FACULTY OF INFORMATION TECHNOLOGY CTU IN PRAGUE

Assignment of bachelor's thesis

Title:	Hiding Leaders in Covert Networks: A Computational Complexity Perspective
Student:	Patrik Drbal
Supervisor:	Ing. Šimon Schierreich
Study program:	Informatics
Branch / specialization:	Computer Science 2021
Department:	Department of Theoretical Computer Science
Validity:	

Instructions

Covert networks are social networks often consisting of criminals or other harmful users. When reducing criminal activities, we can try to detect the most influential users in such networks. Leaders of such networks, as expected, try to hide from being seen, e.g., by introducing new connections. Waniek et al. [1] showed that the problem of hiding the leader is NP-complete for multiple centrality measures. This line of research was followed by other authors [2,3]. In this work, we survey the computational complexity of the problem variants studied in the literature and try to derive new complexity and algorithmic results for various structural restrictions of covert networks using the framework of parameterized complexity [4].

References

[1] Marcin Waniek, Tomasz P. Michalak, Talal Rahwan, and Michael Wooldridge. 2017. On the Construction of Covert Networks. In Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS '17). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 1341–1349.
[2] Palash Dey and Sourav Medya. 2019. Covert Networks: How Hard is It to Hide? In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '19). International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 628–637.
[3] Palash Dey and Sourav Medya. 2020. Manipulating Node Similarity Measures in Networks. In Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS '20). International Foundation for Autonomous Agents

and Multiagent Systems, Richland, SC, 321–329.
[4] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. 2015. Parameterized Algorithms. Springer, Cham. ISBN: 978-3-319-21274-6

Bachelor's thesis

HIDING LEADERS IN COVERT NETWORKS: A COMPUTATIONAL COMPLEXITY PERSPECTIVE

Patrik Drbal

Faculty of Information Technology
Department of Theoretical Computer Science
Supervisor: Ing. Šimon Schierreich
May 25, 2023

Citation of this thesis: Drbal Patrik. *Hiding Leaders in Covert Networks: A Computational Complexity Perspective*. Bachelor's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2023.

Contents

List of Figures

On this place, I would like to express my sincere gratitude to the supervisor of this work, Ing. Šimon Schierreich, for generously providing me with his knowledge, expertise and time. Especially for giving me consultations whenever I needed them, as they always strengthened my motivation to keep working and I always left them a little smarter, for willingly and helpfully pointing out mistakes I made, and for being my guide during the whole process, letting me dive deeper into the fascinating world of theoretical computer science.

I further wish to thank Ing. Dana Vynikarová, Ph.D. for teaching me all the formal aspects of writing and structuring scientific texts, and for the ever-present smile on her face and willingness to help.

I am also grateful to my parents for always believing in me and supporting me.

Last but not least, my deepest thanks belong to my girlfriend Dorotka for all the love and support she gives me. Without her, I would hardly be able to achieve such peace of mind and determination that allowed me to properly work on this thesis.

Declaration

I hereby declare that the presented thesis is my own work and that I have cited all sources of information in accordance with the Guideline for adhering to ethical principles when elaborating an academic final thesis. I acknowledge that my thesis is subject to the rights and obligations stipulated by the Act No. 121/2000 Coll., the Copyright Act, as amended, in particular that the Czech Technical University in Prague has the right to conclude a license agreement on the utilization of this thesis as a school work under the provisions of Article 60 (1) of the Act.

Abstract

In this thesis, we provide an introduction to the topic of covert networks and their analysis, with emphasis on the Hiding Leaders problem which we study with respect to the degree centrality measure and using the framework of parameterized computational complexity.

We analyze results for the Hiding Leaders problem provided in the literature and put them into a perspective of the parameterized complexity. We also point out that there are multiple definitions of the problem and review some of the results from the literature with respect to the definition we picked.

We then bring new results to the domain by showing that the problem is W[1]-hard when parameterized by $b+d$ even when $|L| = 1$, and para-NP-hard with respect to $|L|$, where b denotes the number of edges allowed to be added into a given network, d denotes the number of followers needed to have centrality at least as high as any of the leaders and L denotes a set of leaders. We also show that the problem is in XP when parameterized by b or d.

