# Manipulating Opinion Diffusion in Social Networks

**Robert Bredereck**
University of Oxford
Oxford, United Kingdom,
TU Berlin, Germany
robert.bredereck@tu-berlin.de

**Edith Elkind**
University of Oxford
Oxford, United Kingdom
elkind@cs.ox.ac.uk

## Abstract

We consider opinion diffusion in binary influence networks, where at each step one or more agents update their opinions so as to be in agreement with the majority of their neighbors. We consider several ways of manipulating the majority opinion in a stable outcome, such as bribing agents, adding/deleting links, and changing the order of updates, and investigate the computational complexity of the associated problems, identifying tractable and intractable cases.

## 1 Introduction

It is Saturday morning. Alice wakes up, makes herself a cup of coffee, and logs into her Facebook account. She sees that her friend Betty has shared yet another news story on dangers of GMO corn, and there are likes and supportive comments from a few of their common friends. Alice wonders if perhaps she should stay away from GMO foods, just in case. She scrolls down, and sees a comment by her friend Carol on a post by David. Alice has met David at a party, but has not added him on Facebook. She hesitates for a moment, then presses the "Add friend" button. It then occurs to Alice that she has not seen any posts by Ed in a while. Has he left Facebook? Is he okay? Alice finds Ed's page, and realizes that Ed is still active on Facebook, sharing news of his favorite sports team and pictures of cats. Alice is not particularly interested in these posts, but, as she scrolls down the page, she sees that Ed was in her town a few weeks ago, and was asking if any of his Facebook friends were around.

The little story above illustrates several important aspects of social networks. First, we are influenced by opinions of our friends, and may change our opinion on an issue if a majority of our friends holds the opposite opinion on that issue. Second, social networks are dynamic: new friendships form, and old friendships deteriorate. Importantly, in case of online social networks, the algorithms that decide which information to display to a given user may affect the formation of new links and disappearance of the existing links: it is possible that the connection between Alice and David would not form if Alice was not shown Carol's comment on David's post, and Alice missed a chance to meet up with Ed because she did not react to his recent posts.

The first of these points is perhaps self-evident, and a number of formal models of the respective phenomenon, which is usually referred to as *opinion diffusion*, have been proposed in the literature (we point the reader to a recent survey by Grandi [2017]). There are also several models of how social networks may evolve; e.g., Alice and David forming a link is an example of the *triadic closure* phenomenon [Granovetter, 1973]. However, it is usually assumed that links appear and disappear organically, based on the network structure and possibly users' opinions on issues.

In contrast, motivated by research on bribery and control in voting theory [see, e.g., Faliszewski and Rothe, 2015], we take an adversarial perspective: our aim is to understand if, by making a small number of targeted changes to a network, one can ensure that a substantial fraction of users holds a particular opinion on a given issue once the opinion diffusion process converges. The specific changes we focus on are bribing a subset of users to change their opinion and removing links between users; some of our results also apply to the problem of adding new links and changing the order in which voters update their opinions.[1] We assume that each elementary operation (bribing a single user, adding/deleting a single link) has an associated cost, and we ask if the target outcome can be accomplished within a given budget. For clarity, we focus on a very simple model, where agents hold binary opinions (0 or 1), the social network is given by an undirected graph, and, when updating, agents only change their opinion if it differs from the opinion held by a strict majority of their neighbors.

Now, the details of the opinion diffusion process are important for our analysis: if all agents can be assumed to change their opinions in a lockstep (this is usually referred to as the *synchronous model*), then the opinion diffusion process evolves deterministically (though it is not guaranteed to converge to a single stable state; rather, it may 'flick' with period 2 [Goles and Olivos, 1980]). However, for a social network with more than a few users, this assumption is not realistic. On the other hand, with asynchronous updates, the final outcome may depend on the order of updates. To deal

---

[1]In practice, we do not expect a manipulator to directly add and remove links or define the update sequence, but a network provider (as a manipulator) can suggest friends or decide whether and at which point of time an agent gets to see information. By delaying the display of updates from some users the network effectively controls the order of updates.
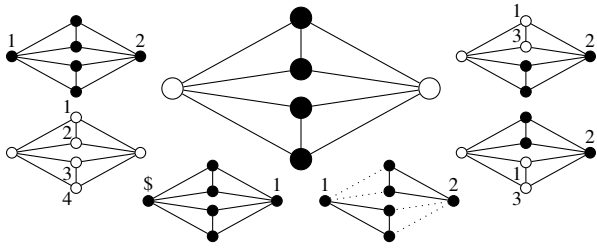
Figure 1: Influence network with six agents (large graph): The left-most and rightmost agents have opinion 0 (white) whereas the four central agents have opinion 1 (black). The small graphs show different possible stable outcomes. Numbers denote steps in which the respective agents update their opinion, a $-symbol means that the agent was bribed to change its opinion, and dotted lines indicate links that have been removed. With no manipulative actions, both opinions can become uniformly accepted by all agents depending on the order of updates (small graphs on the left). Opinions may stabilize in a balanced way (small graphs on the right). One can ensure that opinions stabilize towards the all-1s outcome irrespective of the order of updates, e.g. by bribing the leftmost agent (bottom left small graph) or by removing four edges (bottom right small graph).

