

Opinion Diffusion and Campaigning on Society Graphs

Piotr Faliszewski¹, Rica Gonen², Martin Koutecký^{3,4}, Nimrod Talmon⁵

¹ AGH University of Science and Technology, Krakow, Poland

² The Open University of Israel

³ Technion - Israel Institute of Technology, Haifa, Israel

⁴ Charles University, Prague, Czech Republic

⁵ Ben-Gurion University, Be'er Sheva, Israel

faliszew@agh.edu.pl, ricagonen@gmail.com, koutecky@kam.mff.cuni.cz, talmonn@bgu.ac.il

Abstract

We study the effects of campaigning, where the society is partitioned into voter clusters and a diffusion process propagates opinions in a network connecting the clusters. Our model is very powerful and can incorporate many campaigning actions, various partitions of the society into clusters, and very general diffusion processes. Perhaps surprisingly, we show that computing the cheapest campaign for rigging a given election can usually be done efficiently, even with arbitrarily-many voters.

1 Introduction

The introduction of online social networks to modern political campaigning is a disruptive game changer, as it is now practical to influence individuals on a scale not possible before. Political campaigns now routinely use these networks to attempt to sway elections in their favor, for instance by targeting segments of voters with fake news [Allcott and Gentzkow, 2017]. To be efficient, campaigners would like to factor-in the nuances of how each voter segment behaves and how beliefs diffuse in the underlying social graph. However, following this line of reasoning to its natural conclusion, and relying on an unsegmented model to approach voters individually, leads to targeting algorithms that are unable to compute solutions in a reasonable timeframe. We show that refining the traditional campaign tools of polling and directly considering voter clusters presents a solution to this computational problem.

We study the diffusion of political ideas in a setting where a society is partitioned into clusters of voters. In our model, an external agent with limited funds observes a given election and then alters the preferences of the voters in some of the clusters (through campaign actions or bribes, targeted at specific voter groups). Voters' opinions then diffuse through a network connecting the voter clusters, until convergence, at which point an election winner is chosen according to a predetermined voting rule. We show that in many cases the external agent can efficiently compute the cheapest campaign

for swaying a given election, in time which depends exponentially on the number of candidates but only logarithmically on the number of voters. This makes our algorithm particularly well-suited for political elections, which often involve many voters, but only a handful of candidates.

In the quickly growing literature studying the interface between social choice and social networks (see, e.g., the book chapter of Grandi 2017), each voter is modeled as a vertex and edges represent relations between them (such as, e.g., friendships, common interests, or various forms of interaction). Swaying elections in this standard social network model, however, implies a number of classical worst-case and parameterized hardness results, even for the case of two candidates (i.e., in the case where each voter holds one of two possible opinions).¹ We study a different type of a network, which we refer to as a *society graph*, where each node represents a cluster of voters (e.g., people with similar preferences) and the edges represent possible channels of influence (e.g., clusters of similarly minded people may influence each other). By such clustering of voters, it is possible to model large, realistic elections, while still maintaining computational tractability (in our case, captured by fixed-parameter tractability with respect to the number of different clusters).

Our model is a generalization of the standard model because, as we use more and more fine-grained partitions, the clusters become smaller and smaller and, eventually, end up holding single individuals. More importantly, the model can capture many natural social behaviors within the clusters, a plethora of bribery actions, and various diffusion processes. In the context of an election campaign, this includes, e.g., using polling to understand voters' preferences and preference strength, and creating clusters based on this data, allowing the campaign managers to target those voters that are most likely to influence the final result. Our efficient algorithm naturally extends to these generalized settings.

¹See, e.g., Corollary 1 of Bredereck et al. [2017], showing computational hardness related to the Target Set Selection problem, and Theorem 2 of Wilder and Vorobeychik [2017], showing computational hardness related to the Influence Maximization problem.

Related Work. We study the possibility of manipulating outcomes of elections, assuming social diffusion processes. There is an extensive literature on manipulating elections, especially relevant are the works on bribery in elections [Faliszewski *et al.*, 2009], such as that on swap bribery [Elkind *et al.*, 2009], shift bribery [Elkind and Faliszewski, 2010; Bredereck *et al.*, 2016a], and bribery in multiwinner elections [Faliszewski *et al.*, 2017; Bredereck *et al.*, 2016b], as well as the survey of Faliszewski and Rothe [2015]. There is also a vast amount of literature on diffusion in social network, such as, e.g., the works of Kempe *et al.* [2003] and Chen *et al.* [2009], which, among other results, develop efficient heuristics for maximizing influence in social networks; we also point to reader to the book of Shakarian *et al.* [2015].

The study of the interface between social choice and social networks is gaining attention and is already quite established (see, e.g., Grandi [2017]). Contrary to these works, where each voter is modeled as a vertex, in our model voters are grouped into voter types (clusters), and edges correspond to relations between these types. Our view of voter types follows the model described by Knop *et al.* [2018], which considers manipulation on such so-called *societies*. (In this context we also mention the work of Izsak *et al.* [2018], which considers selecting a committee where candidate synergies are expressed on the candidate-group level.)