In the end, we put together our results and their corollaries – for parameters $|L|$, b, d – and results from the literature – for parameter λ, where λ denotes a maximum degree among leaders – and visualize them altogether in an overviewing graph, providing an easily accessible summary of the current state of the research of Hiding Leaders given the decision version of the problem, the degree centrality and the four aforementioned parameters.

Keywords Hiding Leaders problem, covert networks, parameterized complexity, degree centrality

Abstrakt

V této práci poskytujeme úvod do problematiky skrytých sítí (*covert networks*) a jejich analýzy, s důrazem na problém Hiding Leaders, který zkoumáme s ohledem na míru centrality měřenou stupněm uzlu (*degree centrality*) a pomocí frameworku parametrizované výpočetní složitosti.

Analyzujeme výsledky pro problém Hiding Leaders uvedené v literatuře a dáváme je do perspektivy parametrizované složitosti. Zdůrazňujeme také, že existuje více definic tohoto problému a přezkoumáváme některé výsledky z literatury s ohledem na námi vybranou definici.

Dále přinášíme nové výsledky a ukazujeme, že problém je W[1]-těžký pro parametr $b+d$ i když $|L| = 1$, a para-NP-těžký vzhledem k $|L|$, kde b označuje počet hran, které mohou být přidány do dané sítě, d označuje požadovaný počet následovníků, kteří musí mít centralitu alespoň tak vysokou jako kterýkoli z vůdců, a L označuje množinu vůdců. Také ukazujeme, že problém je v XP při parametrizaci parametry b nebo d.

V závěru práce spojujeme naše výsledky a jejich důsledky – pro parametry $|L|$, b, d – s výsledky z literatury – pro parametr λ, kde λ označuje maximální stupeň mezi vůdci – a vizualizujeme je společně v přehledném grafu, který poskytuje snadno dostupné shrnutí současného stavu výzkumu Hiding Leaders problému vzhledem k rozhodovací verzi tohoto problému, centralitě měřené stupněm uzlu a čtyřem výše uvedeným parametrům.

Klíčová slova problém Hiding Leaders, covert networks, parametrizovaná složitost, degree centrality

List of abbreviations

CNA	Covert network analysis
FPT	Fixed-parameter tractable
HL	Hiding Leaders
SNA	Social network analysis

Introduction

Covert networks, or covert organizations, are social structures whose one of the main concerns is to operate in secret, concealed from the view of public, the view of governments and intelligence agencies or the view of any other unwanted subject. Covert network analysis (CNA), as an important part of social network analysis (SNA), is then a set of tools and techniques used to study covert networks and their members, aiming to uncover connections between the members and identifying those with a great influence and thus importance, despite many connections remaining unknown during this identification. Such influential users are called leaders. One of the most used techniques in the SNA toolset, and CNA in particular, for detecting leaders of covert networks are *centrality measures*, the concept introduced by Bavelas [1] describing how important, or central, a member of a given network inside this network is. Being crucial for their working, leaders try to stay undetected by such measures, for which they use various counter-techniques such as network modification by adding or deleting edges, or creating whole new networks from scratch.

In this thesis, we survey covert networks and hiding inside of them from a perspective of the Hiding Leaders problem (HL), first defined by Waniek et al. [2] and further studied by various authors [3, 4, 5]. More concretely, our main concern is the Hiding Leaders problem with respect to the degree centrality measure and using the framework of parameterized computational complexity. Because the problem is NP-hard by itself, we expose it to various parameters and see how these parameters affect its computational complexity. Our another goal is to analyze the problem variants studied in the literature and describe the current state of the research in the language of the parameterized complexity framework. We provide a profound description of the Hiding Leaders problem in Chapter 3, for introduction to the theory of parameterized complexity, please refer to Section 2.3.

Chapter 1

Literature review

In this chapter, we bring a thorough examination of the literature on two topics; social network analysis together with covert networks, and parameterized complexity. The study of the former topic provides useful context for the HIDING LEADERS problem, whereas the latter topic gives us tools to better reason about NP-hard problems, among which, as the reader will see, the HIDING LEADERS problem surely belongs.

1.1 Social network analysis & covert networks

First of all, we mention the publication from Morselli [6] that the reader can refer to get an extensive study of criminal networks. Next, for a recent and exhaustive overview of SNA techniques used for covert network disruption, the reader can use the paper from Ficara et al. [7].

