109 \times 10^3$	$190 \times 10^3$	$113 \times 10^3$	10,423.88	16,858.20	11,472.56	9824.01	18,093.83	12,013.41

**Table 4.** Baseline results for real-world networks.

G	Degree			TopK			Prob		
	2	5	10	2	5	10	2	5	10
C	2	5	10	2	5	10	2	5	10
SE	1005.18	69.08	27.94	1852.49	135.54	54.45	1408.16	110.77	44.69
TCS	171.96	155.64	57.31	150.14	144.49	48.14	257.26	196.71	67.87
HPC	49.80	10.48	14.92	232.46	57.06	46.01	402.95	91.87	92.15
Wiki	1322.04	359.21	173.85	1336.46	370.66	180.50	304.50	65.13	36.57
CN	1058.30	165.02	349.46	1082.34	187.46	351.23	661.83	108.42	163.51
CGM	$27 \times 10^3$	73.03	158.09	$18 \times 10^3$	47.00	119.36	$44 \times 10^3$	118.67	280.47
AI	84.67	677.83	197.02	264.10	2067.03	625.05	349.74	2612.15	795.94
SI	306.53	697.86	93.54	336.11	757.73	102.47	569.15	1266.83	169.13
SI-z	778.30	778.93	3587.66	585.11	643.20	2946.73	366.46	365.02	1770.74
Epi	255.76	488.91	625.85	275.55	525.05	673.50	372.59	713.16	887.83
Twitter	8609.83	17,899.79	11,862.66	8497.19	9795.90	7198.48	208.64	-373.04	247.40

Figure 1 shows the effectiveness of the algorithm in the scenario with  $m = 5$  candidates running for the elections compared with the other approaches on the Twitter network. Note that this scenario is the “hardest” one among the considered ones, since it has the value of  $\Delta(\emptyset) = -238.7$ , i.e., the target candidate is not winning before the process starts, and Twitter is the largest network among the ones we considered.



**Figure 1.** Value of  $\Delta(S)$  in Twitter Network.

## 8. Conclusions

Using social media to influence elections is a major concern in our society and for this reason it is important to study this phenomenon in order to prevent attacks from manipulators. Our result might be a further step to understand and prevent these situations in real-life.

In this work, we address the problem of selecting a fixed-size set of new connections to a subset of voters that, with their influence, will change the opinion of the network's users about a target candidate, maximizing its chances to win the election. We provide a  $(1 - 1/e)$ -approximation algorithm for the problem of maximizing the score of a target candidate by showing submodularity. We then exploit such property to provide a  $\frac{1}{3}(1 - 1/e)$ -approximation algorithm for the *constructive* election control problem. Finally, we present an experimental study on several heterogeneous real-world networks to validate the model and assess the capability of the algorithm comparing our proposed solution with other link recommendation algorithms and showing its effectiveness.

Our results open several research directions. It could be interesting to study the same approach on different models, e.g., changing the model of diffusion, or using different voting rules, e.g., Borda rule that is used, as stated in Wikipedia, in elections by educational institutions or for granting sports awards in the United States. Moreover, future work should concentrate also in uncertainty models for the diffusion process, for example in robust influence maximization where weights on the edges are not known but a probability distribution is given in input.

**Author Contributions:** conceptualization, F.C., G.D. and Y.V.; methodology, F.C., G.D. and Y.V.; software, F.C., G.D. and Y.V.; validation, F.C., G.D. and Y.V.; formal analysis, F.C., G.D. and Y.V.; investigation, F.C., G.D. and Y.V.; data curation, Y.V.; writing-original draft preparation, F.C.; writing-review and editing, F.C., G.D. and Y.V.; funding acquisition, G.D.

**Funding:** This research was partially funded by the Italian MIURPRIN 2017 Project ALGADIMAR “Algorithms, Games, and Digital Markets”.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Bond, R.M.; Fariss, C.J.; Jones, J.J.; Kramer, A.D.I.; Marlow, C.; Settle, J.E.; Fowler, J.H. A 61-million-person experiment in social influence and political mobilization. *Nature* **2012**, *489*, 295. [[CrossRef](#)] [[PubMed](#)]
- Ferrara, E. Disinformation and social bot operations in the run up to the 2017 French presidential election. *First Monday* **2017**, *22*. [[CrossRef](#)]
- Kreiss, D. Seizing the moment: The presidential campaigns' use of Twitter during the 2012 electoral cycle. *New Media Soc.* **2016**, *18*, 1473–1490. [[CrossRef](#)]
- Stier, S.; Bleier, A.; Lietz, H.; Strohmaier, M. Election Campaigning on Social Media: Politicians, Audiences, and the Mediation of Political Communication on Facebook and Twitter. *Political Commun.* **2018**, *35*, 50–74. [[CrossRef](#)]
- Allcott, H.; Gentzkow, M. *Social Media and Fake News in the 2016 Election*; Working Paper 23089; National Bureau of Economic Research: Cambridge, MA, USA, 2017. [[CrossRef](#)]
- Faliszewski, P.; Rothe, J.; Moulin, H. Control and Bribery in Voting. In *Handbook of Computational Social Choice*; Cambridge University Press: Cambridge, UK, 2016; pp. 146–168. [[CrossRef](#)]
- Wilder, B.; Vorobeychik, Y. Controlling Elections through Social Influence. In Proceedings of the AAMAS 2018 17th International Conference on Autonomous Agents and MultiAgent Systems, Stockholm, Sweden, 10–15 July 2018; pp. 265–273.
- Kempe, D.; Kleinberg, J.M.; Tardos, É. Maximizing the Spread of Influence through a Social Network. *Theory Comput.* **2015**, *11*, 105–147. [[CrossRef](#)]
- Corò, F.; Cruciani, E.; D'Angelo, G.; Ponziani, S. Vote For Me!: Election Control via Social Influence in Arbitrary Scoring Rule Voting Systems. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems AAMAS '19, Montrea, QC, Canada, 13–17 May 2019; pp. 1895–1897.

