Eliminating Majority Illusions

Foivos Fioravantes
Czech Technical University of Prague
Prague, Czech Republic
foivos.fioravantes@fit.cvut.cz

Abhiruk Lahiri
Heinrich Heine University
Düsseldorf, Germany
abhiruk@hhu.de

Antonio Lauerbach
Julius-Maximilians-University
Würzburg, Germany
antonio.lauerbach@stud-mail.uni-wuerzburg.de

Lluís Sabater
Charles University
Prague, Czech Republic
llsabater@iuuk.mff.cuni.cz

Marie Diana Sieper
Julius-Maximilians-University
Würzburg, Germany
marie.sieper@uni-wuerzburg.de

Samuel Wolf
Julius-Maximilians-University
Würzburg, Germany
samuel.wolf@uni-wuerzburg.de

ABSTRACT

An opinion illusion refers to a phenomenon in social networks where agents may witness distributions of opinions among their neighbours that do not accurately reflect the true distribution of opinions in the population as a whole. A specific case of this occurs when there are only two possible choices, such as whether to receive the COVID-19 vaccine or vote on EU membership, which is commonly referred to as a *majority illusion*. In this work, we study the topological properties of social networks that lead to opinion illusions and focus on minimizing the number of agents that need to be influenced to eliminate these illusions. To do so, we propose an initial, but systematic study of the algorithmic behaviour of this problem.

We show that the problem is NP-hard even for underlying topologies that are rather restrictive, being planar and of bounded diameter. We then look for exact algorithms that scale well as the input grows (FPT). We argue the in-existence of such algorithms even when the number of vertices that must be influenced is bounded, or when the social network is arranged in a "path-like" fashion (has bounded pathwidth). On the positive side, we present an FPT algorithm for networks with "star-like" structure (bounded vertex cover number). Finally, we construct an FPT algorithm for "tree-like" networks (bounded treewidth) when the number of vertices that must be influenced is bounded. This algorithm is then used to provide a PTAS for planar graphs.

KEYWORDS

Social Networks, Parameterized Complexity, Approximation Algorithms

1 INTRODUCTION

Opinion illusion occurs for an agent in a social network when the perception within its immediate neighbourhood differs from the broader network's predominant opinion. In democratic societies, decisions often hinge on binary choices such as EU membership or the rightfulness of COVID-19 vaccination. Opinion illusion in situations with binary choices is called a *majority illusion*. Over time, an agent under illusion may alter its opinion if left unaddressed. This change further influences other agents to reconsider their opinions. This cascading effect leads to the spread of misinformation and bias in the community, an undesirable scenario. An exemplary visualisation of the problem is presented in Figure 1.
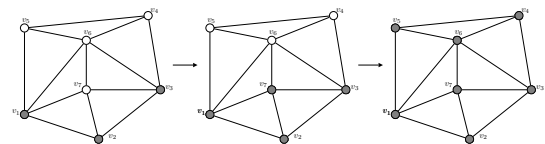

Figure 1: In the leftmost figure, the vertex v_7 is under illusion. It may change its colour over time and subsequently influence the colour of v_6 followed by v_5 and v_4. The final configuration is presented in the rightmost figure.

The process of false information spreading may create significant differences in the outcome of an election. For example, the outcome of the 2018 French and 2018 Italian elections was aligned with the information/misinformation spread from bots [1, 24]. Castiglioni et al. did a theoretical study of the problem of manipulating elections through social influence [10]. Wu et al. [41] showed that although controlling election is generally NP-hard, agents can be partitioned into similar groups and the problem becomes tractable. Faliszewski et al. [23] further showed FPT algorithms to influence the agents in a social network to obtain desirable election outcomes.

The spread of opinion in social networks and arriving at a consensus has been extensively studied earlier [2, 9, 23]. Auletta et al. [2] studied the complexity questions of finding a minimum-sized set of agents to maximise the spread of information. This problem is known under the name of *target set selection* [29]. Bredereck and Elkind considered the problem of maximising opinion diffusion under majority settings with the help of bribery (or influence) [9]. They further established a connection between their model and the target set selection problem. To counter misinformation spreading,

a tried and tested approach involves network intervention [38]. This is often achieved by appointing *champions*, typically opinion leaders or influencers [32]. This method has effectively driven change, notably in health behaviour modification [37, 39].

Lerman et al. [34] first considered majority illusion, examining social networks where such phenomena occur. A graph-theoretic study of different variants of majority illusion is reported in [40]. Grandi et al. [26] initiated the algorithmic study of the problem of verifying the existence of and eliminating majority illusion. They considered the problem where along with the social network a parameter $q \in [0, 1]$ is given, which denotes at least q-fraction of the vertices under an illusion in the social network. They showed the decision problem, that there is a labelling which induces a q-majority illusion, is NP-complete for every rational number $q \in (\frac{1}{2}, 1]$ even for planar graphs, bipartite graphs as well as graphs of bounded maximum degree. On the positive side, they showed FPT algorithms when parameterised by neighbourhood diversity, vertex cover, maximum degree with treewidth or cliquewidth. Furthermore, they proposed two editing operations, edge addition and edge deletion to eliminate majority illusion. From a theoretical point of view, it is rather natural to consider these two operations. However, one should also consider their associated cost, meant to model the effort required to create or remove connections between strangers or friends respectively.

We study the theoretical model of the majority illusion problem while considering the practicality of the proposed solution. A theoretical study of eliminating undesirable properties within social networks is not new. It has been extensively explored in recent years [4, 12]. We consider the network intervention method as suggested in [32, 38], where the objective is to find the minimum number of leaders or influencers. The task of the leader or influencer is to sync its opinion with the global majority opinion[1] and by doing so we remove all illusions of the network. Finding a smallest possible set of nodes/agents in a social network to create desirable influence is a well-established research interest and has been studied in the context of digital marketing [19, 20, 27, 36].

Related problem of independent theoretical interest. We begin this work by observing a strong relation (polynomial equivalence) between the ELIMININATING ILLUSION (EI for short) problem and a variation of the classic dominating set problem, known as TOTAL VECTOR DOMINATION (TVD for short). To the best of our knowledge, the TVD problem has only been considered in [14, 28]. We would like to stress here that none of our infeasibility results follow directly from the known results for the TVD problem. On the contrary, our reduction serves as a way to translate many of the efficient algorithm that can be conceived for EI, including the PTAS we provide, into their direct counterparts for the TVD problem. This last observation has particular importance in view of the sparsity of positive results that exist for the TVD problem.

Our Contribution. We begin by showing the aforementioned equivalence between the EI and TVD problems. We then focus on the EI problem. We show that this problem is NP-hard even on planar bipartite graphs. Moreover, the problem is W[2]-hard when

parameterised by the solution size, i.e., the minimum number of vertices that must be influenced. Both of the previous results hold even if we restrict the input graph to have bounded diameter. It is then natural to wonder about a possible efficient algorithm for solving the problem when considering structural parameters of the input graph. Unfortunately, we show that the problem is XNLP-hard when parameterised by the pathwidth of the input graph (implying the same result when parameterised by the treewidth). Nevertheless, we do provide an FPT algorithm parameterised by the vertex cover number of the input graph. Finally, we provide a PTAS for planar graphs. To achieve this we also construct an FPT algorithm parameterised by the treewidth of the input graph plus the solution size. This implies an XP algorithm parameterised just by the treewidth, which is then used as a building block for the aforementioned PTAS.

2 PRELIMINARIES

Formally, we consider social networks as graphs $G = (V, E)$ where each vertex has a labelling $f : V(G) \rightarrow \{0, 1\}$. We assume there are strictly more vertices with label 0 than 1 in G. We say that a vertex is under illusion if the label 1 has a surplus in its neighbourhood. That is, a vertex $v \in V$ *is under illusion by* f if $|\{u \in N(v) : f(u) = 1\}| > |\{u \in N(v) : f(u) = 0\}|$. We say that a labelling f' is a *solution* to the majority illusion problem on G if f' induces no illusion; the *size* of this solution is $|\{v \in V(G) : f(v) \neq f'(v)\}|$.

