Heuristics for Opinion Diffusion via Local Elections

Rica Gonen, Martin Koutecký[1], Roei Menashof and Nimrod Talmon

Published at SOFSEM 2023, LNCS 13878, pp.144-158, 2023.

Abstract

Most research on influence maximization considers a simple diffusion model, in which binary information is being diffused (i.e., vertices – corresponding to agents – are either *active* or *passive*). Here we consider a more involved model of opinion diffusion: In our model, each vertex in the network has either approval-based or ordinal-based initial opinions and we consider diffusion processes in which each vertex is influenced by its neighborhood following a local election, according to certain "local" voting rules. We are interested in externally changing the preferences of certain vertices (i.e., campaigning) in order to influence the resulting election, whose winner is decided according to some "global" voting rule, operating after the diffusion converges. As the corresponding combinatorial problem is computationally intractable in general, and as we wish to incorporate probabilistic diffusion processes, we consider classic heuristics adapted to our setting: A greedy heuristic and a local search heuristic. We study their properties for plurality elections, approval elections, and ordinal elections, and evaluate their quality experimentally.

The bottom line of our experiments is that the Greedy and Simulated Annealing heuristics we propose perform reasonably well on both the real world and synthetic instances. Moreover, examining our results in detail also shows how the different parameters (ballot type, bribery type, graph structure, number of voters and candidates, etc.) influence the run time and quality of solutions. This knowledge can guide further research and applications.

1 Introduction

Social networks are ubiquitous in our lives and, as such, they have extensive influence on the public opinion in our society (see, e.g., [9]). In this paper we model a scenario in which an external agent wishes to change the public opinion; say, to have its preferred candidate win in an upcoming election (one such classical example is the 2016 US presidential elections [1]).

The situation we set out to study is complex as it consists of an interplay between several factors – a social network, opinions, an external agent, and the public opinion. As a result, our high level modeling contains the following ingredients:

A Social Network. There is a social network where each node initially possesses their own opinion. We model this naturally as a labeled graph, in which each node corresponds to a voter and is labeled by her opinion, and edges correspond to mutual influence of voters. Of course, there are many ways to formalize human opinions; as we are interested in a setting in which there is an upcoming election to be held, we model opinions as ballots. Importantly, we consider several ballot types, in particular, plurality ballots, approval ballots, and ordinal ballots.

A Bribing Agent. There is an external agent that can influence some voters and cause them to change their opinions. We model this through the well-established line of work considering campaigning or bribery in elections (see, e.g., [12, 11, 14]).

A Diffusion Process. There is a process by which information propagates through the network, so that some voters may further change their opinions as they are influenced by their neighbors. We model this through a probabilistic diffusion process in which, repeatedly, voters look at the opinions of their neighbors and may change their own opinion as a result (intuitively, if the opinions of their neighbors are significantly

[1]Partially supported by Charles University project UNCE/SCI/004 and by the project 22-22997S of GA ČR. Computational resources were supplied by the project "e-Infrastruktura CZ" (e-INFRA CZ LM2018140) supported by the Ministry of Education, Youth and Sports of the Czech Republic, and by the ELIXIR-CZ project (LM2018131), part of the international ELIXIR infrastructure.

different than their own opinion). Such processes are studied quite extensively (see, e.g., [18, 22]), however our modeling is different from some existing work and generalizes others: Technically, we introduce the concept of a *local voting rule* that builds upon the neighboring opinions of a voter and returns a score for each alternative, and use it to define a probability distribution for the altered opinion of the voter.

A Voting Rule. There is a mechanism that takes the eventual opinions of the voters and declares a winner of the election. Such mechanisms are usually referred to as voting rules, and are a fundamental structure studied in computational social choice [2, 10].

1.1 Our Contributions

Our first, conceptual contribution is our general model, that is able to capture the diffusion of complex opinions; we then realize our model with plurality ballots, approval ballots, and ordinal ballots. Moreover, as we use local voting rules in a stochastic way, our modeling is inherently stochastic; indeed, introducing further probabilistic diffusion processes to more complex kinds of opinions is a main motivation of our work.

We then take the point of view of the bribing agent and ask whether such an agent can efficiently find a bribery scheme that would maximize the chances of its preferred alternative winning the eventual election and maximize its winning gap. Not surprisingly, the corresponding combinatorial problem is computationally intractable in general. Thus, we describe several heuristic methods and evaluate their effectiveness experimentally. Technically, we consider both standard heuristic approaches (in particular, Simulated Annealing) and heuristics that proved to be effective for related tasks of opinion diffusion (in particular, greedy heuristics for Influence Maximization [20]).

1.2 Related Work

Our work fits naturally within the growing literature on opinion diffusion in social choice [17]. In particular, a recent paper [15] considers a similar setting that, while allowing some islands of computational tractability, differs in that the diffusion process is deterministic. Another paper [6] considers bribery and opinion diffusion for ordinal ballots, however their diffusion process is significantly different than ours and, in our opinion, somewhat problematic. This is because in their diffusion process, only the preferred candidate potentially moves up, but this means that the process depends on the point of view of the briber, while we hold that any definition of diffusion in the context of campaigning or bribery needs to be oblivious to the bribing agent(s). Other related works are the paper of Bredereck and Elkind [3] who consider a particular setting and approach it from a theoretical point of view. Wilder and Vorobeychik [24] consider a diffusion process related to the Linear Cascade model while we take an approach that is more in line with the Threshold Voter model.