Bredereck and Elkind [2017] identify some cases where manipulating the diffusion of opinions in a social network can be done efficiently (e.g., elections with two candidates, held on a path). Wilder and Vorobeychik [2017] consider a related problem, albeit studying the probabilistic Linear Cascade model, while we (as well as Bredereck and Elkind) study the Linear Threshold model. Brill *et al.* [2016] do not consider bribery, but study diffusion of pairwise preferences with arbitrary number of candidates. Botan *et al.* [2017] consider a diffusion process where opinions regarding boolean propositions propagate through a social network. We also mention the work of Procaccia *et al.* [2015], which considers maximum likelihood estimations as related to a certain diffusion process of preferences, the work of Talmon [2017], which considers the possibility of using a social network to increase the social welfare when electing a proportional committee, and further related papers [Silva, 2016; Christoff and Grossi, 2017; Auletta *et al.*, 2015; Sina *et al.*, 2015].

2 Formal Model and Combinatorial Problem

In this section we present a very basic variant of our model, where voter types correspond to preference orders, edges exist between two orders that can be obtained by a single swap of adjacent candidates, the diffusion process is done in a simple, specific way, and the bribery actions are limited. Later, in Section 4, we discuss various generalizations of our approach, by considering arbitrary voter types, arbitrary bribery actions, and generalized diffusion processes. Yet, the basic model will allow us to develop useful intuitions, prove strong hardness results, and clearly present our tractability results. For $n \in \mathbb{N}$, by $[n]$ we mean the set $\{1, \dots, n\}$.

Elections and voting rules. We consider ordinal elections held with n voters, expressing preferences over m candidates

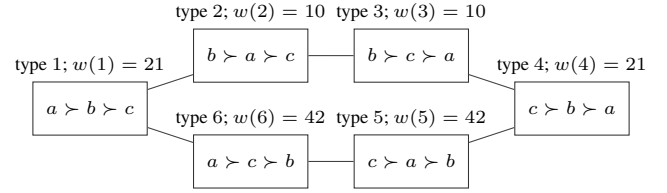


Figure 1: A society graph with three candidates and six types (corresponding to the six possible preference orders on those three candidates). E.g, there are 42 voters of type 6, each with preference order $a \succ c \succ b$; this graph corresponds to a society $\mathbf{w} = [21, 10, 10, 21, 42, 42]$.

$C = \{c_1, \dots, c_m\}$, where the *preference order* of a voter is a linear order over C . A voting rule \mathcal{R} is a function taking an election as input and returning a set of tied winners. A candidate winning under \mathcal{R} for a given election is called an \mathcal{R} -winner of the election. As an example, under the *Plurality* rule the candidates ranked first most frequently win, and under the *Borda* rule, each voter gives $m-i$ points to the candidate she ranks in the i -th position, and the candidates with the highest total number of points win.

Voter types, societies, and society graphs. For the time being, we let the preference order of a voter be her *type*. Thus, there are at most $\tau \leq m!$ types present in a given election and we order them so that we can speak of “the j -th type” for a given j . By the *weight* of voters with type j , denoted either w_j or $w(j)$, depending on the context, we mean the number of voters of type j . Sometimes we represent an election as a vector $\mathbf{w} \in \mathbb{N}^\tau$, whose j -th entry represents the weight of type j . We refer to such vectors as *societies*. (Here we somewhat follow the model of Knop *et al.* [2018].)

As we are interested in a certain diffusion process operating on the voter types, we associate a given election with a vertex-weighted graph $G = (V, \mathbf{w}, E)$, termed the *society graph*. The society graph contains τ vertices, where τ is the number of types in the election (specifically, $V = \{v_1, \dots, v_\tau\}$, where vertex v_j corresponds to voter type j , and its weight w_j is equal to the number of voters of that type in the given election). There is an edge between vertices v_j and $v_{j'}$ if the preference orders corresponding to types j and j' differ by the ordering of a single pair of adjacent candidates (in other words, if it is possible to transform one into the other with a single swap of two consecutive candidates). We show an example of a society graph in Figure 1.

Diffusion of preferences. Given a society graph (which encodes a given election), we consider two variants of a certain diffusion process: asynchronous and synchronous. In the asynchronous variant, in each step of the iterative diffusion process, a certain vertex v of the society graph G is picked, and then the following occurs. We consider the closed neighborhood $N[v]$ of v in G and check whether there is a neighbor x of v for which $w_x > 1/2 \sum_{u \in N[v]} w_u$; that is, a neighbor whose weight exceeds the sum of the weights of all other vertices in the closed neighborhood of v . If such an x exists, then we add the current weight w_v of v to that of x and change the weight w_v to be 0. Intuitively, the voters of type represented at v look at all the voters with similar or identical preferences,

and if there is a majority support among these voters for some preference order, they switch to it. In the synchronous variant we proceed in the same way, but simultaneously for all vertices. The diffusion process halts whenever it stabilizes.