ELIMININATING ILLUSION
Input: A graph $G = (V, E)$, a labelling $f : V \rightarrow \{0, 1\}$ and an integer $k \geq 1$.
Task: Is there a labelling $f' : V \rightarrow \{0, 1\}$ such that f' induces no illusion and $|\{v \in V : f(v) \neq f'(v)\}| \leq k$.

We will follow the usual graph theory notation [18]. For any vertex v of a graph G, we denote by $d_G(v)$ degree of v in G, which is equal to the number of neighbours of v in G. Whenever it is obvious from the context, the subscript will be dropped.

Parameterised Complexity. The goal in the field of parameterised complexity is to construct exact algorithms that are efficient with respect to a measure of time that is extended by a secondary measure of the problem, commonly referred to as the *parameter*. Let n denote the size of the input of a problem, k denote the considered parameter and f be an arbitrary computable function. We consider that a parameterised problem is solved efficiently if it can be determined in $f(k) \cdot n^{O(1)}$ time. In such cases, we say that the problem is *fixed-parameter tractable* and that it belongs to the class FPT. A parameterised problem is *slicewise polynomial* if it can be determined in $n^{f(k)}$ time. In such cases, we say that the problem belongs to the class XP. It should be noted that, unlike NP-complete problems, there is actually a whole hierarchy of infeasibility for parameterised problems, referred to as the W hierarchy. A problem is presumably not in FPT if there exists a $t \geq 1$ such that the problem is W[t]-hard (by a parameterised reduction). Moreover, it is hypothesised that W[t] $\subseteq$ W[$t + 1$] for every $t \geq 1$. Finally, we will need the definition of the recently introduced XNLP class [8]. This class contains the parameterised problems whose input can be encoded with n bits and can be solved non-deterministically in time $f(k) \cdot n^{O(1)}$ and

---

[1]A similar solution has recently been proposed independently by Chitnis and Kumar [11].

space $f(k) \log n$. We refer the interested reader to now classical monographs [15, 22, 25, 35] for a more comprehensive introduction to the topic of parameterised complexity.

Structural Parameters. Let $G = (V, E)$ be a graph. A set $U \subseteq V$ is a *vertex cover* if for every edge $e \in E$ it holds that $U \cap e \neq \emptyset$. The *vertex cover number* of G, denoted $\text{vc}(G)$, is the minimum size of a vertex cover of G.

A *tree decomposition* $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ of G is a tree T, such that the following hold:

- Every node $t \in V(T)$ has an associated bag $X_t \subseteq V$ such that the union of all bags is equal to $V(G)$.
- For each edge $\{u, v\} \in E(G)$, there has to exist at least one bag X_t with $u, v \in X_t$.
- For each vertex $v \in V(G)$, the nodes whose bags contain v induce a connected subtree of T.

The *width* of a tree decomposition is $\max\{|X_t| \mid t \in V(T)\} - 1$. The *treewidth* $\text{tw}(G)$ of a graph G is the smallest value, such that there exists a tree decomposition of G with this width.

It is known that computing a tree decomposition of minimum width is fixed-parameter tractable when parameterised by the treewidth [5, 30], and even more efficient algorithms exist for obtaining near-optimal tree decompositions [31].

A tree decomposition $(T, \{X_t\}_{t \in V(T)})$ is *nice* [6] if T is rooted in $r \in V(T)$ and every node $t \in V(T)$ is exactly of one of the following four types:

1. **Leaf:** t is a leaf of T and $|X_t| = 1$.
2. **Introduce:** t has a unique child i and there exists $v \in V$ such that $X_t = X_i \cup \{v\}$.
3. **Forget:** t has a unique child i and there exists $v \in V$ such that $X_i = X_t \cup \{v\}$.
4. **Join:** t has exactly two children i, j and $X_t = X_i = X_j$.

It is well known that every graph $G = (V, E)$ admits a nice tree decomposition rooted in $r \in V(T)$, that has width equal to $\text{tw}(G)$, $|V(T)| = O(|V|)$ and $X_r = \{\emptyset\}$ [6].

The notions of (nice) *path decomposition* and *pathwidth* are defined analogously, by replacing the third item in the definition of a tree decomposition by the following: for every vertex $v \in V$, the nodes whose bags contain v induce a connected *subpath* of $\mathcal{T}$. Finally, nice path decompositions do not contain any join nodes.

Approximation. The goal of an approximation algorithm is to obtain an approximated solution of an intractable problem in polynomial time. Formally speaking, given a minimisation problem $\mathcal{P}$, a polynomial time algorithm A is an approximation algorithm with an approximation ratio $\alpha \in \mathbb{R}$ if for all instances $I \in \mathcal{P}$, A produces a feasible solution $\text{SOL}(I)$ such that $|\text{SOL}(I)| \leq \alpha \cdot |\text{OPT}(I)|$, where $\text{OPT}(I)$ is the optimum solution of I. A PTAS for a minimisation problem is an approximation algorithm which for every $\varepsilon > 0$ outputs a solution of size $(1 + \varepsilon)|\text{OPT}|$ in time polynomial in the size of the input.

3 CONNECTION TO TOTAL VECTOR DOMINATION

In this section, we will establish a polynomial-time equivalence between the TVD and Elimininating Illusion problems. This

equivalence allows us to, to some extent, interchange results and complexity analyses between the two problems. We begin by formally stating the definition of the TVD problem:

Total Vector Domination

Input: A graph $G = (V, E)$ and a vector $(k(v) : v \in V)$ where $k(v) \in \{0, 1, \ldots, d(v)\}$ for all $v \in V$

Task: Find a minimum-size set $S \subseteq V$ such that $|S \cap N(v)| \geq k(v)$ for all $v \in V$.

From EI to TVD

Given an EI instance (G, f), we will construct a TVD instance (G', k). The main obstacle we have to overcome is that there are some vertices already labelled as 0 in (G, f) that should influence the TVD solution, but are not in the solution of EI. We construct the graph $G' = (V', E')$ in a specific way to combat this limitation. We start with a copy of $G = (V, E)$. Then, for each vertex $v \in V$ with $f(v) = 0$, we attach a leaf w to the vertex v. This finishes the construction of G'. We then define $k(v)$ for all $v \in V'$ as follows:

$$k(v) = \begin{cases} \lceil \frac{d_G(v)}{2} \rceil & \text{if } v \in V \\ 1 & \text{if } v \in V' \setminus V \end{cases}$$

This construction ensures that vertices labelled 0 in G are selected in the TVD solution of G', as we force the leaves of G' to have $k(w) = 1$. Moreover, any TVD solution eliminates all the illusions in the original graph, as we set $k(v)$ to be at least half of the nodes for all vertices $v \in V$. It is important to note that the solution given by the TVD problem will always choose vertices of G as they are at least as good as the new vertices added. This leads us to the following lemma.

Lemma 1. *A solution to the TVD problem on (G', k) corresponds to a solution of the EI problem on (G, f), excluding vertices of G originally labelled 0.*

Observe that G' is essentially the same as G, with some additional leaves. Thus, we obtain the following corollaries that follow directly from results in [13, 28]:

Corollary 1. *There is a polynomial time algorithm for EI on trees.*

Corollary 2. *There is an* FPT *algorithm for EI on planar graphs.*

From TVD to EI

We now present the reverse reduction, transforming a TVD instance into an equivalent Elimininating Illusion instance.