More generally, our work relates to the study on Influence Maximization [20] (and hence, also to Target Set Selection [4]. These works usually deal with similar situations as we do, albeit in which the opinions are rather simple, usually binary. For this setting, there is a greedy heuristic [20] that was later improved [21, 5], which our greedy heuristics builds upon.

2 Formal Model

We describe the ingredients of our setting.

2.1 Opinion Graphs

We have a simple undirected graph $G = (V, E)$, where each vertex corresponds to a voter. The vertices of G are labeled by the ballot, such that the label of $v \in V$ corresponds to the vote of the agent v.

In the elections we consider there is always an underlying set of alternatives A. We consider several types of elections corresponding to the ballots a voter casts: plurality, approval, and ordinal. We identify a voter with their ballot, thus: In plurality elections, each voter $v \in V$ is some $v \in A$; in approval elections, each voter $v \in V$ is some $v \subseteq A$; and in ordinal elections, each voter $v \in V$ is some $v \in L(A)$, where $L(A)$ is the set of linear orders over A. (I.e., formally, V is a multisubset of A, 2^A and $L(A)$, for plurality, approval, and ordinal ballots, respectively.) Then, v is the label of the vertex corresponding to voter v, and such labeled graph is referred to as an *opinion graph*. Open neighborhood of a vertex v is defined as $N(v) = \{u \mid vu \in E\}$. Similarly, closed neighborhood of a vertex v is defined as $N[v] = N(v) \cup \{v\}$.

2.2 Bribery

We are interested in the problem of campaigning (also studied under the name bribery [11]) in our setting. Thus, we assume an external agent (i.e., the briber), who has a given budget, and can perform certain *bribery operations* on the voters, where a bribery operation operating on a certain voter $v \in V$ causes v

to change her vote.[2] We consider several bribery settings, differing by the cost of each possible bribery operation:

- In *simple bribery* [13], the briber pays one coin to change the vote of a voter v to any vote the briber wishes.
- In *approval bribery* [23], which is relevant only for approval elections, the briber pays one coin for adding an alternative to the approval set of a vote v or removing an alternative from the approval set of a vote v.
- In *swap bribery* [8], which is relevant for ordinal elections, the briber pays one coin for a single swap of two consecutive alternatives in the vote of a voter v. We focus on a restricted variant, *shift bribery* [7], in which the briber is only allowed to move the preferred candidate, and to only move them up.

In our setting we have an *initial* society graph, modeling the society **before** the briber bribes; then, the society might change following the bribery operations of the briber, into a society **after** the bribery.

Remark 1. *Technically we speak of bribery but conceptually our work relates to campaigning. One difference between the two is that in bribery one expects the bribed voter to stay loyal, but in campaigning, one attempts to influence the voter more indirectly and thus does not expect loyalty. Our model still allows for this definition of bribery by setting a high stubbornness parameter, defined below. That way, a bribed voter remains loyal, but also influences their peers.*

2.3 Diffusion Processes via Local Elections

We are interested in the propagation of opinions after the bribery happens. Specifically, we focus on synchronous diffusion, where in each step of the diffusion all voters might change their labels simultaneously. An asynchronous diffusion process, where in each step of the diffusion only one voter might change her label, can be defined analogously. The specific way by which voters might change their labels is governed by two parameters: A *local voting rule* $\mathcal{R}_L$, and a *stubbornness parameter* $\alpha \geq 0$.

The local elections voting rule $\mathcal{R}_L$ is a function that takes a certain collection of votes and returns a score for each alternative $a \in A$. It is used as follows: In each step of the diffusion, each vertex $v \in V$ applies $\mathcal{R}_L$ on a collection of votes obtained by taking the votes of all their neighbors, plus α-times their own vote v. We refer to the set of votes obtained by taking the open neighborhood of v, plus $\alpha \cdot d(v)$ copies of v's vote, as the *local election*. In particular, if $\alpha = 0$, then the local election of v consists of her open neighborhood (i.e., her opinion is disregarded); if $\alpha = 1$, then the local election consists precisely of the closed neighborhood; and if $\alpha = 5$, then the local election consists of all neighbors of v plus 5 copies of v.

The scores reported by the local rule $\mathcal{R}_L$ are used to define the probability distribution according to which v changes her opinion. The definitions are specific to the different voting rules, and are given below.

Remark 2. *The fact that the diffusion process is defined by the local election is a major extension of the existing models. Indeed, a main motivation for our work was to enrich existing models of opinion diffusion in social networks and push them closer to reality by considering various probabilistic processes.*

2.3.1 A Diffusion Process for Plurality Elections

Here we use $\mathcal{R}_L$ that returns the plurality score of each alternative. Then, we swap the voter's label to an alternative a with probability which is the score of a in the local election, divided by the number of votes in the local election.