Arguably the most extensive line of research is that in which authors use covert networks to analyze structures consisting of harmful actors. That can be terrorist organizations or other structures interested in involving in illegal activities. These networks are also called *dark networks* [8]. Indeed, research in this direction has a positive impact on understanding such harmful organizations, shedding light on their inner processes and the most potentially hazardous actors and helping in dismantling the entire network, preventing further operations and minimizing the damage done. The understanding of how a covert network works inside can lead to an understanding of how it can be broken. Among many others, we mention the work of these authors [2, 3, 8, 9, 10, 11, 12]. However, as noted by Saavedra-Nieves and Casas-Méndez [12], this view on covert networks as organizations with harmful intends is not the only possible since covert populations can consist for example of drug users, illegal immigrants, persecuted jews, people with infectious diseases, activists, or ravers [13]. The usefulness of covert network analysis then lingers in analyzing even such populations. CNA tools, together with other methods of SNA, can be then used even, for example, for analyzing sports teams [14].

The centrality measurement techniques mentioned in the thesis introduction are widely used because finding the most important vertices within a given network is a natural approach when studying real-world social networks [15]. The most classical centrality measures are degree [16], closeness [17], betweenness [18, 19], and core [20] centrality. On top of the standard centrality measures, there are more recent and advanced techniques suitable for measuring one's rank within a network, for example, the *Game of Thieves* algorithm from Mocanu et al. [21], which can be used even within massive networks. For the sake of completeness, we also mention that there also exist centrality measures on edges, e.g., *WERW-Kpath* algorithm from De Meo [22]. The two aforementioned algorithms are more reviewed by Ficara et al. [23].

After detecting influential members within a given network using centrality measures, it is often desirable to talk about their influence in these networks, for which various *models of influence* can be used. Frequently used models include *Independent Cascade* model and *Linear Threshold* model. More on this topic can be found in the book from Easley and Kleinberg [24].

Apart from the centrality measures, there are other important tools SNA has to offer. One of them are *node similarity measures* which are with advantage used in *link prediction* [25, 26]. Link prediction also plays an important part in the research of covert networks as it allows to predict the existence of otherwise hidden connections between members, potentially leading to identification of important members and relations between them. This is why covert network users might want to manipulate the network to minimize the efficiency of such techniques [27]. Besides CNA, link prediction has other useful applications for example in recommendation systems [28, 29].

Next, we mention that leaders of covert networks often (but not always [30]) face what is called the *efficiency/security trade-off* [31] – the dilemma between staying sufficiently hidden, while keeping enough influence over the network – which is the often-mentioned topic in the literature about covert networks [2, 9, 32]. However, the ways in which this trade-off is addressed vary in the literature. Waniek et al. [2], for example, approach this by modeling the security from the perspective of centrality measures (centrality measure based secrecy), whereas the efficiency from the perspective of models of influence. They then show how to construct a network designed specifically to hide its leaders, while keeping their ability to influence the rest of the network. Lindelauf et al. [9], on the other hand, combine graph and game theory to model this dilemma.

When studying covert networks, we often talk about two parties, the *evaders* and the *seekers*; evaders are members (often leaders) of some covert network; seekers are then those who try to identify the evaders. Evaders are typically aware of the importance of their network structure, which can be seen from the efficiency/security dilemma. However, when talking about seekers and evaders, authors typically consider these three scenarios: (i) Evaders act unstrategically in the sense that they are unaware of the existence of seekers and SNA tools seekers use to reveal them; (ii) Evaders behave as strategic actors, well-awared of the centrality analysis seekers use against them [2, 3, 33, 34]. (iii) Both seekers and evaders are strategic. To this end, we mention the work from Waniek et al. [4] in which the authors propose a strategy which tells the seekers which centrality measures they should use to maximize the chances of detecting a leader of a covert network.

In the terminology of this section, the Hiding Leaders problem describes a covert network consisting of strategic leaders in the role of evaders, which are trying to evade being detected by the use of centrality measuring techniques, for which they modify their network by adding new connections between network members. The leaders do not delete any edges because they face the efficiency/security trade-off and deleting edges may decrease their efficiency in the network. The problem is not bound with any influence model and hence can be studied independently of it, leaving the "efficiency" part of the efficiency/security trade-off untouched.