Given a TVD instance (G, k), we construct an EI instance (G', f) as follows. We start with $G' = (V', E')$ being a copy of $G = (V, E)$, with one copy of the P_2 path attached to each $v \in V$. We say that the vertices that are common between G and G' (i.e., $v \in V \setminus V'$) are the *original* vertices, while any other vertex is *auxiliary*. We label all the original vertices of G' by 1 and its auxiliary vertices by 0. Then, for each original vertex v of G', we define the *budget* of v to be the value $B(v) = 2k(v) + 1 - d_G(v)$. Observe that a vertex might have a negative budget. For each original vertex $v \in V'$ such that $B(v) \geq 0$, we add $B(v)$ copies of the P_3 path to G', attached to v, labelled by the labels $1, 0, 0$, with the newly added vertex labelled 1
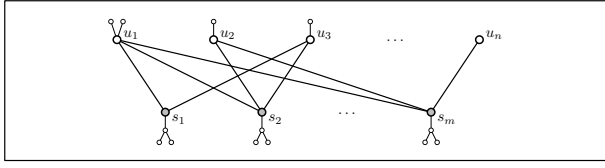
Figure 2: The graph G_I constructed in the proof of Theorem 1. The white vertices have label 0, and the grey vertices have label 1.

Figure 3: The graph G_ϕ constructed in the proof of Theorem 2. The white vertices have label 0, and the grey vertices have label 1. The clauses are $c_1 = (X_1 \vee X_2 \vee X_3)$ and $c_2 = (\overline{X_2} \vee \overline{X_3} \vee \overline{X_k})$.

being the neighbour of v. Then, for each original vertex $v \in V'$ such that $B(v) < 0$, we add $|B(v)|$ copies of the P_2 path to G', attached to v and labelled by the labels $0, 0$. This finishes the construction of the instance (G', k).

Notice that the above construction ensures that label 0 is indeed the majority. Moreover, we may assume that any optimal solution of EI on (G', f) only relabels a subset of the original vertices of G'. Indeed, if an optimal solution contains an auxiliary vertex u belonging to some path attached to an original vertex v, then it suffices to relabel any original neighbour of v instead of u. Finally, the choices of f and $B(v)$, for each original vertex v, are such that all the original vertices of G' are under illusion, and this can only be corrected if at least $k(v)$ neighbours of v are relabelled. From the above observations we obtain the following:

Lemma 2. *A solution to the EI problem on (G', f) corresponds to a solution of the TVD problem on (G, k), restricted to the original vertices of G'.*

4 THE PROBLEM IS HARD

In this section we provide two reductions, both showing hardness under very restricted properties.

The first reduction is from the Set Cover problem. Given a connected bipartite graph $I = (S \cup U, E)$ the Set Cover problem asks for a smallest *cover* C of U, i.e., a minimum size subset C of S such that every vertex in U is adjacent to at least one vertex in C. This problem is known to be W[2]-hard when parameterised by the size of C [21].

Theorem 1. *EI is NP-hard, as well as W[2]-hard when parameterised by the solution size, even if the input graph is bipartite and has bounded diameters.*

Proof. Let $I = (S \cup U, E)$ be an instance of Set Cover. We assume that $|S|, |U| > 0$ and that there is a $u \in U$ adjacent to all $s \in S$. We construct the graph G_I as follows. We start from the graph I. For each $u \in U$, we add $d_I(u) - 1$ leaves attached to u, where $d_I(u)$ is the degree of u in the graph I. For each $s \in S$ we attach a single vertex, with two leaves, to s. Hence, the diameter of the graph is at most 6. To ease the exposition, we will refer to the vertices of $V(G_I) \cap S$ and $V(G_I) \cap U$ as the vertices of S and U respectively. We assign $f(v) = 0$ for all $v \notin S$ and $f(v) = 1$ for all $v \in S$. By the construction of G_I it follows that 0 is the strict majority. Moreover, only the vertices of U are under illusion. Furthermore, for each $u \in U$, it is sufficient that one of its neighbours belonging in S changes its labelling to 0 in order for u to no longer be under illusion. An exemplary visualisation of G_I can be seen in Fig. 2.
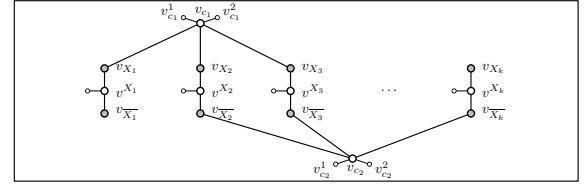
We will show that I has a set cover $C \subseteq S$ of order at most k if and only if there is a solution f' of G_I of size at most k.

Let $C \subset S$ be a covering of U of size k in I. For every $s \in C$ we set $f'(s) = 0$ in G_I. For all other vertices v in G_I we set $f'(v) = f(v)$. Therefore, f and f' differ in exactly k vertices. Further, since C is a covering of U, each $u \in U$ is adjacent to at least one $s \in C$ whose label was changed to 0. Therefore, no vertex in G_I is under illusion in f'. □

The second hardness reduction is from the Planar Monotone 3-SAT problem, a restricted variant of 3-SAT. In this variant, each clause consists exclusively of either positive or negative literals. Moreover, the graph admits a straight-line drawing in which all variables lie on a horizontal line. In this representation, every positive (negative) clause is positioned in the upper (lower) half-plane. This problem is well known to be NP-complete [17]. Similar to the proof idea of Theorem 3 in [16], we can further assume that each literal appears at most three times.

Theorem 2. *EI is NP-hard, even if the input graph G is restricted to be bipartite, planar and of maximum degree 5.*

Proof. Given a formula ϕ that is an instance of Planar Monotone 3-SAT where each literal appears in at most three clauses, we construct an instance G_ϕ of EI. For every variable X, we add the variable-vertices $v_X, v_{\overline{X}}$. For every clause $c \in \phi$, we add a clause-vertex v_c. For every clause-vertex v_c, we add the edge $v_c v_X$ ($v_c v_{\overline{X}}$ resp.) for every literal X ($\overline{X}$ resp.) that appears in c. Further, for every pair of literal vertices $v_X, v_{\overline{X}}$, we add a new vertex v^X, referred to as the *controller* of X, as well as the edges $v^X v_X$ and $v^X v_{\overline{X}}$. For every clause-vertex v_c, we add two leaves v_c^1 and v_c^2. Finally, we add a leaf attached to the controller v^X of every literal X. Let G_ϕ be the resulting graph. It is straightforward to see that G_ϕ is a planar bipartite graph. Moreover, observe that, in G_ϕ, the degree of every vertex is at most 5. Thus, G_ϕ is bipartite, planar and of maximum degree 5. For every vertex $v = v_X$ or $v = v_{\overline{X}}$, for every literal X, we assign $f(v) = 1$. For every other vertex v of G_ϕ, we assign $f(v) = 0$. Notice that 0 is the strict majority, and that the vertices under illusion are precisely the clause-vertices v_c and the controllers v^X. Each of them requires at least one of their neighbours to change its label from 1 to 0 in order to not be under illusion. A visualisation of G_ϕ can be seen in Fig 3.

Let k be the number of variables in ϕ. We will show that there exists a solution f' for G_ϕ of size at most k if and only if ϕ is satisfiable.

If ϕ is satisfiable, there exists a truth assignment t from the variables to {True, False}. For every variable X, we set $f'(v_X) = 0, f'(v_{\overline{X}}) = 1$ if $t(X) = $ True, and $f'(v_X) = 1, f'(v_{\overline{X}}) = 0$ if $t(X) = $ False. For all other vertices v in G_ϕ, we set $f'(v) = f(v) = 0$. In this way, f and f' differ on precisely k vertices. For every X, the controller v^X is not under illusion in f' since the label of precisely one of $v_X, v_{\overline{X}}$ was changed. For every clause c, we know that c was satisfied by t. Hence, there is at least one variable-vertex that neighbours v_c whose label was changed and c is not under illusion in f'. These were the only vertices under illusion in f, thus f' is a solution for G_ϕ.