Example 1. *Consider a voter v with open neighborhood $\{u_1, u_2\}$. Assume that v votes for (i.e., is labeled with) alternative c while u_1 and u_2 vote for alternative d. If $\alpha = 0$, then in the next timestep of a synchoronous diffusion process, v would surely change her vote to d. In contrast, if $\alpha = 1$ then v would change her vote to d with probability $2/3$, and with probability $1/3$ would still vote for c.*

2.3.2 A Diffusion Process for Approval elections

Here $\mathcal{R}_L$ is a function that returns the approval score of each alternative. Then, for each alternative $a \notin v$ (i.e., not currently approved by v), a is added to v with probability that equals the relative approval score of a (the *relative approval* score of an alternative is the fraction of voters approving the alternative); similarly, for each alternative $a \in v$ (i.e., currently approved by v), we remove a from v with probability that is one minus the relative approval score of a.

Example 2. *Consider again a voter v with open neighborhood $\{u_1, u_2\}$. Assume v votes for $\{a, b\}$ while u_1 and u_2 each votes for $\{b, c\}$. If $\alpha = 1$, then v would: definitely keep on approving b; with probability $2/3$ would also approve c; and with probability $2/3$ would cease approving a.*

[2]For simplicity, we assume bribery operations always succeed. A relaxation of this assumption is left for future work.

2.3.3 A Diffusion Process for Ordinal Elections

For ease of presentation, we describe our diffusion process for the case where $\mathcal{R}_L$ is the Borda rule; the description can be generalized to any ordinal voting rule that assigns scores to candidates (importantly, this includes also rules such as Copeland and STV, which can be defined as such).

We proceed as follows: Denote by $c_1, \ldots, c_m$ the candidates ordered by decreasing Borda scores in the local election centered at v; refer to this ordering as the *Borda-order* (in particular, the first candidate in the Borda-order is the Borda winner). The process is iterative, where in iteration i we consider c_i and do as follows: We look at position j of c_i in the ranking of v. Denote the ranking of v as $a_1, \ldots, a_m$; so, in particular, $a_j = c_i$ as c_i is the jth candidate in v's ranking. If $j = 1$ (i.e., if c_i is ranked first by v), then the iteration is complete. Otherwise, look at the Borda scores of a_j and of a_{j-1} (i.e., the candidate ranked by v just above c_i), and denote by $B(c)$ the Borda score of a candidate c. Now, define $x = \frac{B(a_j)}{B(a_{j-1})}$, with probability $\frac{x}{x+\frac{1}{x}}$, swap a_j and a_{j-1} (i.e., shift c_i one position up in v's ranking; otherwise (i.e., with the complement probability), the iteration is complete.

So, intuitively, we go over the candidates in decreasing Borda scores and we bubble-up each candidate with probability related to the Borda score of the candidate and the Borda score of the candidates in front of it in v's ranking. See full example in appendix A.

2.4 Election Results via Global Voting Rules

Intuitively, we wish to study the society after the bribery and after the diffusion process halts. However, as the diffusion process is probabilistic and is not guaranteed to halt, let us consider the expected society at infinity. Let the Markov chain of our process be a directed graph in which the starting node is the society after bribery, and each node corresponds to a possible society reached during the diffusion process; we have an arc from a node to another node with probability p if p is the probability of transitioning from one node to the other. Then, imagining an infinite random walk in this network, we wish to study the distribution of probabilities of where we end up, over all nodes. In particular, the *resulting election* is a probability distribution over the set of votes (i.e., labels) at infinity (wrt. the diffusion steps). In the simple case in which there is one absorbing node (i.e., a node with no outgoing edges), it means that the diffusion would halt on a specific society. Finally, a *global voting rule* $\mathcal{R}_G$ takes the society and returns a single alternative as the winner.

We consider a society stable from the perspective of our problem if the winner of the election is unlikely to change. We ran a sample of simulations for a large number of diffusion steps to determine a number k of steps after which the likelihood of a change of winner becomes reasonably small. Our finding is that after 20 diffusion steps, the proportion of instances in which the winner changes is at most 0.2%, and the trend is clearly decreasing. For figure illustrating see full version [16]. Because modeling a diffusion step is computationally expensive, we will assume from now on that, with respect to who wins the election, the society is close enough to the state of the Markov chain at infinity after 20 steps.

2.5 Optimization Goals

In general, we would like to understand the effect of different bribery actions on the resulting winner. Since the process is stochastic, we define two measures of success:

Definition 1 (PoW)**.** *Given a society after bribery and diffusion, the PoW (Probability of Winning) is the probability mass on the Markov chain nodes in which p wins.*

Definition 2 (MoV)**.** *Given a society after bribery and diffusion, the MoV (Margin of Victory) is the expected MoV of p, defined for a specific society as follows: If p wins, then the MoV is the difference between the score of p and the score of the runner-up (so, in particular, positive); if p loses, then the MoV is the difference between the score of p and the score of the winner (so, in particular, negative).*

To conclude, a specific model is characterized by:

1. A ballot type – Plurality, Approval, or Ordinal;
2. A bribery type – Simple bribery, Approval bribery, or Swap bribery;
3. A local voting rule $\mathcal{R}_L$ – Plurality, Approval, or Borda/Copeland;
4. A stubbornness parameter α;
5. A global voting rule $\mathcal{R}_G$ – Plurality, Approval, or Borda/Copeland;

For such models, we consider two computational problems, corresponding to optimizing either the PoW or the MoV. The input for both problems – referred to as Optimal-PoW and Optimal-MoV, respectively – contains an opinion graph G , preferred winner p and a budget of b coins; Optimal-PoW or Optimal-MoV asks for finding a bribery scheme costing at most b that maximizes the PoW or MoV for p, respectively.