1.2 Parameterized computational complexity

What follows is a summary of the literature on the topic of parameterized complexity, together with a brief introduction to the topic. For a thorough introduction to the topic, please refer to Section 2.3.

The classical complexity analysis is often insufficient when dealing with NP-hard problems, which are problems that are computationally intensive to solve – they are so-called *intractable*. The main reason the classical complexity analysis is an insufficient tool to deal with NP-hard problems is that it cannot distinguish NP-hard problems that are in some sense "harder" than others, or distinguish between "hard" and "easy" instances of some NP-hard problem. This inability of the classical approach to distinguish between the hardness of various NP-hard problems and their instances led to the development of the parameterized complexity theory. Parameterized complexity is a direct generalization of the classical complexity theory and it arms its users with the ability to analyze the running time of algorithms, and thus computational complexities of problems they try to solve, in finer detail – it can be used to show that a particular problem may become tractable when parameterized by some parameter, or that it may stay intractable for some other parameter.

The foundations of parameterized complexity were laid by Downey and Fellows in the series of papers from years 1992 to 1995 [35, 36, 37, 38], which the authors further presented later in 1999 in their book [39], which was refined and once more published in 2013 [40]. Other relevant and potentially useful literature on this topic are two books by Flum with Grohe [41] and by Niedermeier [42] from 2006, book by Hans et al. [43] from 2012, book by Cygan et al. [44] from 2015 (which to a great extent covers knowledge from the previous literature) and two books by Haan [45] and by Fomin et al. (focused mainly on kernelization) [46] from 2019. However, keep in mind that there is a highly active line of research in the field of parameterized complexity, so, as researchers keep making new discoveries, information presented in the work of mentioned authors may not necessarily be up-to-date.

Chapter 2
Preliminaries

Before proceeding to the main parts of the work, let us begin with a couple of definitions and theory the reader might find useful to fully understand later chapters. This chapter introduces preliminaries on graph theory, including graph problems and centrality measures, and classical computational complexity followed by parameterized complexity.

2.1 Graph theory

This section introduces basic concepts from graph theory. For a thorough revision on this topic, please refer to the monograph of Diestel [47].

▶ **Definition 2.1** (Simple graph). *A* simple graph *is an ordered pair $G = (V, E)$, where V and E are disjoint sets, V is a finite nonempty set of arbitrary items called vertices (or nodes) and E is a finite set of unordered pairs of vertices called edges, $E \subseteq \{\{x, y\} \mid x, y \in V \wedge x \neq y\}$.*

Each graph used in this thesis is a simple graph, for this reason, we omit the word "simple" through the text and call each graph just "graph". Also, the terms "graph" and "network" are interchangeable in this thesis as we only use graphs to describe networks. By network, we mean such a graph, where vertices represent members of the network and edges represent, in some sense, a connection between two given members, e.g., enabling their communication.

Sets of all vertices V and edges E of graph G are typically denoted as $V(G)$ and $E(G)$ respectively. The number of vertices in graphs G is typically denoted as n, $|V(G)| = n$ and the number of edges in G is typically denoted as m, $|E(G)| = m$.

▶ **Definition 2.2** (Vertex degree). *A degree of vertex v in some graph G, denoted as $\deg(v)$, is the number of neighbors of v. In other words, it is the number of edges vertex v is a part of, i.e., $\Big|\{x \in V(G) | \{v, x\} \in E(G) \vee \{x, v\} \in E(G)\}\Big|$.*

A set of neighbors of some vertex v is often denoted as $N(v)$, so then $\deg(v) = |N(v)|$.