Conversely, assume there exists a solution f' for G_ϕ of size exactly k. Since no vertex is under illusion under f', for every variable X, at least one neighbour v_X or $v_{\overline{X}}$ of v^X was relabelled. Actually, it is exactly one of the vertices v_X or $v_{\overline{X}}$ that was relabelled, since there are exactly k variables. Since f' is a valid solution, every clause is adjacent to at least one vertex that was relabelled. Thus, the truth assignment that sets X to True (False resp.) for each variable X such that $f'(v_X) = 0$ ($f'(v_{\overline{X}}) = 0$ resp.), yields a satisfying truth assignment for ϕ. □

5 STRUCTURAL PARAMETERS

Theorem 3. *EI is solvable in* FPT *time parameterised by the vertex cover number of the input graph.*

Proof. Let $G = (V, E)$ be the input graph and f be the initial labelling of G. Moreover, let vc be the vertex cover number of G and let $U \subseteq V$ be a vertex cover of G of minimum size. Recall that $I = V \setminus U$ is an independent set of G. Since $|U| \leq$ vc, we may guess an optimal labelling $f'|_U$ of G such that no vertex of I is under illusion by $f'|_U$ (there are at most 2^{vc} such labellings). The labelling $f'|_U$ is optimal in the sense that it differs from f on a minimum number of vertices. All that remains to be done is to extend $f'|_U$ into an optimal solution f' of G by making sure that f' induces no illusion on the vertices of U.

To achieve this, we first arrange the vertices of I into sets according to their neighbourhoods in U. In particular, we partition I into the sets $I_1, \ldots, I_p$, for $p \leq 2^{\mathrm{vc}}$, such that for every $i \in [p]$, the vertices of I_i are twins, *i.e.*, they have the same neighbourhood. Formally, for every $u, v \in I$, we have that $u \in I_i$ and $v \in I_i$, for some $i \in [p]$, if and only if $N_G(u) = N_G(v)$. Then we compute the exact number of vertices of I_i, for each $i \in [p]$, whose label must be changed in order for the resulting labelling f' to be a solution of EI by modelling this problem as an ILP on bounded number of variables.

Variables

$x_i \quad i \in [p] \quad$ number of vertices of I_i labelled 1 by f'

Constants

$a(i) \quad i \in [p] \quad$ number of vertices of I_i labelled 1 by f

$n(u) \quad u \in U \quad$ number of neighbours of u in U labelled 1 by $f'|_U$

$d(u) \quad u \in U \quad$ degree of u in G

$i(u, i) \quad u \in U, i \in [p] \quad$ set to 0 if $u \notin N(v), \forall v \in I_i$, and to 1 otherwise

Objective

$$\max \sum_{i \in [p]} x_i \qquad (1)$$

Constraints

$$n(u) + i(u, i) \cdot x_i \leq d(u)/2 \qquad \forall i \in [p], u \in U \qquad (2)$$

$$x_i \leq a(i) \qquad i \in [p] \qquad (3)$$

The variable x_i represents the number of vertices of I_i that were labelled 1 by f and whose label will remain unchanged in f'. Constraint 3 makes sure that there are not more vertices in I_i labelled 1 by f' than they were by f. Constraint 2 ensures that no vertex $u \in U$ is under illusion by f'. Since the model has $p \leq 2^{\mathrm{vc}}$ variables, we can and obtain the x_is' in FPT time, parameterised by vc (by running the Lenstra algorithm [33]). Finally, note that the x_is' are enough to compute a solution f'. Indeed, it is sufficient to change the label of any $|I_i| - x_i$ vertices of I_i, for each $i \in [p]$, from 1 to 0 in order to extend $f'|_U$ into f'. This is immediate by the definition of the I_is'. □

From Theorem 1 we already know that EI is W[2]-hard if parameterised by the solution size. However, it is in XP parameterised by the solution size k, since trying all possible solutions of size at most k is bounded by $O(|V|^k)$. Looking for other promising parameters, and in view of Corollary 1, one could hope for the existence of an FPT algorithm for EI parameterised by the treewidth of the input graph. In the following theorem, we show that this is highly unlikely. We provide a reduction from the Minimum Maximum Outdegree (MMO for short) problem. In this problem, we are given a graph $G = (V, E)$, an edge weighting $w \colon E \to \mathbb{N}$ and a bound $R \in \mathbb{N}$ (both w and R are given in unary). The question is to find an edge orientation E' of E, such that for every $v \in V$, the weighted outdegree of v in $G' = (V, E')$ is at most R. It was recently shown that MMO is XNLP-hard when parameterised by the pathwidth of the input graph [7]. This means that MMO is W[t]-hard for every $t \in \mathbb{Z}_{\geq 1}$. It follows from our reduction that:

Theorem 4. *EI is* XNLP-*hard parameterised by the pathwidth of the input graph.*

Proof. Let x_1, x_2 be two vertices with label 1. A *switch structure* between the vertices x_1 and x_2 is a vertex y with label 0 that is incident to both of them and has an additional leaf attached to y labelled 0. Notice that y is under illusion, and in order for this to change at least one of x_1, x_2 must be relabelled. A *chessboard* of size k is a graph with $4k$ vertices labelled 1, arranged on a $2k \times 2$ grid with a switch structure between all vertex pairs having a Manhattan distance of 1 on the grid. We denote by *grid vertices* the vertices of a chessboard that also belong to the underlying grid. Because of the switch structures, for every pair u, v of adjacent grid vertices, either u or v (or both) must be relabelled to eliminate all illusions in the chessboard. Therefore, a chessboard of size k requires at least $2k$ label changes. Further, $2k$ label changes are enough if and only if all relabelled vertices have an even distance to each other on the underlying grid. Note that the pathwidth of a chessboard is constant, since traversing the grid from left to right yields a decomposition with bags of constant size.

Now let $G = (V, E), w, R$ be an instance of MMO. Slightly abusing the notation, we denote the weighted degree of v in G by $d_G(v)$,
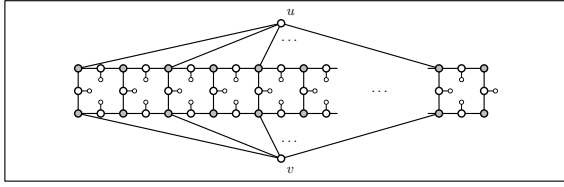
Figure 4: A chessboard structure between vertices u and v, used in the proof of Theorem 4. The grey vertices have label 1, and the others label 0.

for every $v \in V$. If E' is an edge orientation of E and $G' = (V, E')$ is the corresponding graph, we denote by $d^-_{G'}(v)$ ($d^+_{G'}(v)$ resp.) the weighted indegree (outdegree resp.) of v in G'. Further, we denote by W the sum of edge weights in G. Observe that for every edge orientation E', and for any $v \in V$, we have that $d^+_{G'}(v) \le R$ if and only if $d^-_{G'}(v) \ge d_G(v) - R$.