3 Computing Optimal Bribery Schemes

Not surprisingly, the problems we set out to solve are NP-hard. In fact, even if there is no graph at all, our problems are intractable in general, since they reduce to bribery in elections [11] when there is no graph, and it is known, e.g., that bribery is hard for approval elections [12, Theorem 4.2]. Our setting is drastically more involved as we also consider a graph and a stochastic diffusion process operating on it.

3.1 Heuristic Methods

As our problems are generally intractable, our aim is to evaluate the possibility of efficiently solving them by heuristic methods. We report on computational simulations performed on their implementations. In particular, we use two heuristic algorithms. While there are indeed many other possibilities of heuristic algorithms one might consider, here we concentrate on two classic methods that proved to be useful in the setting of influence maximization (see below). Note that, interestingly, the heuristics we consider are, in a sense, oblivious to the specifics of the diffusion process considered; that is, their specific operation does not depend on the specifics of the problem we consider (e.g., the ballot type and other problem parameters).

Greedy. Our first heuristic approach is an adaptation of an algorithm considered for Influence Maximization [20] that works as follows: We iterate for b times where in each iteration we bribe the vertex that, if bribed, would increase the probability of p winning after the diffusion. Notice that computing the probability of p winning after the diffusion is a non-trivial sub-problem. In our simulations we handle this issue as follows: We perform 50 independent runs of the diffusion process using Monte Carlo, where in each run we perform 20 diffusion steps. Then we use the average over the 50 runs as an estimation of this probability.

Simulated Annealing. Our second heuristic approach is a local search algorithm that is an adaptation of an algorithm considered for Influence Maximization [19] that works as follows: With budget b, we start by selecting b vertices and bribe them, each by one coin (e.g., for plurality elections this corresponds to selecting an initial solution uniformly at random). Then, we estimate the probability of p winning after the diffusion, again using 50 iterations of Monte Carlo. Then, in each iteration of local improvement, we select one of the currently-bribed voters and one of her neighbors, and instead of bribing her, we bribe the neighbor. If this small change to the current solution increases the probability of p winning (as estimated by our Monte Carlo repetitions), then we keep this local improvement; otherwise, with increasing probability, we reject it and consider a different local improvement.

4 Simulations

We implemented the heuristics described above and evaluated them in various settings. The main goal of the simulations was to understand the possibility of computing optimal briberies in practice and to better identify the problem parameters that make finding optimal bribery schemes hard. Below we describe our experimental design for plurality elections, approval elections, and ordinal elections.

4.1 Experimental Design

First, let us describe our input graphs. We use both synthetic and real world data.

Synthetic graphs. We use the following models:

- $G(n, p)$ – We generate input graphs from $G(n, p)$, as follows: For a given number n of vertices and a value $0 \leq p \leq 1$, we first create n independent vertices. Then, for each pair of vertices, independently and uniformly at random, we flip a coin and with probability p put an edge between them. After we generate the $G(n, p)$ graph as just described, we assign labels to its vertices; we do so uniformly at random (thus, effectively, the resulting election behaves according to the Impartial Culture model).
- *k-k-clusters* – We create k subgraphs, each a $G(n/k, p_1)$ graph with some p_1. Then, for each pair of vertices u, v which are from different subgraphs, we put an edge with probability $p_2 < p_1$, independently and uniformly at random. This model creates the graph together with the labels as follows:
  - For plurality elections – The label of the vertices in the jth subgraph ($j \in [k]$) is j.
  - For approval elections – Make k random ballots and label them with $b_1, \ldots, b_k$. Denote b_j the cluster V_j's "base ballot". Set $0 < \alpha < 1$. Label each vertex V_j with an α-pertubation of b_j, where α-pertubation of a ballot b is defined as follows. Take each candidate approved in ballot b and make it a not-approval with probability α, and take each candidate not approved in ballot b and make it an approval with probability α.

– For ordinal elections – Make k random ballots and label them with $b_1, \ldots, b_k$. Donate b_j the cluster V_j's "base ballot" Just like in approval elections. Set $0 < \alpha < 1$. Label each vertex V_j with an α-pertubation of b_j. α-pertubation of a ballot $b = b_1, b_2, \ldots$ defined as follows. Begin with a blank ranking r (this is the ballot we are creating). For i from 1 to m: insert b_i into r at position $j \leq i$ with probability $\alpha^{i-j}/(1 + \alpha + \ldots + \alpha^{i-1})$.

Intuitively, while the $G(n, p)$ model creates uniform graphs, the k-k-clusters model creates random graphs that aim at mimicking communities.