10. Corò, F.; Cruciani, E.; D'Angelo, G.; Ponziani, S. Exploiting Social Influence to Control Elections Based on Scoring Rules. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI, Macao, China, 10–16 August 2019; pp. 201–207. [[CrossRef](#)]
11. Faliszewski, P.; Gonen, R.; Koutecký, M.; Talmon, N. Opinion Diffusion and Campaigning on Society Graphs. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18. International Joint Conferences on Artificial Intelligence Organization, Stockholm, Germany, 13–19 July 2018; pp. 219–225. [[CrossRef](#)]
12. Brederbeck, R.; Elkind, E. Manipulating Opinion Diffusion in Social Networks. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 894–900.
13. D'Angelo, G.; Severini, L.; Velaj, Y. Recommending links through influence maximization. *Theor. Comput. Sci.* **2019**, *764*, 30–41. [[CrossRef](#)]
14. D'Angelo, G.; Severini, L.; Velaj, Y. Selecting Nodes and Buying Links to Maximize the Information Diffusion in a Network. In Proceedings of the 42nd International Symposium on Mathematical Foundations of Computer Science, Aalborg, Denmark, 21–25 August 2017; pp. 75:1–75:14. [[CrossRef](#)]
15. Sheldon, D.; Dilkina, B.; Elmachtoub, A.N.; Finseth, R.; Sabharwal, A.; Conrad, J.; Gomes, C.; Shmoys, D.; Allen, W.; Amundsen, O.; et al. Maximizing the spread of cascades using network design. In Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence, Catalina Island, CA, USA, 8–11 July 2010.
16. Amelkin, V.; Singh, A.K. Fighting Opinion Control in Social Networks via Link Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Anchorage, AK, USA, 4–8 August 2019.
17. Zhang, Y.; Adiga, A.; Vullikanti, A.; Prakash, B.A. Controlling propagation at group scale on networks. In Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM), Atlantic City, NJ, USA, 14–17 November 2015.
18. Crescenzi, P.; D'angelo, G.; Severini, L.; Velaj, Y. Greedily improving our own closeness centrality in a network. *ACM Trans. Knowl. Discov. Data* **2016**, *11*, 9. [[CrossRef](#)]
19. Parotsidis, N.; Pitoura, E.; Tsaparas, P. Centrality-aware link recommendations. In Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, San Francisco, CA, USA, 22–25 February 2016.
20. Papagelis, M. Refining social graph connectivity via shortcut edge addition. *ACM Trans. Knowl. Discov. Data* **2015**, *10*, 12. [[CrossRef](#)]
21. Kempe, D.; Kleinberg, J.M.; Tardos, É. Maximizing the spread of influence through a social network. In Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 24–27 August 2003; pp. 137–146. [[CrossRef](#)]
22. Chen, W.; Wang, C.; Wang, Y. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 25–28 July 2010; pp. 1029–1038. [[CrossRef](#)]
23. Nemhauser, G.L.; Wolsey, L.A.; Fisher, M.L. An analysis of approximations for maximizing submodular set functions-I. *Math. Program.* **1978**, *14*, 265–294. [[CrossRef](#)]
24. Aboueimehrizi, M.; Corò, F.; Cruciani, E.; D'Angelo, G. Election Control with Voters' Uncertainty: Hardness and Approximation Results. *arXiv* **2019**, arXiv:1905.04694.
25. Corò, F.; D'Angelo, G.; Velaj, Y. Recommending Links to Maximize the Influence in Social Networks. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI, Macao, China, 10–16 August 2019; pp. 2195–2201. [[CrossRef](#)]
26. Cohen, E.; Delling, D.; Pajor, T.; Werneck, R.F. Sketch-based Influence Maximization and Computation: Scaling Up with Guarantees. In Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, Shanghai, China, 3–7 November 2014.
27. Kunegis, J. KONECT-The Koblenz Network Collection. In Proceedings of the International World Wide Web Conferen, Rio de Janeiro, Brazi, 13–17 May 2013.
28. Arnetminer. Available online: <http://arnetminer.org> (accessed on 15 January 2015).
29. Leskovec, J.; Krevl, A. SNAP Datasets: Stanford Large Network Dataset Collection. 2014. Available online: <http://snap.stanford.edu/data> (accessed on 28 September 2019).

30. Bollobás, B.; Borgs, C.; Chayes, J.; Riordan, O. Directed scale-free graphs. In Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, MD, USA, 12–14 January 2003.
31. Newman, M.E. Clustering and preferential attachment in growing networks. *Phys. Rev. E* **2001**, *64*, 025102. [[CrossRef](#)] [[PubMed](#)]
32. Jaccard, P. Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bull. Soc. Vaudoise Sci. Nat.* **1901**, *37*, 547–579.
33. Boldi, P.; Vigna, S. Axioms for centrality. *Internet Math.* **2014**, *10*, 222–262. [[CrossRef](#)]



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).