We construct a new graph G^* with vertex labels in the following way. First, we add all vertices in V to G^* and label them with 0. For each $e = uv \in E$, we replace e with a chessboard C_e of size $w(e)$. For $i \in [2]$ and $j \in [2k]$, let $z_{i,j}$ be the grid vertex of C_e that lies on the ith row and j^{th} column in the underlying grid of C_e. Then, we connect u to $z_{1,1}, z_{1,3}, \ldots, z_{1,2k-1}$ and v to $z_{2,1}, z_{2,3}, \ldots, z_{2,2k-1}$. This construction is illustrated in Fig. 4. For every $v \in V(G)$, we add leaves attached to v such that the demand of v, i.e., difference between $|\{u \in N(v) : f(u) = 1\}|$ and $|\{u \in N(v) : f(u) = 0\}|$, in G^* becomes precisely $d_G(v) - R$. This can be achieved by adding leaves labelled 1 (0 resp.) as long as the demand of v is smaller (larger resp.) than $d_G(v) - R$. Since every added leaf can change the demand by at most one, and two leaves of the same type change the demand of v by at least two, this procedure can be applied to set an arbitrary demand for v. Adding the chessboards and the leaves increases the pathwidth of the resulting graph only by a constant. Further, the number of vertices we added is linear in $W + R$. Since w and R were given in unary, the size of G^* is polynomial in (G, w, R).

We will now show that there exists a solution for EI in G^*, labelled as described above, of size $2W$ if and only if there exists a solution for MMO for $G = (V, E), w, R$.
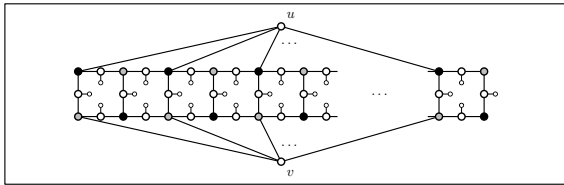

Figure 5: The same structure as in Fig. 4. The grey vertices were relabelled with label 0. This corresponds to the edge $uv \in E(G)$ being directed from u to v in G^*. The only other valid relabelling with size $2w(u, v)$ is the reversed relabelling.

Assume there exists a solution E' for MMO in $G = (V, E), w, R$. For every $(u, v) \in E'$, we relabel the vertices of C_{uv} that are adjacent to v. Then, we also relabel the grid vertices of $C(uv)$ that are not adjacent to u and were not relabelled in the previous step. In this manner, we relabelled precisely $2w(u, v)$ of the chessboard vertices

that replaced the edge uv. In total, we relabel $2W$ vertices. See Fig. 5 for such a relabelling. We claim that after this relabelling, no vertex in G^* remains under illusion. We have already shown that the relabelling of the chessboards satisfies every internal vertex of the chessboard. Further, all leaves that were added to adjust the demand for the original vertices have only one neighbour with the label 0, therefore they were not under illusion from the start. Thus, the only vertices left to verify, are those that came from $V(G)$. Let us consider such a vertex v. Since E' is a valid solution of MMO, we have that $d^-_{G'}(v) \ge d_G(v) - R$. Therefore, v has at least $d(v) - R$ relabelled vertices as neighbours, and these relabelled neighbours are precisely those chessboard vertices that correspond to incoming edges of v in G'. Since this was the demand of v in G^*, it follows that v is not under illusion in the relabelled graph.

Next, assume that there exists a solution f^* of G^* of size $2W$. For every edge $e \in E$, we have that C_e requires at least $2w(e)$ relabellings. Since the total sum of sizes among all included chessboards is W, this means that every chessboard is labelled optimally. Therefore, for every edge $e = uv \in E$, either all neighbours of u in C_{uv}, or all neighbours of v in C_{uv}, are relabelled. We add (u, v) to E' if all neighbours of v are relabelled, otherwise, we add (v, u) to E'. Then, we set $G' = (V, E')$. Now, for every $v \in V(G')$, we have that $d^-_{G'}(v)$ is equal to the number of relabelled neighbours of v, which is at least $d_G(v) - R$. Equivalently, $d^+_{G'}(v) \le R$ for every $v \in V(G')$, and the edge orientation E' is a valid solution for G, w, R. $\square$

6 A PTAS FOR PLANAR GRAPHS

In this section, we use the classic layering technique introduced by Baker [3] for designing approximation algorithms on planar graphs. On a high level, we break the input graph into layers to solve the problem optimally in each layer. Then we take a union of these solutions to return a feasible solution for the original input. Our algorithm computes this solution on several *layered decompositions* and returns the minimum among them. In order to solve the problem in each layer, we run the XP algorithm (parameterised by the treewidth) that solves the following generalisation of the EI problem, which follows from the upcoming Theorem 5.

SUBSET ELIMININATING ILLUSION
Input: A graph $G = (V, E)$, a set $S \subseteq V$, a labelling $f : V \to \{0, 1\}$ and an integer $k \ge 1$.
Task: Is there a labelling $f' : V \to \{0, 1\}$ such that f' induces no illusion on the vertices of S and $|\{v \in V : f(v) \ne f'(v)\}| \le k$.

Before we proceed with the next theorem, allow us to introduce some additional notation. We denote the set $\{0, \ldots k\}$ by $[k]$. When considering the SUBSET ELIMININATING ILLUSION problem, let $\mu : S \to [M]$ be a function that is equal to the minimum number of vertices adjacent to v that must be relabelled such that v is not under illusion any more, for every vertex $v \in S$. We will also extend this function such that $\mu(v) = 0$ for every $v \in V \setminus S$. We say that μ is the *demand function* and denote M as the *maximum demand*.

THEOREM 5. *There exists an* FPT *algorithm for* SUBSET ELIMIN-INATING ILLUSION *parameterised by the treewidth* $\text{tw}(G)$ *of the input graph and the maximum demand M, with running time* $M^{O(\text{tw}(G))} \cdot n^{O(1)}$.

Proof. Let (G, S, f) be an instance of Subset Elimininating Illusion and let $\mathcal{T} = (T, \{X_t\}_{t \in V(T)})$ be a nice tree decomposition of G, with T being rooted at a leaf $r \in V(T)$. For some $t \in V(T)$, we define the subgraph G_t of G to be the graph induced by the union of bags contained in the subtree of T rooted at t. For instance, the induced graph G_r with respect to the subtree rooted at the root node r is precisely G. We use dynamic programming on $\mathcal{T}$ to find the minimum number of vertices in $V(G)$ whose label is required to change, such that no vertex in S remains under illusion. Before we describe how the dynamic step is performed on each type of node in the tree decomposition, we introduce some notation. We use vectors $D_t \in [M]^{|X_t|}$ to express the remaining demand for vertices in X_t. In particular, for every $v \in X_t \cap S$, the value of $D_t[v]$ is the number of vertices that still need to be relabelled in the neighbourhood of v in order to bring v out of illusion. Also, for every $v \in X_t \setminus S$, we set $D_t[v] = 0$. We also define some operations on these vectors. We denote by $D_t + \delta_v$ the operation where for each $w \in (N(v) \cap X_t) \cap S$, the entry of $D_t[w]$ is increased by one and for each $w \in (N(v) \cap X_t) \setminus S$, the entry of $D_t[w]$ remains unchanged. Furthermore, we write $D_t|_{D_i}$ for the restriction of the vector D_t to D_i. That is, $D_t|_{D_i}$ is a vector whose entries for the vertices $X_t \cap X_i$ is the same as D_t. Finally, we use $D^{v=a}$ to represent a vector with an entry $D[v]$ equal to a.

We define $W[t, D_t, U]$ to be the minimum number of relabellings such that:

(1) the labels of vertices in $U \subseteq X_t$ are relabelled,
(2) the labels of vertices in $X_t \setminus U$ are not relabelled,
(3) no vertex of $(V(G_t) \cap S) \setminus X_t$ is under illusion and
(4) vertices $v \in X_t$ have a remaining demand of at most $D_t[v]$.

From this definition, it follows that the minimum number of relabellings required is exactly $W[r, \emptyset, \emptyset]$. We now proceed to describe the recursive formulas for the different node types of $\mathcal{T}$.

Leaf node. A leaf node $\ell \neq r$ corresponds to an empty graph. Thus $W[\ell, D_\ell, U] = 0$, as no vertex can be under illusion. Therefore, leaf nodes serve as our base case.