Real-world Graphs. We use graphs from the *email-Eu-core* network[3], referred to below as *real-world network graph*. Vertices in this graph correspond to real people, and there is a directed edge from one vertex to another if the person corresponding to the head of the directed edge sent at least one email to the person corresponding to the tail of the directed edge.

4.1.1 Metrics.

We evaluate the heuristic algorithms described above by estimating the PoW and the MoV (see Definitions 1 and 2). To estimate PoW (MoV) we perform 50 Monte Carlo iterations to estimate the probability that p wins (the expected margin of victory of p) after the bribery operations performed by the heuristic algorithm and after 20 iterations of the diffusion process (recall 2.4 and the discussion above of why 20 iterations are a reasonable proxy to the stable state). Recall that the higher the PoW (MoV) the better. Furthermore, we report on the running times of our heuristics.

4.1.2 Model settings

summary of the various inputs for the model, as mentioned above, with the parameter settings:

- Voting rule – Plurality, Approval, Borda or Copeland.
- Graph type – $G(n, p)$, 5-k-clusters or Real-world Graphs.
- Budget – Supposedly, each coin can bribe a single node, there are b coins to use for bribery. We experimented with amounts between 5 and 50.
- Number of candidates – The number of candidates that are running for election is also expressed as m. We experimented with leaps of 5 between range of 5 to 50.
- Number of voters – The number of voters, also expressed as n. For $G(n, p)$ and 5-k-clusters we experimented with leaps of 100 between range of 100 to 1500 and also 1005, the values are fixed on 1005 nodes for Real-world Graphs.

4.2 Results

Our main results are threefold:

- When **plurality election** is used in the diffusion process and MoV is optimized at the cost of a slower run time, then Simulated Annealing performs the best.
- When **approval election** is used in the diffusion process and the available budget is relatively high then again Simulated Annealing performs the best with regards to all studied objectives, namely, it achieves higher PoW, higher MoV, and shorter run time than Greedy.
- When **ordinal election** is used in the diffusion process then Greedy and Simulated Annealing perform about the same though Greedy takes longer time to run. A significant factor in the run time is the choice of an ordinal rule, since computations for Borda are easier than computations for Copeland and indeed runs with Copeland as the ordinal rule tend to last ten times longer than runs with Borda as the ordinal rule. However, for Copeland, significantly higher MoV in the real world network graph and slightly higher PoW in both synthetic and real world network graphs is achieved.

More specifically, we compare the Greedy heuristic method performance with that of Simulated Annealing on the different optimization goal and the different diffusion processes (see full version [16] for additional details).

- In plurality and approval elections, the Simulated Annealing heuristic achieve better MoV and PoW than the Greedy heuristic as the **budget** increases, in particular with real world graph (See Figure 2). For ordinal elections, however, we were mostly unable to find a correlation.
- The impact of the number of **candidates and voters** on the Greedy and Simulated Annealing heuristics is not significant.

[3] http://snap.stanford.edu/data/email-Eu-core.html

- With high budgets, the Simulated Annealing heuristic performs better in terms of the **PoW run time**, because it finds the maximum point quickly.
- The greedy heuristic performs and scales better in terms of the **MoV run time**. The Simulated Annealing heuristic took longer to execute in most cases, and there was not always a clear correlation (see Figure 2).

Several more in-depth observations of the impact of the various parameters in our analysis are:

1. We found that the PoW and the MoV performance of both the Greedy and the Simulated Annealing heuristics are positively correlated with the budget increase. This makes intuitive sense: the more money, the better the outcome. This observation is true for the diffusion processes of plurality, approval, and Borda in the real world graph (slightly) and Copeland in $G(n,p)$ and the real world graph. However we could not find correlation for Borda in $G(n,p)$ and the 5-clusters graphs and Copeland in 5-clusters graph. We can see that even with a small number of candidates, there is a correlation to the budget; however, as the number of candidates grows, the budget becomes irrelevant. The intuitive reason is the implementations of the ordinal bribe and diffusion operations; the bribe increases the candidate's rank by one place only, while the diffusion favors high-ranking candidates, so the diffusion process may often change the ranking back down (see full version [16] for additional details). While the Greedy heuristic run time is also positively correlated with the budget increase with respect to the diffusion processes, the Simulated Annealing heuristic's run time is not correlated with the budget increase with respect to the plurality, approval, and ordinal diffusion processes. The intuitive reason is that, while the Greedy heuristic performs more iterations as the budget increases, Simulated Annealing operates by local improvement, irrespective of the budget.
2. We find that the Simulated Annealing heuristic takes less time to run than the Greedy heuristic for PoW, especially with high budgets with a large number of candidates and voters in any type of election. However, the MoV Simulated Annealing heuristic performs slower than the Greedy heuristic when used with a plurality or approval diffusion process. This is true for execution with the ordinal diffusion process only when budgets are relatively small.
3. The MoV performance of the Simulated Annealing heuristic is higher than that of the Greedy heuristic when measured as a function of the number of voters in terms of plurality and approval diffusion processes. In terms of the ordinal diffusion process, however, the Simulated Annealing heuristic's MoV performance is similar to that of Greedy.
4. The MoV performance of Simulated Annealing is similar to that of Greedy as a function of the number of candidates in terms of plurality, approval, and ordinal diffusion processes.
5. We found negative correlation between the number of voters and PoW performance for both heuristics with respect to all three diffusion processes. One exception to the above is no correlation between the number of voters and PoW performance using the Greedy heuristic for Copeland $G(n,p)$ and 5-clusters graphs.