with this issue, we identify two update sequences that are, in some sense, extreme: one results in the maximum possible number of 1s, and the other results in the maximum possible number of 0s. We show that any update sequence that results in the maximum number of 1s or 0s produces the same profile of opinions as the two canonical sequences that we consider; that is, the 1-maximizing sequence produces not just the maximum possible number of 1s, but the maximal set of users with that opinion. This means that, from the perspective of a manipulator who (without loss of generality) wants to maximize the number of 1s in the network, these update sequences represent, respectively, the best-case scenario and the worst-case scenario. Therefore, we focus on the complexity of achieving a desirable stable outcome by means of changing a network subject to a budget constraint, with respect to one of these update sequences. Towards the end of the paper, we also discuss what can be achieved by altering the update sequence itself. We refer to Figure 1 for an illustration of our model, including some manipulative actions.

The bribery problem is closely related to the well-studied TARGET SET SELECTION problem; we establish a formal connection between the two problems, which implies a number of classical worst-case and parameterized hardness results for our problem. On the other hand, we obtain a polynomial-time algorithm for the case where the social network is acyclic. Perhaps less obviously, a similar hardness reduction works for the link deletion and link insertion problems; the first of these problems also admits an efficient algorithm for acyclic networks. Finally, we investigate the case where the adversary controls the update order (but cannot bribe agents or remove links). Again, we get a hardness result for the general case, but an efficient algorithm for paths and cycles.

Of course, we are not claiming that any of the existing online social networks aims to promote certain opinions by means of manipulative actions: most likely, the decisions as to which posts to show to each user, and in what order, are

made with the aim of enhancing user's experience. However, the same mechanisms can be used to facilitate the spread of a specific opinion across the network, and therefore we feel that it is important to have a formal model that enables us to quantify the complexity of manipulative behavior.

**Related Work** There is a vast literature on various aspects of opinion diffusion [see, e.g., Grandi, 2017]. In the interest of space, in what follows we primarily survey results for our setting, where opinions are binary (1/0, or black/white) and updated according to the simple majority of neighbors, the graph is undirected, and agents are unweighted.

For binary opinions, the pioneering work of Goles and Olivos [1980] shows that a sequence of synchronous updates always converges to a stable state or cycles with period two. Their analysis also provides an upper bound of $\mathcal{O}(n^2)$ on the number of updates, where $n$ is the number of agents; this bound was subsequently shown to be essentially tight [Frischknecht *et al.*, 2013], both for the synchronous model and for the asynchronous model (in the latter case, the sequence of updates is assumed to be selected adversarially).

Our setting is closely related to that of *discrete preference games* [Chierichetti *et al.*, 2013]. In these games, each agent holds an initial opinion from a finite metric space, and has to decide which opinion to declare; the agent's cost is a weighted sum of the distance between his declared opinion and his true opinion and the distance between his declared opinion and those of his neighbors. Specifically, our model can be seen as a special case of discrete preference games where the opinion space is binary and agents do not care about the distance between their declared opinion and their initial opinion. Chierichetti *et al.* [2013] describe a linear-time procedure for finding a Nash equilibrium for binary discrete preference games; this procedure can be viewed as a sequence of asynchronous updates and is the basis of our analysis in Section 3. Discrete preference games have been subsequently studied by other authors; for instance, Auletta *et al.* [2015] characterize networks on which the opinion held by a minority of agents in the initial state may become the majority opinion in a stable state.

In a somewhat related paper, Yildiz *et al.* [2013] consider a binary opinion diffusion model that includes "stubborn agents", which influence others but do not change their opinions. Influencing the opinion in dynamic networks of agents at cost is described by Silva [2017], but for non-binary, continuous opinions. In this setting, optimal manipulation can be reduced to Knapsack and Mixed-Integer Programming, providing efficient algorithms for practical instances. The complexity of manipulating a binary opinion diffusion process has also been considered by Akutsu *et al.* [2006] in context of bioinformatics for a much more complex model with node-individual boolean functions defining whether nodes update their opinion, leading to NP-hardness even in extremely restricted settings.

Another related problem is TARGET SET SELECTION (TSS) [Kempe *et al.*, 2005]: Given an undirected graph $G$, a threshold function $\mathrm{thr} : V(G) \to \mathbb{N}$, and an integer $h \geq 0$, TSS asks whether there is a "target set" $V' \subseteq V(G)$ with $|V'| \leq h$ activating all vertices in $V$, where a vertex $v \in V(G) \setminus V'$ is activated when at least $\mathrm{thr}(v)$ of $v$'s

neighbors are active and an active vertex never becomes inactive; note that in this model the order of updates does not matter. This problem is computationally hard [Kempe *et al.*, 2005], even for special graph classes and restricted threshold functions. For example, Nichterlein *et al.* [2013] showed that TSS is W[2]-hard with respect to $h$ even for diameter two, and Chen [2009] showed hardness of approximation even for graphs with constant degree and for threshold two. While TSS is linear-time solvable on trees [Chen, 2009], Ben-Zwi *et al.* [2011] showed that TSS is W[1]-hard with respect to treewidth; however, some tractable cases are identified by Chopin *et al.* [2014].