Bribery in society graphs. Besides issues related to stability and convergence of society graphs, in this paper we are mainly interested in understanding the possibility of manipulating the outcome of an election. Thus we assume an external briber which has some budget and, using this budget, can change the way by which certain voters vote. Specifically, by one bribery action the briber can choose one voter and shift the position of his preferred candidate forward, by swapping him with the candidate ahead of him in the corresponding vote. (Indeed, this type of bribery is usually termed unit-cost shift-bribery [Bredereck *et al.*, 2016a].) Crucially, after the act of bribery takes place, the diffusion process initiates, and the goal of the briber is to have his preferred candidate win in the resulting election, using some predetermined voting rule. Formally, we are interested in the following general problem.

\mathcal{R} -BRIBERY IN SOCIETY GRAPHS (\mathcal{R} -BSG)

Input: A society graph G (given directly as a graph), a preferred candidate p , and a budget b .

Question: Are there at most b (unit-cost, shift-) bribery actions, such that, after performing them on G , and then running the diffusion process, p is an \mathcal{R} -winner of the election corresponding to G after convergence?

Corresponding to the synchronous and asynchronous diffusion processes, we consider both *sync*- \mathcal{R} -BSG and *async*- \mathcal{R} -BSG problems. For the asynchronous diffusion, we further consider the *optimistic* and *pessimistic* variants of the problem: In the former, we ask whether there exists some asynchronous diffusion process, initiated after the bribery actions, such that after reaching convergence p is an \mathcal{R} -winner; in the latter, we ask whether this happens for every possible order of diffusion steps. (Indeed, as shown in Example 1, the order of diffusion can sometimes affect the result of a given election.)

Remark 1. The input to \mathcal{R} -BSG is a labeled graph with weighted vertices (and a preferred candidate p and a budget b). Thus, the size of the input encoding is linear in the number of voter types and only logarithmic in the number of voters.

2.1 Convergence and Diffusion Order

Before we tackle the \mathcal{R} -BSG problem, we first show that consider the diffusion process always converges. For the synchronous case, this follows by arguing that in each diffusion step at least one vertex loses its weight completely, and a vertex of weight zero never increases its weight. In consequence, the number of synchronous diffusion steps is bounded by the number of voter types. The asynchronous case is even simpler, but requires appropriate terminology: If a diffusion step does not change the society graph (e.g., due to the choice of the vertex) then we call it *redundant*. We say that a sequence of non-redundant diffusion steps is *irredundant*. A *maximal irredundant sequence* is an irredundant sequence after executing which all remaining steps are redundant.

Proposition 1. *For each society graph G , the synchronous diffusion process converges in at most τ steps. The asynchronous diffusion process converges if the sequence contains a maximal irredundant sequence as a subsequence. The length of a maximal irredundant sequence is bounded by τ .*

Proof. For the asynchronous model, either no vertex changes any further, or at least one vertex, if chosen for the next diffusion iteration, will have its weight reduced to zero. Since no weight-zero vertex can ever increase its weight (by the definition of the diffusion step), it follows that every irredundant sequence consists of at most τ steps. By definition, if a sequence contains a maximal irredundant subsequence, it produces the same graph as this subsequence.

For the synchronous model, suffices to show that after every diffusion step (prior to convergence), the number of vertices with non-zero weight decreases. Consider a diffusion step before convergence. There is some vertex v , which is to be assimilated into one of its neighbors, u . If no other neighbor of v is to be assimilated by v in this step, then we are done: The number of vertices with non-zero weight would decrease by at least one. Perhaps, however, there is some neighbor v' of v that is to be assimilated by v in the current step. It must be that $v' \neq u$, as we require a strict majority for a vertex to be assimilated by one of its neighbors. If no neighbor of v' is assimilated by v' , then we are done (by the same token as before, we see that the number of weight-zero vertices will increase). Otherwise, there is some neighbor v'' of v' which is to be assimilated by v' . Exhaustively following this logic, either we reach a vertex whose weight is to decrease to zero, or some vertex repeats. However, the latter is impossible as, by definition of our diffusion process, the weights of the vertices that we encounter form a decreasing sequence. Thus the number of weight-zero vertices has to increase after executing the diffusion step and the proof is complete. \square

Convergence is guaranteed, but different asynchronous diffusion orders may lead to different election results.

Example 1. Consider the society graph depicted in Figure 1. If we select first type 3, then 4, then 6, and then 2, then we reach convergence with 115 voters with $c \succ a \succ b$ and 31 voters with $a \succ b \succ c$; thus, Plurality selects c . However, if we select first type 2, then 1, then 5, and then 3, then we reach convergence with 115 voters with $a \succ c \succ b$ and 31 voters with $c \succ b \succ a$; thus, Plurality selects a .

The above example suggests that an external agent might be interested in controlling the diffusion process, to help a certain candidate win. This task is NP-hard.

Theorem 1. *Given a society graph G and a preferred candidate p , deciding whether there is an order of asynchronous diffusion steps which results in p being a Plurality winner in the converged election is NP-hard.*

Proof. We reduce from the NP-hard problem Partition [Garey and Johnson, 1979], in which we are given integers x_1, \dots, x_n whose sum is $2B = \sum_{i \in [n]} x_i$ and we ask if it is possible to partition them into two groups of equal sum. We assume, w.l.o.g, that $x_i > n^7$ for each $i \in [n]$, and reduce a given instance of Partition to an instance of our problem.