The full details of our observations and graphs comparison are summarized in appendix B including Tables 1. Due to space limitation some of the plots supporting our findings are omitted.

5 Outlook

We proposed a general model for diffusion of opinions in social networks and considered heuristics that optimize over bribery schemes for several realizations of the model. Our model is general enough to incorporate various diffusion processes, including different ballot types and quite complex stochastic elements. We performed simulations to evaluate the performance of heuristic solutions that solve the task at hand; while our task is theoretically computationally intractable, our simulations are quite encouraging, in the sense that they reach good results in reasonable time. We also highlighted several parameter factors affecting the quality of the heuristics. We briefly discuss some avenues for future research.

5.0.1 Improved Heuristics

A natural future direction is to design better heuristic solutions, in particular such that are oblivious to the specific type of diffusion; E.g., local search heuristics with better initial solutions, other greedy approaches, as well as methods based on general solvers would be natural to try.

5.0.2 Other Settings

While we considered plurality, approval, and ordinal elections, it is natural to also consider utility-based elections and elections with cumulative ballots. Furthermore, there are other natural ways to treat the diffusion of complex opinions, most notably ordinal opinions. Fitting real-world data to various modeling choices would be an interesting future direction.

References

[1] Hunt Allcott and Matthew Gentzkow. Social media and fake news in the 2016 election. *Journal of economic perspectives*, 31(2):211–36, 2017.

[2] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. P rocaccia. *Handbook of computational social choice*. Cambridge University Press, 2016.

[3] R. Bredereck and E. Elkind. Manipulating opinion diffusion in social networks. In *Proceedings of IJCAI '17*, 2017.

[4] Ning Chen. On the approximability of influence in social networks. *SIAM J. Discret. Math.*, 23(3): 1400–1415, 2009. doi: 10.1137/08073617X. URL `https://doi.org/10.1137/08073617X`.

[5] Wei Chen, Yajun Wang, and Siyu Yang. Efficient influence maximization in social networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 199–208, 2009.

[6] Federico Coro, Emilio Cruciani, Gianlorenzo D'Angelo, and Stefano Ponziani. Exploiting social influence to control elections based on scoring rules. In *Proceedings of IJCAI '19*, 2019.

[7] E. Elkind and P. Faliszewski. Approximation algorithms for campaign management. In *Proceedings of WINE '10*, pages 473–482, 2010.

[8] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. Swap bribery. In *Proceedings of AGT '09*, pages 299–310. Springer, 2009.

[9] Nicole B Ellison, Cliff Lampe, and Charles Steinfield. Social network sites and society: Current trends and future possibilities. *Interactions*, 16(1):6, 2009.

[10] Ulle Endriss. *Trends in Computational Social Choice*. AI Access, 2017.

[11] P. Faliszewski and J. Rothe. Control and bribery in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 7. Cambridge University Press, 2015.

[12] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[13] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. How hard is bribery in elections? *Journal of artificial intelligence research*, 35:485–532, 2009.

[14] Piotr Faliszewski, Piotr Skowron, and Nimrod Talmon. Bribery as a measure of candidate success: Complexity results for approval-based multiwinner rules. In *Proceedings of AAMAS '17*, pages 6–14, 2017.

[15] Piotr Faliszewski, Rica Gonen, Martin Koutecký, and Nimrod Talmon. Opinion diffusion and campaigning on society graphs. In *Proceedings of IJCAI '18*, pages 219–225, 2018.

[16] Rica Gonen, Martin Koutecky, Roei Menashof, and Nimrod Talmon. Heuristics for opinion diffusion via local elections full version. 2022. URL `https://www.openu.ac.il/personal_sites/rica-gonen/`.

[17] Umberto Grandi, Emiliano Lorini, and Laurent Perrussel. Propositional opinion diffusion. In *Proceedings of AAMAS 2015*, pages 989–997, 2015.

[18] Adrien Guille, Hakim Hacid, Cecile Favre, and Djamel A Zighed. Information diffusion in online social networks: A survey. *ACM Sigmod Record*, 42(2):17–28, 2013.

[19] Qingye Jiang, Guojie Song, Gao Cong, Yu Wang, Wenjun Si, and Kunqing Xie. Simulated annealing based influence maximization in social networks. In *Proceedings of AAAI 2011*, volume 11, pages 127–132, 2011.

[20] D. Kempe, J. Kleinberg, and É. Tardos. Maximizing the spread of influence through a social network. In *Proceedings of KDD '03*, pages 137–146, 2003.

[21] Jure Leskovec, Andreas Krause, Carlos Guestrin, Christos Faloutsos, Jeanne VanBriesen, and Natalie Glance. Cost-effective outbreak detection in networks. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 420–429, 2007.

[22] Paulo Shakarian, Abhinav Bhatnagar, Ashkan Aleali, Elham Shaabani, and Ruocheng Guo. *Diffusion in Social Networks*. Springer, 2015.