## 2 Model and Notation

For a positive integer $n$, we set $[n] = \{1, \ldots, n\}$. An (undirected) *influence network* (InfNet) is a pair $(G, \circ)$ where $G = (V(G), E(G))$ is a simple, undirected graph with $|V(G)| = n$, $|E(G)| = m$, and $\circ : V(G) \to \{0, 1\}$ is an *opinion mapping* that describes the initial (binary) opinions of all vertices. For simplicity (and visualization) we say a vertex $v$ is *black* with respect to an opinion mapping $\circ$ if $\circ(v) = 1$ and *white* otherwise. For a vertex $v \in V$, we denote by $N(v) := \{u \in V \mid \{u, v\} \in E\}$ the *(open) neighborhood* of $v$, that is, the set of all vertices that are connected to $v$ by an edge. Let $d(u, v)$ denote the length of the shortest path between vertices $u, v \in V(G)$; the *diameter* of $G$ is $\mathrm{diam}(G) = \max_{u,v \in V(G)} d(u, v)$. We denote the *treewidth* of $G$ by $\mathrm{tw}(G)$; this is a parameter measuring how tree-like $G$ is [Robertson and Seymour, 1986].

## 3 Update Process

A vertex $v$ in an influence network $G$ may change its opinion during an *update step* if the opinion of a strict majority of vertices in $N(v)$ differs from her own opinion. Updates may be performed synchronously or asynchronously. We describe the order of updates by a sequence $\sigma$ of subsets of $V$; the $i$-th subset consists of vertices that consider changing their opinions at step $i$. The update sequence $(V(G), \ldots, V(G))$ models the *synchronous* update process used, e.g., by Goles and Olivos [1980]. An *asynchronous* update process corresponds to each subset being a singleton. We denote by $\circ[G, \sigma, z]$ the opinion mapping resulting from $\circ$ after $z$ update steps following the sequence $\sigma$. We use $\circ[G, \sigma]$ as shortcut for the final outcome $\circ[G, \sigma, |\sigma|]$. An opinion mapping $\circ$ is *stable* in a network $G$ if no vertex changes its opinion, i.e., $\circ[G, (V(G)), 1] = \circ$. A sequence is stable if the corresponding final outcome is stable.

**Optimistic and Pessimistic Updates** Allowing arbitrary update sequences leads to a huge number of possible outcomes. If the manipulator does not control the update sequence, it is natural for him to focus on the best-case and the worst-case scenarios. We say that an update sequence is *optimistic* (resp. *pessimistic*) if it is stable and maximizes (resp. minimizes) the number of black vertices in the final state. We will now describe an algorithm that constructs optimistic and pessimistic update sequences with very desirable properties; it is inspired by ideas of Chierichetti *et al.* [2013].

**Proposition 1.** *For every InfNet $(G, \circ)$ there is an optimistic (resp. pessimistic) update sequence $\sigma$ such that (i) $\sigma$ is asynchronous, (ii) $|\sigma| \leq 2n$, (iii) every vertex changes its opinion at most twice, (iv) $\sigma$ can be computed in $\mathcal{O}(n \cdot m)$ time, and (v) $\circ[G, \sigma] = \circ[G, \sigma^*]$ for every other optimistic (resp. pessimistic) update sequence $\sigma^*$.*

*Proof.* We describe a linear-time algorithm that constructs the desired optimistic update sequence $\sigma$; for the pessimistic case, one simply has to swap the phases of the algorithm (and adapt the proof of correctness accordingly). The algorithm starts with an empty sequence $\sigma$ and works in two phases.

(1) As long as there is a white vertex $v$ with a majority of black neighbors, append $\{v\}$ to the end of $\sigma$ and flip the opinion of $v$.

(2) As long as there is a black vertex $v$ with a majority of white neighbors, append $\{v\}$ to the end of $\sigma$ and flip the opinion of $v$.

Property (i) is clear by the definition of the algorithm. Properties (ii) and (iii) hold since the opinion of each vertex is flipped at most once in each phase. Property (iv) holds, because one can find white (resp. black) vertices with a majority of black (resp. white) neighbors in $\mathcal{O}(|V(G)| \cdot |E(G)|)$ time. It remains to show that property (v) holds, which also proves that $\sigma$ is optimistic.