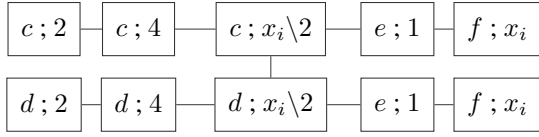


Figure 2: An element gadget used in the proof of Theorem 1.

We create five *important* candidates, p (the preferred candidate), c , d , e , and f , and we will also use a few dummy ones. We will make sure that for each *no*-instance of Partition, one of c , d , e , or f will be the election winner, irrespective of the order of asynchronous diffusion steps, whereas for each *yes*-instance, there will be a diffusion order leading to p 's victory.

We describe the society graph by specifying the weight and the candidate ranked first by each vertex, as well as the edges connecting these vertices; importantly, it is possible to realize such a graph by adding dummy candidates and setting the swap distances according to the edges. A vertex which ranks candidate x on top is called an x -vertex.

We construct a p -vertex v_p of weight $5B + 7n^5$, a c -vertex v_c of weight $4B$, a d -vertex v_d of weight $4B$, an e -vertex v_e of weight $5B + 7n^5 - n$, and an f -vertex v_f of weight $5B + 7n^5 - 4B - n$. These vertices are isolated (and remain so). The idea is that, as v_p is the only p -vertex, to make sure that p is the winner of the converged election, there can be c -vertices of total weight at most $B + 7n^5$, d -vertices of total weight at most $B + 7n^5$, e -vertices of total weight at most n , and f -vertices of total weight at most $4B + n$. The main part of the construction is an *element gadget*, which is a subgraph G_i , created for each integer x_i . Each G_i acts as a selection gadget, selecting, through different diffusion orders, whether a weight of roughly x_i will be counted for candidate c or d .

The subgraph G_i , $i \in [n]$, shown in Figure 2, contains the following 10 vertices: v_1 is a d -vertex of weight 2; v_2 is a d -vertex of weight 4; v_3 is a d -vertex of weight $x_i/2$; v_4 is a e -vertex of weight 1; v_5 is a f -vertex of weight x_i ; v_6 is a c -vertex of weight 2; v_7 is a c -vertex of weight 4; v_8 is a c -vertex of weight $x_i/2$; v_9 is a e -vertex of weight 1; v_{10} is a f -vertex of weight x_i . Each G_i is a connected component (in particular, different G_i 's are not connected to each other), which is internally connected using the edges $\{v_1, v_2\}$, $\{v_2, v_3\}$, $\{v_3, v_4\}$, $\{v_4, v_5\}$, $\{v_6, v_7\}$, $\{v_7, v_8\}$, $\{v_8, v_9\}$, $\{v_9, v_{10}\}$, $\{v_3, v_8\}$. This completes the reduction.

For correctness, consider a solution to the Partition instance, corresponding to $X \subseteq \{x_1, \dots, x_n\}$ such that $\sum_{x_i \in X} x_i = \sum_{x_i \notin X} x_i = B$. For each $x_i \in X$ we perform the following diffusion steps (the order is crucial): we assimilate v_1 into v_2 , v_2 into v_3 , v_8 into v_3 , v_4 into v_3 , v_6 into v_7 , and finally we assimilate v_9 into v_{10} . After these diffusion steps only v_3 (with weight $x_i + 7$), v_5 (with weight x_i), v_7 (with weight 6), and v_{10} (with weight $x_i + 1$), remain. Similarly, for each $x_i \notin X$ we assimilate v_6 into v_7 , v_7 into v_8 , v_3 into v_8 , v_9 into v_8 , v_1 into v_2 , and finally we assimilate v_5 into v_5 . After these diffusion steps only v_8 (with weight $x_i + 7$), v_{10} (with weight x_i), v_2 (with weight 6), and v_5 (with weight $x_i + 1$), remain.

After performing the above asynchronous diffusion steps for each element gadget, the diffusion process is conver-

ged and no more steps are possible. The plurality score of each of p , e , and f , is $5B + 7n^5$, as v_p is the only p -vertex, one e -vertex of weight 1 remains in each element gadget, and two f -vertices of total weight $2x_i + 1$ remain in each element gadget. The plurality score of each of c and d is $4B + B + 7|X| \leq 5B + 7n^5$. Thus, p is a plurality winner.

For the other direction, we assume a *no*-instance of Partition and show that no diffusion order results in p being a plurality winner. To see this, consider each element gadget and notice that for one of the e -vertices to be assimilated into a c -vertex or a d -vertex, it must be that v_3 is first assimilated into v_8 or that v_8 is first assimilated into v_3 . As the two e -vertices cannot remain e -vertices nor can both be assimilated into the f -vertices (to make sure that p indeed wins), each G_i is indeed a selection gadget, resulting in a plurality score of at least x_i for either c or d . Since we consider a *no*-instance of Partition, it follows that either c or d would be selected in G_i 's corresponding to x_i 's whose sum is strictly greater than B . As $x_i > n^7$ holds for each x_i , it follows that the score of either c or d would be at least $5B + n^7$. \square

3 Complexity of Manipulating Society Graphs

\mathcal{R} -BSG is generally intractable (both in the synchronous and asynchronous variants), but fixed-parameter tractable wrt. the number of candidates for a large class voting rules.