Introduce node. Let v be the vertex that has been introduced in node t and let node i be the only child of t. If $D_t[v] < \mu(v) - |N(v) \cap U|$, we set $W[t, D_t, U] = \infty$ since this violates property 4 as by definition of $\mathcal{T}$ the neighbourhood of v that has been introduced so far must be contained in X_t. Otherwise, we distinguish between two cases. If $v \in U$, then relabelling v decreases the remaining demand for all of its adjacent vertices. Thus, v relaxes the remaining demand vector D_t by one for all vertices in the neighbourhood of v introduced so far. We can therefore calculate $W[t, D_t, U]$ with the help of the child i and the relaxed vector: $W[t, D_t, U] = W[i, D_t|_{D_i} + \delta_v, U \setminus \{v\}] + 1$. If $v \notin U$, we can look up the required number of relabelling in the child node, since in this case $U \subseteq X_i$ by the definition of a nice tree decomposition: $W[t, D_t, U] = W[i, D_t|_{D_i}, U]$. To process a specific node of type introduced, we need to consider each possible vector D_t and each possible subset $U \subseteq X_t$. For a given D_t and U we can look up the solution in polynomial time in the size of the bag X_t. This yields an overall runtime of $O((M + 1)^{|X_t|} \cdot 2^{|X_t|} \cdot |X_t|^c) \subseteq O(\text{tw}(G)^c \cdot (M+1)^{\text{tw}(G)} \cdot 2^{\text{tw}(G)})$, where c is a constant, to process a node of type introduce.

Forget node. Let v be the vertex that has been forgotten at node t with child node i. Since v is removed from X_t, we need to guarantee that property 3 holds, i.e., the remaining demand of v must be 0. If $v \notin S$, then we are done, as in this case $\mu(v) = D_t[v] = 0$ by definition. So, we may assume that $v \in S$. For a vector $D_t^{v=0}$, the vertex v can either be in the set of vertices U that are relabelled or not. We take the minimum of all these cases: $W[t, D_t, U] = \min\{W[i, D_t^{v=0}, U], W[i, D_t^{v=0}, U \cup \{v\}]\}$. The processing time of a node of type forget is analogous to the processing time of a node of type introduce, resulting in the same runtime of $O(\text{tw}(G)^c \cdot (M + 1)^{\text{tw}(G)} \cdot 2^{\text{tw}(G)})$, where c is a constant.

Join node. Let t be a join node with children i and j. Node t merges two previously disjoint subgraphs G_i and G_j. Consider a vertex $v \in X_t = X_i = X_j$. The vector $D_t[v]$ describes the remaining demand of v by the property 4. Thus, at least $\mu(v) - D_t[v]$ vertices have been relabelled in the neighbourhood of v. These relabelled vertices in the neighbourhood of v can either be already forgotten in G_i or G_j or still remain in $U \subseteq X_t$. Therefore, $\mu(v) - D_t[v] = (\mu(v) - D_i[v]) + (\mu(v) - D_j[v]) - |N(v) \cap U|$ from which follows that $D_i[v] + D_j[v] = D_t[v] + \mu(v) - |N(v) \cap U|$, where we need to subtract $|N(v) \cap U|$ since these vertices are accounted for in $\mu(v) - D_i[v]$ and $\mu(v) - D_j[v]$. To determine the minimum number of relabellings $W[t, D_t, U]$, we must identify combinations C_v of values (a_i, b_j) satisfying $a_i + b_j = D_t[v] + \mu(v) - |N(v) \cap U|$ for each vertex $v \in X_t$. These pairs represent distinct valid pairings of vectors D_i and D_j for the specific entry v. For every valid combination in C_v, we can pair it with all other valid combinations C_u of vertices $u \in X_t \setminus \{v\}$ to generate vector pairs $(D_i', D_j') \in C(D_t)$ that are valid for all vertices $v \in X_t$. Using C_t, we can then calculate $W[t, D_t, U]$ with the following equation: $W[t, D_t, U] = \min_{(D_i', D_j') \in C(D_t)}\{W[i, D_i', U] + W[j, D_j', U] - |U|\}$. To compute a join node, we need to calculate C_v for each $v \in X_t$ for a given D_t and U. This can be done in $O(M)$ time. To construct a single vector pair in $C(D_t)$, we need $O(|X_t|)$ time. In total, we have $|C(D_t)| \in O((M + 1)^{|X_t|})$ many valid vectors that need to be checked, yielding a processing time of $O((M+1)^{|X_t|} \cdot 2^{|X_t|} \cdot (M + 1)^{|X_t|} \cdot M \cdot |X_t|) \subseteq O((M+1)^{2\,\text{tw}(G)+1} \cdot 2^{\text{tw}(G)} \cdot \text{tw}(G)^c)$, for some constant c.

The number of nodes in $V(T)$ is in $O(\text{tw}(G) \cdot n)$, where $n = |V(G)|$. So the runtime of our algorithm is $O((M + 1)^{2\,\text{tw}(G)+1} \cdot 2^{\text{tw}(G)} \cdot \text{tw}(G)^c \cdot n)$, for some constant c. □

Since the maximum demand M is a lower bound for the solution size k, which is upper bounded by the number of vertices n in the input graph, we obtain the following.

Corollary 3. *There exists an* FPT *algorithm for* Subset Eliminating Illusion *parameterised by the treewidth* tw *of the input graph plus the solution size. This implies an* XP *algorithm for the same problem parameterised just by* tw.

We are now ready to present the PTAS for planar graphs. A description of our algorithm is given in Algorithm 1.

Theorem 6. *For any given $\varepsilon > 0$, there exists an approximation algorithm which computes the solution of the EI problem on a planar graph in $O(n^{\frac{1}{\varepsilon}})$ time and computes a solution which is $(1 + \varepsilon)$-times the optimum.*

Algorithm 1: PTAS on planar graphs

Input: G, f, ε
$k = 4/\varepsilon$;
Find the outerplanar layers of the vertices;
for $i = 0, \ldots, k - 1$ **do**
    Find the components $G_1^i, G_2^i, \ldots,$ of G, where G_j^i
    contains vertices on the layers from $i + j(k + 1)$ to
    $i + (j + 1)(k + 1) + 1$;
    for $j = 0, 1, \ldots,$ **do**
        Compute A_j^i, the solution of SUBSET ELIMININATING
        ILLUSION on G_j^i setting $S = \overline{G}_j^i$ where $\overline{G}_j^i$ contains
        vertices on the layers from $i + j(k + 1) + 1$ to
        $i + (j + 1)(k + 1)$;
        $A^i = \cup_j A_j^i$;
    end
end
Let A be the minimum solution among $\{A^0, A^1, \ldots, A^{k-1}\}$;
return A

PROOF. For given $\varepsilon > 0$, let k be the nearest positive integer of $\frac{4}{\varepsilon}$. Consider a planar embedding of the input graph $G = (V, E)$. We assign a layer to each vertex according to its depth from the outerface, starting from layer zero for the vertices on the outerface. Clearly, the vertices of layer ℓ are on the outerface of the residual graph obtained after deleting the vertices of the layers $\{0, 1, 2, \ldots, \ell - 1\}$. Clearly, each G_j^i $\underline{G}_j^i$, as defined in Algorithm 1, is at most a k-outerplanar graph. That is, for each vertex v of G_j^i, there exists a sequence of at most k consecutive layers of G such that v belongs in the kth such layer. It is known that every k-outerplanar graph has treewidth bounded by $3k - 1$ [6]. It follows from Corollary 3 that there exists an algorithm which runs in $n^{O(k)} \cdot n^{O(1)}$ time and solves SUBSET ELIMININATING ILLUSION optimally on G_j^i for each i, j considering $S = \overline{G}_j^i$ as defined in Algorithm 1. For every i, we join the solutions we computed for the G_j^i's. We return the solution which is minimum among all i's. Hence, the running time of this algorithm is $O(n^{\frac{1}{\varepsilon}})$.