Observe that $\sigma$ is indeed stable: By the definition of the second phase, no black vertex has an incentive to change its opinion. Further, no white vertex has an incentive to change its opinion at that point, as it has not gained new black neighbors compared to the end of the first phase. Now, consider an arbitrary optimistic update sequence $\sigma^*$. To prove that $\circ[G, \sigma] = \circ[G, \sigma^*]$, we show that (1) each vertex that flipped from white to black under $\sigma^*$ also flipped from white to black under $\sigma$ and that (2) each vertex that flipped from black to white under $\sigma$ is white under $\circ[G, \sigma^*]$. As the proofs are similar, we only provide the proof for case (1).

Assume towards a contradiction that there is a vertex that flipped from white to black under $\sigma^*$, but not under $\sigma$. Let $v^*$ be the first such vertex in $\sigma^*$, that is, $v^*$ flipped in step $\ell$ of $\sigma^*$ and all vertices that flipped from white to black in some step $\ell' < \ell$ of $\sigma^*$ also flipped from white to black under $\sigma$. Then all neighbors of $v^*$ that are black under $\circ[G, \sigma^*, \ell' - 1]$ would also be black at the end of the first phase of our algorithm. Hence, a majority of $v^*$'s neighbors would be black at the end of the first phase, yet $v^*$ would not flip; a contradiction with the definition of the first phase. □

## 4 InfNet-Bribery vs. Target Set Selection

We will now formalize the relationship between the problem of bribing the vertices to change their opinion and TARGET SET SELECTION. Formally, we consider the following computational problem.

OPTIMISTIC INFNET BRIBERY (OPT-INFNET-BRIB)
**Input:** An influence network $(G, \circ)$, and two positive integers $k$ and $q$.
**Question:** Can one change at most $k$ values in $\circ$ so that $\sum_{v \in V(G)} \circ[G, \sigma](v) \geq q$ for every optimistic update sequence $\sigma$?
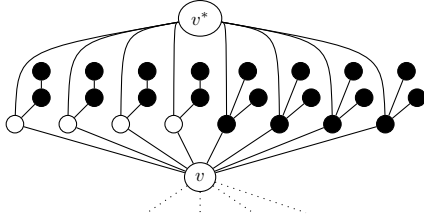
Figure 2: Stabilizer gadget for a vertex $v \in V(G)$ that has four neighbors.

PESSIMISTIC INFNET BRIBERY (PESS-INFNET-BRIB) is defined similarly.

First, observe that TSS with strict majority threshold functions can be seen as a special case of OPT INFNET-BRIB.

**Observation 1.** *Let $G$ be a graph and let $k$ be a positive integer. Then $((G, \circ), k, |V(G)|)$ with $\circ(v) = 0$ for all $v \in V(G)$ is in OPT INFNET-BRIB if and only if $(G, \mathrm{thr}, k)$ with $\mathrm{thr}(v) = \lceil (|N(v)| + 1)/2 \rceil$ for all $v \in V(G)$ is in TSS.*

We use the following reduction to show that both OPT and PESS INFNET-BRIB are at least as hard as TSS with arbitrary threshold functions in terms of approximability and parameterized complexity for several parameters, even if the bribery goal is to make all vertices black.

**Proposition 2.** *There is a polynomial-time algorithm that transforms every TSS instance $(G, \mathrm{thr}, h)$ into an OPT (resp. PESS) INFNET-BRIB instance $((G', \circ), k, q)$ such that*

- *$(G, \mathrm{thr}, h) \in$ TSS if and only if $((G', \circ), k, q) \in$ OPT (resp. PESS) INFNET-BRIB, and*
- *$k = h$, $q = |V(G')|$, $\mathrm{tw}(G') \in \mathcal{O}(\mathrm{tw}(G))$, $\mathrm{diam}(G') \le \mathrm{diam}(G) + 6$.*

*Proof.* We describe the reduction for OPT INFNET-BRIB and then discuss how to extend it for PESS INFNET-BRIB.

*Optimistic sequences.* We start with the graph $G' = G$, $\circ(v) = 0, \forall v \in V(G)$, $k = h$, and $q = |V(G)|$. By Observation 1, we are done if $\mathrm{thr}(v) = \lceil (|N(v)|+1)/2 \rceil \forall v \in V(G)$.

Let $y(v) := \mathrm{thr}(v) - \lceil (|N(v)| + 1)/2 \rceil$. Now, we show how to fix the construction by appending some dummy vertices to each vertex $v \in V(G)$ with $y(v) \ne 0$. If $y(v) > 0$, we add $2y(v)$ white vertices $d_w(v, i)$, $i \in [y(v)]$, to $V(G')$, for each $i \in [y(v)]$ add the edge $\{v, d_w(v, i)\}$ to $E(G')$, and increase $q$ by $|y(v)|$. If $y(v) < 0$, we add $2|y(v)|$ pairs of black vertices $d_b(v, i, 1)$, $d_b(v, i, 2)$, $i \in [y(v)]$, to $V(G')$, for each $i \in [y(v)]$ add the edges $\{v, d_b(v, i, 1)\}$ and $\{d_b(v, i, 1), d_b(v, i, 2)\}$ to $E(G')$, and increase $q$ by $2|y(v)|$.