3.1 General Intractability of BSG

We first consider Borda as an example for intractability.

Proposition 2. *Borda-BSG is NP-hard both in the synchronous and in the asynchronous case.*

Proof. We provide a reduction from the problem Borda-SB (SB stands for Shift Bribery) in which, given an election with voters v_1, \dots, v_n and candidates c_1, \dots, c_m , a distinguished candidate p , and a budget b , the task is to decide whether b unit-cost shift bribery actions suffices to cause p to be a Borda winner. It is NP-hard [Bredereck *et al.*, 2016a, Proposition 3].

Given an instance of Borda-SB, we create an instance of Borda-BSG. The idea is to alter the instance so that, even after any set of bribery actions, any two voters would have swap distance at least 2. This would prevent any diffusion from happening, and, as Borda-BSG is in essence Borda-SB with diffusion, the result would follow. To this end, we introduce $2n$ additional dummy candidates $d_1, \dots, d_n, e_1, \dots, e_n$. We set the preference orders of the voters such that all voters prefer any of c_j to any of d_j and any of d_j to any of e_j , $j \in [n]$. Moreover, voter v_i prefers d_i to any d_j , $j \neq i$, and prefers e_i to any e_j , $j \neq i$. This finishes the description of the reduction. Correctness follows by the discussion above. \square

The above proof works for every voting rule for which (i) Shift Bribery with unit costs is NP-hard; and (ii) whose results do not change after we add some candidates which voters rank last. Such rules include, e.g., Copeland and Maximin [Bredereck *et al.*, 2016a] and, indeed, both conditions seem to be commonly satisfied (yet, Plurality is an example of a rule that fails the first criterion, and Veto is an example of a rule that fails the second one).

3.2 Voting Rules and Integer Linear Programs

In the next section we show that the BSG problem is fixed-parameter tractable for the parametrization by the number of candidates. Our algorithm is based on solving an integer linear program (ILP), and we use the next definition to capture the class of rules for which the algorithm is applicable.

Definition 1 (ILP- τ -expressible voting rule). Let $\mathbf{w} \in \mathbb{N}^\tau$ be a society representing an election E (where the j th element of \mathbf{w} represents the number of voters of type j) and $p \in C$ be a candidate. A voting rule \mathcal{R} is *ILP- τ -expressible* if there exists a computable function f and integers $\tau', r \leq f(\tau)$, a matrix $W \in \mathbb{Z}^{r \times (\tau + \tau')}$, and a vector $\mathbf{b} \in \mathbb{Z}^r$ such that:

$$\exists \mathbf{x} \in \mathbb{Z}^{\tau'} : W(\mathbf{w}, \mathbf{x}) \leq \mathbf{b} \quad \Leftrightarrow \quad p \in \mathcal{R}(E),$$

where $(\mathbf{w}, \mathbf{x}) \in \mathbb{Z}^{\tau + \tau'}$ is the concatenation of \mathbf{w} and \mathbf{x} .

Example 2 (Borda is ILP- τ -expressible). For a candidate $c \in C$, $i \in [m]$ and $j \in [\tau]$, by $\text{rank}(c, j)$ we mean the position on which c is ranked by the type- j voters. To show that Borda is ILP- τ -expressible, we give a collection of linear inequalities (defining the matrix W), satisfied exactly if the Borda score of $p \in C$ is not smaller than that of any other candidate:

$$\sum_{\substack{i \in [m], j \in [\tau] \\ \text{rank}(c, j) = i}} (m-i)w_j \leq \sum_{\substack{i \in [m], j \in [\tau] \\ \text{rank}(p, j) = i}} (m-i)w_j \quad \forall c \in C, c \neq p.$$

Definition 1 is a bit stronger than analogous definitions of Dorn and Schlotter [2012] and Faliszewski et al. [2011].

3.3 Fixed-Parameter Tractability of BSG

We prove that \mathcal{R} -BSG is FPT wrt. the number m of candidates for any ILP- m -expressible rule. We mention that, e.g., all scoring rules, all C1 rules (these are rules depending only on the majority graph), Bucklin, STV, and Kemeny, are ILP- m -expressible. The result follows by formulating \mathcal{R} -BSG as an integer linear program and invoking Lenstra's famous result [Lenstra Jr, 1983] (which implies that ILP is FPT wrt. number of integer variables); it is arguably quite surprising, since, as it turns out, it is possible to encode the complete diffusion process using integer variables and linear constraints.

Theorem 2. *Synchronous \mathcal{R} -BSG is fixed-parameter tractable with respect to the number m of candidates for any ILP- m -expressible voting rule \mathcal{R} .*

Proof. As a preprocessing phase, we augment the given society graph to have exactly $m!$ vertices, one vertex for each possible preference order; to this end, we might create some vertices of weight zero. Thus, the number of types in the input is $\tau = m!$, and the number of voters of type i , $i \in [\tau]$, is w_i . Let k be the number of steps of the diffusion process; Proposition 1 says that $k \leq \tau$. For $i \in [\tau]$, denote by $N[i]$ the closed neighborhood of i and by $N(i) = N[i] \setminus \{i\}$ its open neighborhood. For an expression exp , denote by $[\text{exp}]$ its binary value (e.g., $[2 > 1] = 1$ and $[1 > 2] = 0$).