Let A_j^i be the solution of the problem on G_j^i for each i, j using Corollary 3. Let $\underline{G}_j^i \subset G_j^i$ be the induced subgraph on the vertices of layers $i + j(k + 1) + 2$ to $i + (j + 1)(k + 1) - 1$. Let $\overline{A}_j^i = A_j^i \cap \underline{G}_j^i$. Let S be the minimum set of vertices that must be relabelled to eliminate all the illusions in G. Let $\overline{S}_j^i = S \cap \underline{G}_j^i$. Clearly, $\overline{A}_j^i = \overline{S}_j^i$. Let $V_j^i \subset V$ be the vertices on layers $i + j(k + 1)$ and $i + j(k + 1) + 1$. Let $S_j^i = S \cap V_j^i$. Let $S^i = \bigcup_j S_j^i$. Observe that $\overline{S}_j^i$ and S^i are disjoint and $\sum_i^k |S^i| = 2|S|$. Hence, there exists one i^* such that $|S^{i^*}| \leq 2|S|/k$. Let $A = \bigcup_j A_j^{i^*}$ be the union of the solution of the G_j^i's. Then, A is a potential candidate set of vertices for relabelling for the graph G, as for a fixed i we get that the G_j^i's cover the entire graph G. Let, $A_j = A \cap V_j^{i^*}$ be the output of the algorithm restricted on the vertices on layers $i^* + j(k + 1)$ and $i^* + j(k + 1) + 1$.

We claim that for every j, we have that $|A_j| \leq 2|S_j^{i^*}|$. Indeed, the set $S_j^{i^*}$ contains the subset of the optimum solution restricted to the vertices on layers $i^* + j(k + 1)$ and $i^* + j(k + 1) + 1$. Hence, $S_j^{i^*}$ can be seen as the union of two sets $\bar{S}_j^{i^*}$ and $\bar{S}_{j'}^{i^*}$ where $\bar{S}_j^{i^*}$ contains the vertices on layers $i^* + j(k + 1)$ and $\bar{S}_{j'}^{i^*}$ contains the vertices on layers $i^* + j(k + 1) + 1$. Moreover, the set A_j can be seen as a union of four sets; $A_j^{i^*}, A_{j'}^{i^*}, A_j^*$ and $A_{j'}^*$ where

- $A_j^{i^*}$ contains vertices from the layer $i^* + j(k + 1)$ that appear in the solution of the piece $G_{j-1}^{i^*}$; and
- $A_{j'}^{i^*}$ contains vertices from the layer $i^* + j(k + 1) + 1$ that appear in the solution of the piece $G_j^{i^*}$; and
- A_j^* contains the vertices from the layer $i^* + j(k + 1)$ that appear in the solution of the piece $G_j^{i^*}$; and
- $A_{j'}^*$ contains the vertices from the layer $i^* + j(k + 1) + 1$ that appear in the solution of the piece $G_{j-1}^{i^*}$.

Now, neighbours of a vertex at some layer i can only be present in layers $i - 1, i$ and $i + 1$. Hence, vertices of $A_j^{i^*}$ are essential for the solution of $G_{j-1}^{i^*}$ and vertices of A_j^* are essential for the solution of $G_j^{i^*}$. Thus, $|A_j^* \cup A_j^{i^*}| \leq 2|\bar{S}_j^{i^*}|$. Following a similar argument, we can show that $|A_{j'}^* \cup A_{j'}^{i^*}| \leq 2|\bar{S}_{j'}^{i^*}|$. Together, these two inequalities imply that $|A_j| \leq 2|S_j^{i^*}|$.

To finish the proof, it suffices to set k to be $4/\varepsilon$. As discussed above, a solution using our algorithm is a feasible solution for the problem. Consider the solution A which is the minimum among different choices of i. It holds that $|A| = \sum_j |A_j^{i^*}| + \sum_j |A_j| \leq |S| + 4|S|/k = (1 + \varepsilon)|S|$. Hence the theorem. □

7 CONCLUSION

In this paper, we initiated the algorithmic study of the EI problem. The main takeaway message is that the problem is computationally hard. Thus some compromises have to be made in order to solve it efficiently. In this work, we decided to focus on solutions whose correctness is theoretically guaranteed. This leaves the more heuristic-oriented approach largely untouched (and highly motivated).

REFERENCES

[1] Alexandre Alaphilippe, Chiara Ceccarelli, Léa Charlet, and Martin Mycielski. 2018. Developing a disinformation detection system and sourcing it live – The case study of the 2018 Italian elections. In *EU Disinfo Lab*. https://www.disinfo.eu/wp-content/uploads/2018/05/20180517_Scientific_paper.pdf

[2] Vincenzo Auletta, Diodato Ferraioli, and Gianluigi Greco. 2020. On the complexity of reasoning about opinion diffusion under majority dynamics. *Artif. Intell.* 284 (2020), 103288. https://doi.org/10.1016/j.artint.2020.103288

[3] Brenda S. Baker. 1994. Approximation Algorithms for NP-Complete Problems on Planar Graphs. *J. ACM* 41, 1 (1994), 153–180. https://doi.org/10.1145/174644.174650

[4] Kshipra Bhawalkar, Jon M. Kleinberg, Kevin Lewi, Tim Roughgarden, and Aneesh Sharma. 2015. Preventing Unraveling in Social Networks: The Anchored k-Core Problem. *SIAM J. Discret. Math.* 29, 3 (2015), 1452–1475. https://doi.org/10.1137/14097032X

[5] Hans L. Bodlaender. 1996. A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth. *SIAM J. Comput.* 25, 6 (1996), 1305–1317. https://doi.org/10.1137/S0097539793251219

[6] Hans L. Bodlaender. 1998. A Partial *k*-Arboretum of Graphs with Bounded Treewidth. *Theor. Comput. Sci.* 209, 1-2 (1998), 1–45. https://doi.org/10.1016/S0304-3975(97)00228-4

[7] Hans L. Bodlaender, Gunther Cornelissen, and Marieke van der Wegen. 2022. Problems Hard for Treewidth but Easy for Stable Gonality. In *Graph-Theoretic Concepts in Computer Science - 48th International Workshop, WG 2022, Tübingen, Germany, June 22-24, 2022, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 13453)*. Springer, 84–97. https://doi.org/10.1007/978-3-031-15914-5_7

[8] Hans L. Bodlaender, Carla Groenland, Jesper Nederlof, and Céline M. F. Swennenhuis. 2024. Parameterized problems complete for nondeterministic FPT time and logarithmic space. *Inf. Comput.* 300 (2024), 105195. https://doi.org/10.1016/j.ic.2024.105195

[9] Robert Bredereck and Edith Elkind. 2017. Manipulating Opinion Diffusion in Social Networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. ijcai.org, 894–900. https://doi.org/10.24963/ijcai.2017/124

[10] Matteo Castiglioni, Diodato Ferraioli, Nicola Gatti, and Giulia Landriani. 2021. Election Manipulation on Social Networks: Seeding, Edge Removal, Edge Addition. *J. Artif. Intell. Res.* 71 (2021), 1049–1090. https://doi.org/10.1613/jair.1.12826

[11] Rajesh Chitnis and Pankaj Kumar. 2024. Personal communication.