Let us discuss the correctness of the construction for OPT INFNET-BRIB. Consider a target set $V' \subseteq V(G)$. It can be verified that by bribing all target set vertices, we can ensure that all original vertices from $V(G)$ become black when applying an optimistic update sequence. Clearly, the (white) dummy vertices will also become black.

For the converse direction, assume that there is an opinion mapping $\circ'$ which differs from $\circ$ in at most $k$ values such that $\sum_{v \in V(G')} \circ'[G', \sigma](v) \ge q$ with respect to some optimistic update sequence $\sigma$. Let $V' = \{v \in V(G') \mid \circ(v) \ne \circ'(v)\}$ be the set of vertices whose opinion has changed. Now, to

construct a set $V^*$ from $V'$, we replace each dummy vertex by the respective original vertex (that is, any vertex from $\{d_w(v, i, 1), d_b(v, i, 1), d_b(v, i, 2)\}, i \in [y(v)]$ is replaced y $v$). It can be verified that such set $V^* \subseteq V(G)$ is a target set for $G$. Finally, observe that the treewidth is not affected by appending additional leaves.

*Pessimistic sequences.* If we use the above reduction, then the converse direction of the proof still holds. However, bribing all vertices of a given target set does not ensure that all original vertices from $V(G)$ become black when applying a pessimistic update sequence since, initially, some of the target set vertices might have a majority of white neighbors. To ensure that such vertices do not immediately become white, we extend the construction by appending a *stabilizer gadget* to each non-dummy vertex (see Figure 2): For each vertex $v \in V(G)$ add an *activator vertex* $v^*$ as well as $|N(v)|$ white and $|N(v)|$ black *connector vertices*. For each connector vertex $w$, add edges $\{v, w\}$ and $\{v^*, w\}$. Furthermore, append two additional black leaves to each black connector vertex and append a path of two black vertices to each white connector vertex. The stabilizer gadget has three important properties, which hold under *every* stable update sequence. First, no vertex will ever flip (back) from black to white. Second, the whole gadget becomes black when any of its white vertices flips. Third, the original vertex attached to the stabilizer gadget must become black as well if the gadget becomes black.

Let us discuss the correctness of the modified construction for PESS INFNET-BRIB. Due to the properties of the stabilizer gadget discussed above, the proof for the converse direction still goes through. Let $V' \subseteq V(G)$ be a target set of size $k$. By considering the pessimistic update sequence constructed in Proposition 1, we can verify that bribing all vertices from $V'' := \{v^* \mid v \in V'\}$ will ensure that all vertices from $V(G)$ become black: During the update process, no black vertex will ever have a majority of white neighbors. (Indeed, this holds for *every* stable update sequence.)

Finally, we discuss some parameters of the constructed instance. It can be shown that appending the stabilizer gadgets does not increase the treewidth by more than a factor of two (we omit the argument due to space constraints). The diameter is increased by at most 6, since every newly introduced vertex is at distance at most 3 from some original vertex. □

Proposition 2 enables us to transfer many negative hardness results from TSS to INFNET-BRIB. We concentrate on the parameterized intractability for the parameters (1) maximum number of agents to bribe, (2) diameter, and (3) treewidth of the influence network, because we consider these parameters to appear most promising for the exploration of tractable special cases. We remark that Chen [2009] showed that it is hard to approximate the size of the target set even for graphs with maximum degree bounded by a constant and when the threshold of every vertex is two. This translates to hardness of approximating the number of agents to bribe for INFNET-BRIB on graphs with maximum degree bounded by a constant.

**Corollary 1.** OPT *(resp.* PESS*)* INFNET BRIBERY *is* NP-*complete,* W[2]-*hard parameterized by the number $k$ of ver-*

*tices to bribe even if the diameter of $G$ is at most 8, and* W[1]-*hard parameterized by the treewidth of the input graph.*

*Proof.* Membership in NP is immediate. All other claims follow directly from Proposition 2, because TSS is W[2]-hard with respect to the number $h$ of vertices to activate even for diameter two [Nichterlein *et al.*, 2013] and W[1]-hard with respect to treewidth [Ben-Zwi *et al.*, 2011]. □

Despite these negative results, we can show that INFNET-BRIB becomes polynomial-time solvable when the influence network is acyclic. The following theorem relies on a dynamic programming algorithm similar to the one in the proof of Theorem 2 (see Section 5).

**Theorem 1.** OPT *(resp.* PESS*)* INFNET BRIBERY *can be solved in polynomial time for acyclic influence networks.*

Theorem 1 is mainly a classification result. It remains open whether there is a practical, efficient algorithm as for TSS [Chen, 2009].