We construct an ILP with the following variables. For each type $i \in [\tau]$ and diffusion step $\ell \in [k]$ we define an integer variable x_i^ℓ representing the number of voters of type i after ℓ diffusion steps. For types $i, j \in [\tau]$ we define variables β_{ij}

$$\sum_{j=1}^{\tau} \beta_{ij} = w_i \quad \forall i \in [\tau] \quad (1)$$

$$\sum_{i=1}^{\tau} \beta_{ij} = x_j^0 \quad \forall j \in [\tau] \quad (2)$$

$$z_{ij}^\ell = \left[\sum_{a \in N[i]} x_a^{\ell-1} \frac{1}{2} < x_j^{\ell-1} \right] \quad \forall j \in N(i), \ell \in [k] \quad (3)$$

$$\sum_{j \in N[i]} z_{ij}^\ell = 1 \quad \forall i \in [\tau] \quad (4)$$

$$t_{ij}^\ell = z_{ij}^\ell x_i^{\ell-1} \quad \forall i, j \in [\tau], \ell \in [k] \quad (5)$$

$$x_j^\ell = \sum_{i \in N[j]} t_{ij}^\ell \quad \forall j \in [\tau], \ell \in [k] \quad (6)$$

$$W(\mathbf{y}, \mathbf{x}^k) \leq \mathbf{w} \quad (7)$$

Figure 3: Constraints used in the proof of Theorem 2. For completeness, one shall make sure that the variables are in the right domains.

describing the bribery, where β_{ij} corresponds to the number of voters bribed from being of type i to being of type j ; note that we also consider β_{ii} , the number of voters of type i which are not bribed. For every $i, j \in [\tau]$ and $\ell \in [k]$, we define a binary variable z_{ij}^ℓ indicating whether in the ℓ -th step the voters of type i are being assimilated into type j (for technical reasons, we also use variables t_{ij}^ℓ ; see explanations below).

Let c_{ij} be the cost of bribing one voter of type i to become a voter of type j (we set c_{ij} to be ∞ if j is not reachable from i). As our aim is to minimize the cost of bribery, the objective of our ILP is to minimize $\sum_{i,j} c_{ij} \beta_{ij}$.

Our ILP contains the constraints presented in Figure 3. Constraints (1) and (2) are standard and express that the vector \mathbf{x}^0 describes the society after the bribery (recall that β_{ii} corresponds to non-bribed voters of type i).

Constraint (3) assigns 1 to z_{ij}^ℓ if the weight in type j exceeds half of the total weight of $N[i]$ and 0 otherwise. While linear in this form, it can be made linear using standard tricks, which introduce only constantly-many auxiliary variables [Bisschop, 2006, Section 7.4]. Note that we do *not* affect z_{ii}^ℓ here as the constraint goes only over j in the open neighborhood $N(i)$. Constraint (4) enforces that at least one of z_{ij}^ℓ is 1, and this includes z_{ii}^ℓ ; thus, if there is no $j \in N(i)$ with weight more than half of the weight of $N[i]$, then $z_{ii}^\ell = 1$ shall hold, which corresponds to i keeping its weight (i.e., voters of type i are not being assimilated into some other type). Constraints (5) and (6) define the weights for step ℓ , given the weights of step $\ell - 1$. Precisely, x_j^ℓ takes the weight of all its neighbors (including itself) who specify by $z_{ij}^\ell = 1$ that they shift their weight to j . We use the t_{ij}^ℓ variables as temporary variables that are non-zero for those i and j for which $z_{ij}^\ell = 1$. Notice that Constraint (5) is non-linear but again can be handled using standard tricks [Bisschop, 2006, Section 7.7]; it essentially implements the following:

$$[(z_{ij}^\ell = 1) \implies (t_{ij}^\ell = x_i^{\ell-1})] \wedge [(z_{ij}^\ell = 0) \implies (t_{ij}^\ell = 0)].$$

Finally, Constraint (7) corresponds to the specific voting rule being considered, with \mathbf{y} as the auxiliary variables (called \mathbf{x} in Definition 1); it is satisfied if and only if the given, preferred candidate p wins the election specified by \mathbf{x}^k (i.e., the election after the bribery and at the end of the diffusion process), and is expressible as \mathcal{R} is ILP- τ -expressible. \square

Corollary 1. *Asynchronous \mathcal{R} -BSG is FPT wrt. the number m of candidates for any ILP- m -expressible voting rule \mathcal{R} , for both the optimistic and the pessimistic variants.*

Proof. For the optimistic variant, we modify the ILP described above, as follows. We add variables y_i^ℓ , representing whether type i is updated in the ℓ th step, thus require $\sum_i y_i^\ell = 1$ to enforce that exactly one vertex is updated. We add variables \hat{z}_{ij}^ℓ and enforce that $\hat{z}_{ij}^\ell = z_{ij}^\ell \wedge y_i^\ell$ (expressing this is folklore, and can be done, e.g., by requiring $\hat{z}_{ij}^\ell \leq z_{ij}^\ell$, $\hat{z}_{ij}^\ell \leq y_i^\ell$); the interpretation is that i might be assimilated only if $y_i^\ell = 1$. Then, it suffices to replace z_{ij}^ℓ with \hat{z}_{ij}^ℓ in constraint (6). For the pessimistic variant, notice that any sequence which converges contains a maximal irredundant sequence, each of which is of length at most τ (this follows from Proposition 1). It thus suffices to consider the set of permutations of $[\tau]$, and expressing that in none of them p is losing can be done by a long conjunction of ILPs given by constraints (3)-(7), with a clause for each sequence. \square