[23] Jianxin Wang, Weimin Su, Min Yang, Jiong Guo, Qilong Feng, Feng Shi, and Jianer Chen. Parameterized complexity of control and bribery for d-approval elections. *Theoretical Computer Science*, 595: 82–91, 2015.

[24] Bryan Wilder and Yevgeniy Vorobeychik. Controlling elections through social influence. In *Proceedings of AAMAS '18*, pages 265–273, 2018.

A Diffusion Process for Ordinal Elections - Example

Example 3. *Consider voter v with open neighborhood $\{u_1, u_2\}$. Assume v has stubbornness $\alpha = 1$ and she votes for (x, y, z) while u_1 votes for (y, z, x) and u_2 votes for (z, y, x). Then the Borda score*[4] *would give $x = 5$, $y = 7$ and $z = 6$ and v's local Borda election will result with $c = (y, z, x)$. The first iteration over c would then be: $i = 1$, $c_i = y$, $j = 2$. With probability $\frac{\frac{7}{5}}{\frac{7}{5}+\frac{1}{\frac{7}{5}}} = 0.66$ candidate y will be bubbled up resulting with v voting for (y, x, z). If the first iteration resulted in v voting for (y, x, z), then the second iteration over c would be: $i = 2$, $c_i = z$, $j = 3$. With probability $\frac{\frac{6}{5}}{\frac{6}{5}+\frac{1}{\frac{6}{5}}} = 0.59$ candidate z will be bubbled up resulting with v voting for (y, z, x). If the second iteration resulted in v voting for (y, z, x), then the third iteration over c would be: $i = 3$, $c_i = x$, $j = 3$. With probability $\frac{\frac{5}{6}}{\frac{5}{6}+\frac{1}{\frac{5}{6}}} = 0.4$ candidate x will be bubbled up, and with probability 0.6 v's vote will not change from the last iteration and will remain (y, z, x). With the highest probability after all three iterations v's vote is identical to c's result, i.e., the Borda local election.*

Now consider Copeland as $\mathcal{R}_L$ and v, u_1, u_2 vote in the same way above. According to Copeland tournament y beats x, z beats x, and y beats z, and so y has two outgoing arcs and z has one outgoing arc. In order to avoid division by zero, we normalize the scores by adding 1. Then the Copeland score would give $x = 1$, $y = 3$, and $z = 2$, and the Copeland v's local election will result with $c = (y, z, x)$. The first iteration over c would be then: $i = 1$, $c_i = y$, $j = 2$. With probability $\frac{\frac{3}{1}}{\frac{3}{1}+\frac{1}{\frac{3}{1}}} = 0.9$ candidate y will be bubbled up resulting with v voting for (y, x, z). Then the second iteration over c would be: $i = 2$, $c_i = z$, $j = 3$. With probability $\frac{\frac{2}{1}}{\frac{2}{1}+\frac{1}{\frac{2}{1}}} = 0.8$ candidate z will be bubbled up resulting with v voting for (y, z, x). Then, finally, the third iteration over c would be: $i = 3$, $c_i = x$, $j = 3$. With probability $\frac{\frac{1}{2}}{\frac{1}{2}+\frac{1}{\frac{1}{2}}} = 0.2$ candidate x will be bubbled up, and with probability 0.8 v's vote will not change from the last iteration and will remain (y, z, x).

B Performance summary

Results comparing the various graph types are shown in Figure 2. The main insight is that, for plurality elections, the Simulated Annealing heuristic outperforms the Greedy heuristic in terms of MoV and PoW in the different types of graphs, but at a significantly higher time cost.

For graph comparisons with our observations see full version [16].

[4] To avoid division by zero, we define the Borda score of a candidate ranked as jth to be $|A| - j + 1$ instead of $|A| - j$, although the latter is more common. These definitions are mathematically equivalent.

		Plurality			Approval		
		Candidates	Budget	Voters	Candidates	Budget	Voters
PoW	Comparison	SA ≅ greedy	SA > greedy (slightly)	SA > greedy	SA > greedy (slightly)	SA > greedy	SA ≥ greedy
	Greedy Correlation	not correlated negatively correlated (Gnp)	positively correlated	negatively correlated (slightly)	negatively correlated (slightly)	positively correlated	negatively correlated
	SA Correlation	not correlated	positively correlated	negatively correlated (slightly)	not correlated	positively correlated	negatively correlated (slightly)
MoV	Comparison	SA > greedy	SA > greedy (slightly)	SA > greedy	SA ≥ greedy	SA ≥ greedy	SA > greedy (slightly)
	Greedy Correlation	not correlated	positively correlated	not correlated	not correlated	positively correlated	negatively correlated (slightly)
	SA Correlation	not correlated	positively correlated	negatively correlated (slightly)	not correlated	positively correlated	negatively correlated (slightly)
Runtime PoW	Comparison	greedy > SA (low budgets) SA > greedy (high budgets)	greedy > SA (high budgets) SA > greedy (low budgets)	greedy > SA (high budgets) SA > greedy (low budgets)	greedy > SA (high budgets) SA > greedy (low budgets)	greedy > SA (high budgets) SA > greedy (low budgets)	SA ≅ greedy (with exception in high budgets)
	Greedy Correlation	not correlated	positively correlated	positively correlated	positively correlated	positively correlated	positively correlated
	SA Correlation	not correlated	negatively correlated	positively correlated	positively correlated	negatively correlated (slightly)	positively correlated
Runtime MoV	Comparison	SA > greedy	SA > greedy	SA > greedy	SA > greedy (With exception in GNP)	SA > greedy greedy > SA (GNP high budgets)	SA > greedy (except for high budgets with high number of voters)
	Greedy Correlation	negatively correlated (slightly)	positively correlated (slightly)	positively correlated (slightly)	positively correlated (slightly)	positively correlated	positively correlated
	SA Correlation	positively correlated (slightly)	not correlated	positively correlated	positively correlated	not correlated	positively correlated