## 5 Controlling Network Links

While computational hardness results for INFNET BRIBERY, which is very similar to TARGET SET SELECTION, are not particularly surprising, in this section we show that for a seemingly significantly less powerful action[2], such as removing a link in a network, similar hardness results hold. Formally, we consider the following computational problem.

OPTIMISTIC INFNET UNLINK CONTROL (OPT-INFNET-UNLINK)
**Input:** An influence network $(G, \circ)$, and two positive integers $k$ and $q$.
**Question:** Can one remove at most $k$ edges from $G$ so that $\sum_{v \in V(G)} \circ[G, \sigma](v) \geq q$ for any optimistic update sequence $\sigma$?

PESSIMISTIC INFNET UNLINK CONTROL (PESS-INFNET-UNLINK) is defined similarly.

**Proposition 3.** *There is a polynomial-time algorithm that transforms every* TSS *instance* $(G, \text{thr}, h)$ *into an* OPT *(resp.* PESS*)* INFNET-UNLINK *instance* $((G', \circ), k, q)$ *such that (i)* $(G, \text{thr}, h) \in$ TSS *if and only if* $((G', \circ), k, q) \in$ OPT *(resp.* PESS*)* INFNET-UNLINK*, and (ii)* $k = h$, $q = |V(G')|$, $\text{tw}(G') \in \mathcal{O}(\text{tw}(G))$, $\text{diam}(G') \leq \text{diam}(G) + 6$.

*Proof Idea.* We can use the same construction as used for the pessimistic case in Proposition 2. To see this, observe two things. First, to "activate" some vertex $v \in V(G)$ and the corresponding stabilizer gadget, one can simply remove an edge between $v^*$ and one if its white connector vertices. Then, any stable update sequence will make the whole gadget—and then also $v$—become black. Second, when there is a solution for the INFNET-UNLINK instance that removes an original edge $\{u, v\}$, then one can obtain an alternative solution by removing an edge between $u^*$ (resp. $v^*$) and one of its white connector vertices. □

---

[2]For example, with unbounded budget INFNET-BRIB becomes trivial whereas INFNET-UNLINK remains non-trivial.

**Corollary 2.** OPT *(resp.* PESS*)* INFNET UNLINK CONTROL *is* NP-*complete,* W[2]-*hard parameterized by the number* $k$ *of vertices to bribe even if the diameter of* $G$ *is at most 8, and* W[1]-*hard parameterized by the treewidth of the input graph.*

We mention that by a slight adaption of the stabilizer gadget we can obtain the same set of intractability results for the problem of *adding* at most $k$ links.

Just as for INFNET-BRIB, we show that INFNET-UNLINK becomes polynomial-time solvable for acyclic networks.

**Theorem 2.** OPT *(resp.* PESS*)* INFNET-UNLINK *can be solved in polynomial time for acyclic influence networks.*

*Proof Idea.* We describe the main idea of the algorithm and omit some technical details. Also, to avoid technicalities, we focus on the optimistic case and assume that the graph is a tree, which is arbitrarily rooted. (The pessimistic case is very similar and the extension to forests is straightforward.) The crucial idea is that we can use property (iii) in Proposition 1 in order to characterize the influence of the root of a subtree on all of its descendants. To this end, we define a set $S = \{b, wb, bw, wbw, w\}$ of five possible states for each vertex (with respect to our canonical optimistic update sequence):

b The vertex is always black.

wb The vertex flips from white to black.

bw The vertex flips from black to white.

wbw The vertex flips from white to black and back.

w The vertex is always white.

In order to cover the case that the edge between some vertex and its parent is removed, we annotate each state $x \in S$ by an asterisk. Thus, we end up with a set $S^* := \{s, s^* \mid s \in S\}$ of ten states when also considering edge removals.

Let table entry $T[v, \ell, s_v, s_p]$ contain the maximum number of black vertices within the subtree rooted in $v \in V(G)$ at the end of an optimistic update process given that the root $v$ has the state $s_v \in S^*$, its parent has state $s_p \in S^*$, and we removed at most $\ell$ edges within the subtree.

It is easy to see how to initialize table $T$ by filling in all entries for the leaf nodes. The update process, that is, filling the entries for non-leaf vertices is non-trivial and relies on solving an instance of a special case of THREE-DIMENSIONAL MULTIPLE-CHOICE KNAPSACK (3-MCKP) [Kellerer *et al.*, 2004]. Given $n$ classes of items, where each item has a three-dimensional weight and a utility value, 3-MCKP finds a selection of $n$ items, one from each class, which maximizes the total utility and satisfies a capacity constraint for each weight dimension.