[12] Rajesh Hemant Chitnis, Fedor V. Fomin, and Petr A. Golovach. 2013. Preventing Unraveling in Social Networks Gets Harder. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, July 14-18, 2013, Bellevue, Washington, USA*. AAAI Press, 1085–1091. https://doi.org/10.1609/aaai.v27i1.8462

[13] Ferdinando Cicalese, Gennaro Cordasco, Luisa Gargano, Martin Milanic, and Ugo Vaccaro. 2014. Latency-bounded target set selection in social networks. *Theor. Comput. Sci.* 535 (2014), 1–15. https://doi.org/10.1016/j.tcs.2014.02.027

[14] Ferdinando Cicalese, Martin Milanic, and Ugo Vaccaro. 2013. On the approximability and exact algorithms for vector domination and related problems in graphs. *Discret. Appl. Math.* 161, 6 (2013), 750–767. https://doi.org/10.1016/j.dam.2012.10.007

[15] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. 2015. *Parameterized Algorithms*. Springer. https://doi.org/10.1007/978-3-319-21275-3

[16] Andreas Darmann and Janosch Döcker. 2021. On simplified NP-complete variants of Monotone 3-SAT. *Discret. Appl. Math.* 292 (2021), 45–58. https://doi.org/10.1016/j.dam.2020.12.010

[17] Mark de Berg and Amirali Khosravi. 2012. Optimal Binary Space Partitions for Segments in the Plane. *Int. J. Comput. Geom. Appl.* 22, 3 (2012), 187–206. https://doi.org/10.1142/S0218195912500045

[18] Reinhard Diestel. 2012. *Graph Theory, 4th Edition*. Graduate texts in mathematics, Vol. 173. Springer. https://doi.org/10.1007/978-3-662-53622-3

[19] Thang N. Dinh, Dung T. Nguyen, and My T. Thai. 2012. Cheap, easy, and massively effective viral marketing in social networks: truth or fiction?. In *23rd ACM Conference on Hypertext and Social Media, HT '12, Milwaukee, WI, USA, June 25-28, 2012*. ACM, 165–174. https://doi.org/10.1145/2309996.2310024

[20] Thang N. Dinh, Huiyuan Zhang, Dzung T. Nguyen, and My T. Thai. 2014. Cost-Effective Viral Marketing for Time-Critical Campaigns in Large-Scale Social Networks. *IEEE/ACM Trans. Netw.* 22, 6 (2014), 2001–2011. https://doi.org/10.1109/TNET.2013.2290714

[21] Rodney G. Downey and Michael R. Fellows. 1995. Fixed-Parameter Tractability and Completeness I: Basic Results. *SIAM J. Comput.* 24, 4 (1995), 873–921. https://doi.org/10.1137/S0097539792228248

[22] Rodney G. Downey and Michael R. Fellows. 2013. *Fundamentals of Parameterized Complexity*. Springer. https://doi.org/10.1007/978-1-4471-5559-1

[23] Piotr Faliszewski, Rica Gonen, Martin Koutecký, and Nimrod Talmon. 2022. Opinion diffusion and campaigning on society graphs. *J. Log. Comput.* 32, 6 (2022), 1162–1194. https://doi.org/10.1093/logcom/exac014

[24] Emilio Ferrara. 2017. Disinformation and social bot operations in the run up to the 2017 French presidential election. *First Monday* 22, 8 (2017). http://dx.doi.org/10.5210/fm.v22i18.8005

[25] Jörg Flum and Martin Grohe. 2006. *Parameterized Complexity Theory*. Springer. https://doi.org/10.1007/3-540-29953-X

[26] Umberto Grandi, Lawqueen Kanesh, Grzegorz Lisowski, Ramanujan Sridharan, and Paolo Turrini. 2023. Identifying and Eliminating Majority Illusion in Social Networks. In *Thirty-Seventh AAAI Conference on Artificial Intelligence, AAAI 2023, Washington, DC, USA, February 7-14, 2023*. AAAI Press, 5062–5069. https://doi.org/10.1609/aaai.v37i4.25634

[27] Jing (Selena) He, Shouling Ji, Raheem A. Beyah, and Zhipeng Cai. 2014. Minimum-sized influential node set selection for social networks under the independent cascade model. In *The Fifteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc'14, Philadelphia, PA, USA, August 11-14, 2014*. ACM, 93–102. https://doi.org/10.1145/2632951.2632975

[28] Toshimasa Ishii, Hirotaka Ono, and Yushi Uno. 2016. (Total) Vector domination for graphs with bounded branchwidth. *Discret. Appl. Math.* 207 (2016), 80–89. https://doi.org/10.1016/j.dam.2016.03.002

[29] David Kempe, Jon M. Kleinberg, and Éva Tardos. 2005. Influential Nodes in a Diffusion Model for Social Networks. In *Automata, Languages and Programming, 32nd International Colloquium, ICALP 2005, Lisbon, Portugal, July 11-15, 2005, Proceedings (Lecture Notes in Computer Science, Vol. 3580)*. Springer, 1127–1138. https://doi.org/10.1007/11523468_91

[30] Ton Kloks. 1994. *Treewidth, Computations and Approximations*. Lecture Notes in Computer Science, Vol. 842. Springer. https://doi.org/10.1007/BFb0045375

[31] Tuukka Korhonen and Daniel Lokshtanov. 2023. An Improved Parameterized Algorithm for Treewidth. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*. ACM, 528–541. https://doi.org/10.1145/3564246.3585245

[32] Carl A. Latkin and Amy R. Knowlton. 2007. Social Network Assessments and Interventions for Health Behavior Change: A Critical Review. *Behavioral Medicine* 34, 6 (2007), 881–896. https://doi.org/10.1177/1090198106297855

[33] H. W. Lenstra Jr. 1983. Integer Programming with a Fixed Number of Variables. *Mathematics of Operations Research* 8, 4 (1983), 538–548.

[34] Kristina Lerman, Xiaoran Yan, and Xin-Zeng Wu. 2016. The Majority Illusion in Social Networks. *PLoS ONE* 11 (2016), 1–13. Issue 2. https://doi.org/10.1371/journal.pone.0147617

[35] Rolf Niedermeier. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press. https://doi.org/10.1093/ACPROF:OSO/9780198566076.001.0001

[36] Jiehui Pan and Tian-Ming Bu. 2019. A Fast Greedy Algorithm for Finding Minimum Positive Influence Dominating Sets in Social Networks. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops, INFOCOM Workshops 2019, Paris, France, April 29 - May 2, 2019*. IEEE, 360–364. https://doi.org/10.1109/INFCOMW.2019.8845129

[37] Fenella Starkey, Suzanne Audrey, Jo Holliday, Laurence Moore, and Rona Campbell. 2009. Identifying influential young people to undertake effective peer-led health promotion: the example of A Stop Smoking In Schools Trial (ASSIST). *Health Educ. Res.* 24, 6 (2009), 977–988. https://doi.org/10.1093/her/cyp045

[38] Thomas W. Valente. 2012. Network Interventions. *Science* 337, 6090 (2012), 49–53. https://doi.org/10.1126/science.1217330

[39] Thomas W. Valente and Patchareeya Pumpuang. 2007. Identifying Opinion Leaders to Promote Behavior Change. *Health Educ. Behav.* 34, 6 (2007), 881–896. https://doi.org/10.1177/1090198106297855

[40] Maaike Venema-Los, Zoé Christoff, and Davide Grossi. 2023. On the Graph Theory of Majority Illusions. In *Multi-Agent Systems - 20th European Conference, EUMAS 2023, Naples, Italy, September 14-15, 2023, Proceedings (Lecture Notes in Computer Science, Vol. 14282)*. Springer, 17–31. https://doi.org/10.1007/978-3-031-43264-4_2

[41] Junlin Wu, Andrew Estornell, Lecheng Kong, and Yevgeniy Vorobeychik. 2022. Manipulating Elections by Changing Voter Perceptions. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI 2022, Vienna, Austria, 23-29 July 2022*. ijcai.org, 557–563. https://doi.org/10.24963/ijcai.2022/79