		Ordinal Borda			Ordinal Copeland		
		Candidates	Budget	Voters	Candidates	Budget	Voters
PoW	Comparison	greedy ≅ SA	greedy ≅ SA	greedy ≅ SA	greedy ≅ SA	greedy ≅ SA	greedy ≅ SA
	Greedy Correlation	negatively correlated	not correlated	negatively correlated	not correlated (gnp & 5-clusters) negatively correlated (real-world graph)	not correlated (gnp & 5-clusters) positively correlated (real-world graph)	not correlated (gnp & 5-clusters) positively correlated (real-world graph)
	SA Correlation	negatively correlated	not correlated	negatively correlated	not correlated (gnp & 5-clusters) negatively correlated (real-world graph)	not correlated (gnp & 5-clusters) positively correlated (real-world graph)	not correlated
MoV	Comparison	greedy ≅ SA	greedy ≅ SA	greedy ≅ SA	greedy ≅ SA	greedy ≅ SA	greedy ≅ SA
	Greedy Correlation	negatively correlated	not correlated (Gnp & 5-clusters) positively correlated (real world graph, slightly)	negatively correlated	negatively correlated	positively correlated	positively correlated (slightly)
	SA Correlation	negatively correlated	not correlated	negatively correlated	negatively correlated	not correlated	not correlated
Runtime PoW	Comparison	greedy > SA (when budget > 7)	greedy > SA (high budgets) SA > greedy (low budgets)	greedy > SA (when budget > 7)	greedy > SA (high budgets) SA > greedy (low budgets)	greedy > SA (high budgets) SA > greedy (low budgets)	greedy > SA (high budgets with high number of voters)
	Greedy Correlation	positively correlated	positively correlated	positively correlated	positively correlated	positively correlated (gnp & 5-clusters) not correlated (real-world graph)	positively correlated (gnp & 5-clusters) not correlated (real-world graph)
	SA Correlation	positively correlated	not correlated	positively correlated	positively correlated	not correlated	not correlated
Runtime MoV	Comparison	greedy > SA (when budget > 15)	greedy > SA (high budgets) SA > greedy (low budgets)	greedy > SA (high budgets with high number of voters)	greedy > SA (high budgets) SA > greedy (low budgets)	greedy > SA (high budgets) SA > greedy (low budgets) (With exceptions in greedy gnp)	greedy > SA (high budgets with high number of voters)
	Greedy Correlation	positively correlated	positively correlated	positively correlated	positively correlated (slightly)	positively correlated	positively correlated
	SA Correlation	positively correlated	not correlated	positively correlated	positively correlated	not correlated	not correlated

Figure 1: A summary of the performance of the Greedy and Simulated Annealing heuristics with respect to each other in the different scenarios of plurality and approval (top) ordinal Borda and Copeland (bottom) diffusion processes, optimization goal, and input parameters. Correlation calculated based on Pearson correlation coefficient.

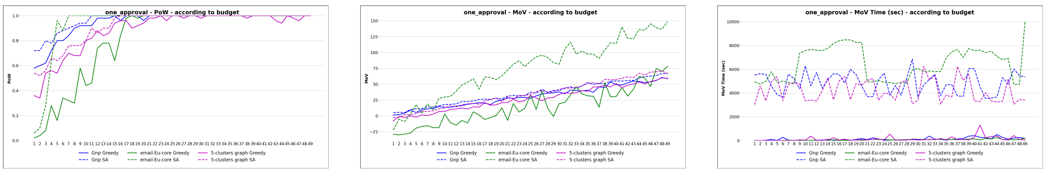

Figure 2: Results for comparing the Greedy and SA heuristics in plurality elections on 3 different graphs and labels. The leftmost (middle, rightmost) shows the results for PoW (respectively, MoV, running time) as a function of the budget b; The green (blue, purple) represents the real-world network graph, i.e., email-Eu-core network (respectively, $G(n,p)$, and k-clusters graph where k=5 (named in the plot 5-connected).

Rica Gonen
The Open University of Israel
Raanana, Israel
Email: ricagonen@gmail.com

Martin Koutecký
Charles University
Prague, Czech Republic
Email: koutecky@iuuk.mff.cuni.cz

Roei Menashof
The Open University of Israel
Raanana, Israel
Email: RoeiMena@gmail.com

Nimrod Talmon
Ben-Gurion University
Be'er Sheva, Israel
Email: talmonn@bgu.ac.il