4 Model Generalizations

Here We generalize the simple model described above and demonstrate far broader scenarios for which \mathcal{R} -BSG remains fixed-parameter tractable (e.g., models with arbitrary connections between voter types and models for which not all voters of a single type are assimilated in another).

Various voter types. Instead of partitioning the voters by preference orders, which might be too crude, we can consider arbitrary partitions. As the number of variables in the ILP described in the proof of Theorem 2 depends only on the number of types, it follows that \mathcal{R} -BSG remains fixed-parameter tractable with respect to the number τ of types.² Taken to the extreme, namely if we set each voter in a given election to constitute her own voter type, we arrive at the model of diffusion studied, e.g., by Bredereck and Elkind [2017].

Arbitrary bribery operations and manipulative actions. Our model can incorporate, e.g., all bribery operations mentioned by Faliszewski and Rothe [2015]. Indeed, the constants c_{ij} used in the proof of Theorem 2 encode the cost of transforming a voter of type i into a voter of type j and can be redefined for other bribery operations. Further, following the discussion of Knop et al. [2018, Section 3.2], this approach can be extended to other types of manipulative operations, such as to *voter control* [Faliszewski and Rothe, 2015], at no asymptotical cost in terms of computational complexity.

General diffusion processes. Our model can incorporate directed arcs, where a vertex would be influenced by those vertices for which it has an outgoing arc to and, in particular, we do not have to be confined to connections between types associated with preferences that differ in the ranking of a single pair of candidates. Further, those arcs can be weighted, representing different influence strengths (e.g., consider damping the influence of voters which are, swap

distance-wise, farther). Adding weights can be done by modifying Equation (3) in a straightforward way. Moreover, and most importantly, we can express in our model a large class of diffusion processes. The following definition is inspired by viewing the diffusion of preferences as an abstract process, in which each voter holds a local election to decide which preference order to assume. E.g., the diffusion process described in Section 2 corresponds to holding an election containing the voters of swap distance at most one, and changing to the preference order of the majority, if such exists.

Definition 2 (ILP- τ -expressible diffusion process). Let k be an upper bound on the number of diffusion steps, recall that for diffusion step $\ell \in [k]$, the variables $x_i^{\ell-1}$, $i \in [\tau]$ express the current society, and let f be a computable function. Then, an *ILP- τ -expressible diffusion process* is a process such that for each $i, j \in [\tau]$ and $\ell \in [k]$, there are integers $r(i, j, \ell), \tau(i, j, \ell) \leq f(\tau)$, a matrix $D_{i,j,\ell} \in \mathbb{Z}^{r(i,j,\ell) \times \tau + \tau(i,j,\ell)}$, and a vector $\mathbf{b}_{i,j,\ell} \in \mathbb{Z}^{r(i,j,\ell)}$ such that, in the ℓ -th diffusion step, voters of type i are assimilated into type j if and only if the following formula is satisfied:

$$\exists \mathbf{x}' \in \mathbb{Z}^{\tau(i,j,\ell)} \quad D_{i,j,\ell}(\mathbf{x}', \mathbf{x}_i^{\ell-1}) \leq \mathbf{b}_{i,j,\ell}.$$

Our basic diffusion process corresponds to Equation (3). Another ILP- τ -expressible diffusion process is that each voter replaces her preference order by the Kemeny ranking computed for the voters in her neighborhood.

Remark 2. Proposition 1 does not hold for all generalized diffusion processes, as the number of diffusion steps might not be bounded by the society graph's size. (Also, new voter types might sometimes appear as a result of diffusion steps.) Thus, the corresponding ILP to solve \mathcal{R} -BSG would have to be supplied with the number k of diffusion steps to simulate. Sometimes it indeed is plausible that an agent can estimate the number of diffusion steps to occur after the manipulative actions, e.g., when he knows the time of the election.

Theorem 3. *\mathcal{R} -BSG is fixed-parameter tractable with respect to the number τ of types and the number k of diffusion steps if both \mathcal{R} and the diffusion process are ILP- τ -expressible.*

We conclude this section with an example of a scenario which is captured by such generalized diffusion processes.

Example 3 (Multidimensional Societies). Consider voters of different age groups. It is plausible that the tendency to be influenced by other voters depends on age, and so we might have a voter type for each tuple of (preference order, age group), with different outgoing arcs and different diffusion conditions. This way we can express, e.g., that voters may be strongly influenced by some voter groups, yet cannot be assimilated into them (e.g., a junior person can be influenced by a senior one, but this does not make him or her senior).

5 Outlook

We described a powerful model capturing various scenarios of opinion diffusion in networks, under various manipulative actions. By considering voter types and society graphs, we were able to provide practically useful tractability results.