Let us focus on $T[v, \ell, s_v, s_p]$ for some non-leaf $v$. The difficulty of computing this table entry lies in the fact that (i) we have to decide how to distribute the at most $\ell$ (resp. at most $\ell - 1$ if $s_v \in S^* \setminus S$) edge removals to the children of $v$, and (ii) we have to ensure that the state $s_v$ can indeed be realized. Note that the second part would be easy if we knew the states of all children of $v$ (the state of $v$'s parent is fixed as $s_p$). However, we cannot afford to go through all possible state combinations of the children of $v$ and ways of distributing the edge removals, because the degree of $v$ is unbounded. Instead, each possible realization of the subtree

rooted in some child of $v$ can be seen as a potential item for our knapsack. More precisely, for each child $v'$ we introduce a class $X_{v'}$ of items. For each $\ell' \leq \ell$ and each state $s' \in S^*$ the class $X_{v'}$ contains the item $x(v', \ell', s')$ whose value is defined as $T[v', \ell', s', s_v]$ and whose three-dimensional weight is as follows: the first dimension is the number of edge removals and has value $\ell'$, the second dimension is the color of $v'$ at the end of the first phase of the optimistic update process, and the third dimension is the color of $v'$ at the end of the optimistic update process. Herein, black is encoded by weight 1, white by weight -1, and a disconnected child by weight 0.5. A capacity constraint on the first dimension ensures that all edge removals are distributed, and capacity constraints on the other two dimensions ensure that the state of $v$ is consistent with the state of its neighbors. Hence, a solution to our 3-MCKP instance gives as the correct value of $T[v, \ell, s_v, s_p]$. Note that 3-MCKP with weights encoded in unary can be solved in polynomial time via dynamic programming. □

## 6 Controlling Update Sequences

In the previous sections, we focused on manipulative actions that aimed to maximize the number of black vertices, and only considered two specific classes of update sequences (optimistic and pessimistic updates). Another interesting question is whether we can achieve a specific outcome by allowing arbitrary update sequences. We restrict ourselves here to the (seemingly) simpler case where only asynchronous updates are allowed. Thus, we obtain the following problem.

INFNET ASYNC-UPDATE SEQUENCE (INFNET-ASEQ)
**Input:** An undirected graph $G = (V, E)$, an initial opinion mapping $\circ : V \to \{0, 1\}$, and a stable opinion mapping $\circ' : V \to \{0, 1\}$.
**Question:** Is there an asynchronous update sequence $\sigma$ such that $\circ[G, \sigma] = \circ'$?

It turns out that most of the hardness results that we obtained for INFNET-BRIB also hold for INFNET-ASEQ, even though INFNET-ASEQ does not even involve a budget, and INFNET-BRIB becomes trivial with unlimited budget. The construction behind the following proposition is a non-trivial extension of the construction in Proposition 3. The high-level idea is to introduce a gadget that allows up to $k$ of the activator vertices from the stabilizer gadget to become black before some part of this new gadget becomes white.

**Proposition 4.** *There is a polynomial-time algorithm that transforms every* TSS *instance* $(G, \text{thr}, h)$ *into a* INFNET-ASEQ *instance* $(G', \circ, \circ')$ *such that*

- $(G, \text{thr}, h)$ *is in* TSS *if and only if* $(G', \circ, \circ')$ *is in* INFNET-ASEQ*, and*
- $\text{tw}(G') \in \mathcal{O}(\text{tw}(G))$, $\text{diam}(G') \leq \text{diam}(G) + 6$.

The construction behind Proposition 4 preserves most of the negative results that we have derived for INFNET-BRIB and INFNET-UNLINK. However, due to the global activation gadget that is connected to every stabilizer gadget, it does not preserve (approximation) hardness for bounded degree. Further, the diameter of the graph is 12 instead of 8.
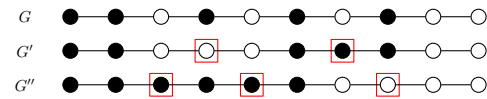


Figure 3: Visualization of possible outcomes ($G'$ and $G''$) for some path $G$. No vertex that agrees with some neighbor can change its opinion which implies that no two neighbors can change their opinion (after each other).

**Corollary 3.** INFNET ASYNC-UPDATE SEQUENCE *is* NP-*complete even if the diameter of* $G$ *is at most* 12*, and* W[1]-*hard parameterized by the treewidth of the input graph.*

Identifying the first tractable special case, we show that INFNET-ASEQ becomes linear-time solvable when the degree of every vertex in the network does not exceed two (i.e., the network is a collection of paths and cycles). The main insight is illustrated in Figure 3.

**Theorem 3.** INFNET ASYNC-UPDATE SEQUENCE *can be solved in linear time if the degree of every vertex in* $G$ *is at most two.*

## 7 Conclusion

We have proposed several models of manipulating the process of opinion diffusion on social networks. It turns out that most of the manipulation problems we consider are computationally hard. This is good news, as computational hardness provides partial defence against strategic behavior [see, e.g., the discussion by Faliszewski and Procaccia, 2010].