We hope this paper to have the side-effect of popularizing the ILP-techniques within in the COMSOC community.

²There is a technicality here as the number of variables depends both on the number of types in the input and on the number of preference orders that bribery operations may introduce.

Directions for future research include further studies of our *generalized diffusion processes* (including, e.g., finding sufficient conditions for convergence) and establishing the range of problems that our models can capture (including, e.g., analysis of probabilistic behavior of voters).

Acknowledgments

PF was supported by AGH grant 11.11.230.337 (statutory research). MK was supported by a Technion postdoctoral fellowship and the project 17-09142S of GA ČR.

References

- [Allcott and Gentzkow, 2017] H. Allcott and M. Gentzkow. Social media and fake news in the 2016 election. *Journal of Economic Perspectives*, 31(2):211–236, 2017.
- [Auletta *et al.*, 2015] V. Auletta, I. Caragiannis, D. Ferraioli, C. Galdi, and G. Persiano. Minority becomes majority in social networks. In *Proceedings of WINE '15*, pages 74–88, 2015.
- [Bisschop, 2006] J. Bisschop. *AIMMS Optimization Modeling*. Paragon Decision Technology BV, 2006.
- [Botan *et al.*, 2017] S. Botan, U. Grandi, and L. Perrussel. Propositionwise opinion diffusion with constraints. In *Proceedings of EXPLORE workshop, AAMAS '17*, 2017.
- [Bredereck and Elkind, 2017] R. Bredereck and E. Elkind. Manipulating opinion diffusion in social networks. In *Proceedings of IJCAI '17*, 2017.
- [Bredereck *et al.*, 2016a] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. Prices matter for the parameterized complexity of shift bribery. *Information and Computation*, 251:140–164, 2016.
- [Bredereck *et al.*, 2016b] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. Complexity of shift bribery in committee elections. In *Proceedings of AAAI '16*, pages 2452–2458, 2016.
- [Brill *et al.*, 2016] M. Brill, E. Elkind, U. Endriss, and U. Grandi. Pairwise diffusion of preference rankings in social networks. In *Proceedings of IJCAI '16*, pages 130–136, 2016.
- [Chen *et al.*, 2009] W. Chen, Y. Wang, and S. Yang. Efficient influence maximization in social networks. In *Proceedings of KDD '09*, pages 199–208, 2009.
- [Christoff and Grossi, 2017] Z. Christoff and D. Grossi. Stability in binary opinion diffusion. In *Proceedings of LORI '17*, pages 166–180, 2017.
- [Dorn and Schlotter, 2012] B. Dorn and I. Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica*, 64(1):126–151, 2012.
- [Elkind and Faliszewski, 2010] E. Elkind and P. Faliszewski. Approximation algorithms for campaign management. In *Proceedings of WINE '10*, pages 473–482, 2010.
- [Elkind *et al.*, 2009] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of SAGT '09*, pages 299–310, 2009.
- [Faliszewski and Rothe, 2015] 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.
- [Faliszewski *et al.*, 2009] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.
- [Faliszewski *et al.*, 2011] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.
- [Faliszewski *et al.*, 2017] P. Faliszewski, P. Skowron, and N. Talmon. Bribery as a measure of candidate success: Complexity results for approval-based multiwinner rules. In *Proceedings of AAMAS '17*, pages 6–14, 2017.
- [Garey and Johnson, 1979] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [Grandi, 2017] U. Grandi. Social choice and social networks. In U. Endriss, editor, *Trends in Computational Social Choice*. 2017.
- [Izsak *et al.*, 2018] R. Izsak, N. Talmon, and G. J. Woeginger. Committee selection with intraclass and interclass synergies. In *Proceedings of AAAI '18*, 2018.
- [Kempe *et al.*, 2003] 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.
- [Knop *et al.*, 2018] D. Knop, M. Koutecký, and M. Mnich. A unifying framework for manipulation problems. *arXiv preprint arXiv:1801.09584*, 2018.
- [Lenstra Jr, 1983] H. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of operations research*, 8(4):538–548, 1983.
- [Procaccia *et al.*, 2015] A. D. Procaccia, N. Shah, and E. Sodomka. Ranked voting on social networks. In *Proceedings of IJCAI '15*, pages 2040–2046, 2015.
- [Shakarian *et al.*, 2015] P. Shakarian, A. Bhatnagar, A. Aleali, E. Shaabani, and R. Guo. *Diffusion in social networks*. Springer, 2015.
- [Silva, 2016] A. Silva. Opinion manipulation in social networks. In *Proceedings of NETGCOOP '16*, pages 187–198, 2016.
- [Sina *et al.*, 2015] S. Sina, N. Hazon, A. Hassidim, and S. Kraus. Adapting the social network to affect elections. In *Proceedings of AAMAS '15*, pages 705–713, 2015.
- [Talmon, 2017] N. Talmon. Structured proportional representation. In *Proceedings of AAMAS '17*, pages 633–641, 2017.
- [Wilder and Vorobeychik, 2017] B. Wilder and Y. Vorobeychik. Controlling elections through social influence. *arXiv preprint arXiv:1711.08615*, 2017.