It would be interesting to extend our analysis to directed networks; while the hardness results still hold, for easiness results the picture is not clear. Control by adding links deserves further study as well. For INFNET-ASEQ, we focused on the question of arriving to a specific combination of opinions. However, there are many other interesting questions one can ask, such as whether we can achieve a good balance of opinions in the society (e.g., at least 30% of each color), or whether we can ensure that each player is exposed to an opinion that differs from her own (note that maximizing or minimizing the number of black vertices is easy, as this can be achieved by optimistic/pessimistic update sequences). In this context, one can also ask what happens if we allow simultaneous updates by arbitrary vertex subsets. We restricted ourselves to stable update sequences but it could be useful to omit this assumption. More sophisticated network models (e.g. including unobservability) would allow us to avoid full-knowledge assumption. Further, we treat adding and removing links as permanent changes to the network. Instead, one can allow the manipulator to disable/enable links for a few steps; indeed, this is something that can be done quite inconspicuously in an online social network. One can also ask what can be accomplished by combining different control actions in the spirit of Faliszewski *et al.* [2011].

## Acknowledgments

# References

[Akutsu *et al.*, 2006] Tatsuya Akutsu, Morihiro Hayashida, Wai-Ki Ching, and Michael K. Ng. On the complexity of finding control strategies for Boolean networks. In *Proceedings of the 4th Asia-Pacific Bioinformatics Conference (APBC'06)*, pages 99–108. Imperial College Press, 2006.

[Auletta *et al.*, 2015] Vincenzo Auletta, Ioannis Caragiannis, Diodato Ferraioli, Clemente Galdi, and Giuseppe Persiano. Minority becomes majority in social networks. In *Proceedings of the 11th International Conference on Web and Internet Economics (WINE'15)*, pages 74–88. Springer, 2015.

[Ben-Zwi *et al.*, 2011] Oren Ben-Zwi, Danny Hermelin, Daniel Lokshtanov, and Ilan Newman. Treewidth governs the complexity of target set selection. *Discrete Optimization*, 8(1):87–96, 2011.

[Chen, 2009] Ning Chen. On the approximability of influence in social networks. *SIAM Journal on Discrete Mathematics*, 23(3):1400–1415, 2009.

[Chierichetti *et al.*, 2013] Flavio Chierichetti, Jon M. Kleinberg, and Sigal Oren. On discrete preferences and coordination. In *Proceedings of the 14th ACM Conference on Electronic Commerce (EC'13)*, pages 233–250. ACM, 2013.

[Chopin *et al.*, 2014] Morgan Chopin, André Nichterlein, Rolf Niedermeier, and Mathias Weller. Constant thresholds can make target set selection tractable. *Theory of Computing Systems*, 55(1):61–83, 2014.

[Faliszewski and Procaccia, 2010] Piotr Faliszewski and Ariel Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):52–64, 2010.

[Faliszewski and Rothe, 2015] Piotr Faliszewski and Jörg 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*. Cambridge University Press, 2015.

[Faliszewski *et al.*, 2011] Piotr Faliszewski, Edith Hemaspaandra, and Lane Hemaspaandra. Multimode control attacks on elections. *Journal of AI Research*, 40:305–351, 2011.

[Frischknecht *et al.*, 2013] Silvio Frischknecht, Barbara Keller, and Roger Wattenhofer. Convergence in (social) influence networks. In *Proceedings of the 27th International Symposium on Distributed Computing (DISC'13)*, volume 8205 of *LNCS*, pages 433–446. Springer, 2013.

[Goles and Olivos, 1980] Eric Goles and Jorge Olivos. Periodic behaviour of generalized threshold functions. *Discrete Mathematics*, 30(2):187–189, 1980.

[Grandi, 2017] Umberto Grandi. Social choice and social networks. In U. Endriss, editor, *Trends in Computational Social Choice*. AI Access, 2017.

[Granovetter, 1973] Mark S. Granovetter. The strength of weak ties. *American Journal of Sociology*, 78(6):1360–1380, 1973.

[Kellerer *et al.*, 2004] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack problems*. Springer, 2004.

[Kempe *et al.*, 2005] David Kempe, Jon M. Kleinberg, and Éva Tardos. Influential nodes in a diffusion model for social networks. In *Proceedings of 32nd International Colloquium on Automata, Languages and Programming (ICALP'05)*, pages 1127–1138, 2005.

[Nichterlein *et al.*, 2013] André Nichterlein, Rolf Niedermeier, Johannes Uhlmann, and Mathias Weller. On tractable cases of target set selection. *Social Network Analysis and Mining*, 3(2):233–256, 2013.

[Robertson and Seymour, 1986] Neil Robertson and Paul D. Seymour. Graph minors. II. algorithmic aspects of treewidth. *Journal of Algorithms*, 7(3):309–322, 1986.

[Silva, 2017] Alonso Silva. *Opinion Manipulation in Social Networks*, pages 187–198. Springer, 2017.

[Yildiz *et al.*, 2013] Ercan Yildiz, Asuman Ozdaglar, Daron Acemoglu, Amin Saberi, and Anna Scaglione. Binary opinion dynamics with stubborn agents. *ACM Transactions on Economics and Computation*, 1(4):19:1–19:30, 2013.