▶ **Definition 2.3** (r-regular graph). *A graph G is* r-regular *if the degree of each vertex from $V(G)$ is r, i.e., $\forall v \in V(G)\colon \deg(v) = r$.*

▶ **Definition 2.4** (Regular graph). *A graph G is* regular *if there exists some $r \in \mathbb{N}$ for which G is r-regular.*

▶ **Definition 2.5** (Complement graph). *A* complement graph *$\overline{G}$ of graph G is the graph on vertices $V(G)$, where two vertices are adjacent (there is an edge between them) if and only if they are not adjacent in G, i.e., $\forall e \in V(G) \times V(G)\colon e \in E(\overline{G}) \Leftrightarrow e \notin E(G)$.*

▶ **Definition 2.6** (Induced subgraph). *An* induced subgraph $G[S]$ *of graph* G *is the graph on vertices* $V(G[S]) = S \subset V(G)$*, where two vertices are adjacent if and only if they are adjacent in* G*, i.e.,* $(\forall u, v \in S)(\{u, v\} \in E(G[S]) \Leftrightarrow \{u, v\} \in E(G))$*.*

▶ **Definition 2.7** (Complete graph). *A* complete graph *is the graph* G *if there is an edge between every pair of vertices from* $V(G)$*, i.e.,* $(\forall u, v \in V(G))(\{u, v\} \in E(G))$*.*

▶ **Definition 2.8** (Clique). *A clique* C *is the subgraph of graph* G *where* $G[E(C)]$ *is a complete graph.*

▶ **Definition 2.9** (Vertex cover). *A* vertex cover *of graph* G *is the subset* $S \subseteq V(G)$ *where each edge from* $E(G)$ *is covered by some vertex from* S*. In other words, at least one endpoint of each edge is present in* S*, i.e.,* $(\forall \{u, v\} \in E(G))(u \in S \vee v \in S)$*.*

2.1.1 Graph problems

What follows is a definition of the NP-hard graph problem we will use in Proof 4.2.

▶ **Definition 2.10** (k-Clique problem). *Given graph* G*, the* k*-Clique problem is to determine if there exists a clique* C *in* G *where* $|V(C)| = k$*.*

There are many other graph problems related to cliques. The k-Clique problem is a decision problem but there are also optimization variants, for example, the Maximum Clique problem.

2.1.2 Centrality measures

Centrality measure, or centrality for short, is a measure from graph theory that is widely used in social network analysis. Generally, centrality is a function $c : G \times V \to \mathbb{R}$ which describes the importance of a node in a given network. With centrality, we can measure the ranking (position among other nodes) of a given node within their network. What follows is a definition of a degree centrality introduced by Shaw [16], but there are other centralities used in SNA like closeness [17], betweenness [18, 19] or core [20] centrality.

▶ **Definition 2.11** (Degree centrality). *A degree centrality measures the importance of a vertex by its degree. A degree centrality of the vertex* v *in network* G *is defined as:*

$$c_{deg}(G, v) = \deg(v).$$

2.2 Classical computational complexity

In this section, we briefly describe the two most fundamental classical complexity classes, P and NP. Detailed revision on this topic can be found in the textbook from Arora and Barak [48].

Let us start with defining the DTIME complexity class, which we then use to define the P class.

▶ **Definition 2.12** (DTIME). *A complexity class DTIME(f(n)) is the set of all decision problems that are computable in time* $c \cdot f(n)$ *for some constant* $c > 0$ *and some function* $f : \mathbb{N} \to \mathbb{N}$*.*

▶ **Definition 2.13** (P). *A complexity class* P *is the set of all decision problems that are in class DTIME(*n^c*) for any* $c \geq 1$*, i.e.* $\cup_{c \geq 1}$*DTIME*(n^c)*.*

Informally, complexity class P consists of all decision problems that can be solved in polynomial time. Problems for which exists in practice efficient enough algorithm are called *tractable*.

▶ **Definition 2.14** (NP). *A complexity class NP is the set of all decision problems L for which there exist a polynomial $p\colon \mathbb{N} \to \mathbb{N}$ and a deterministic Turing machine M such that for every instance x of L x is a yes-instance if and only if there exists a polynomial-size solution u of x that can be verified on M in polynomial time. Such solution u is then called the* certificate *or the* witness.

Informally, complexity class NP consists of all decision problems that can be solved in polynomial time when multiple steps can be done in parallel. Note that $\mathsf{P} \subseteq \mathsf{NP}$, because, for any problem from P, it takes only a polynomial amount of time to solve it in the first place.

At the end of this section, let us briefly recall computational complexity terms *hardness* and *completeness*. A problem p is, given some complexity class C, C-hard if for any problem c from C there exists a polynomial reduction from c to p. A problem p is then C-complete if it is C-hard and it belongs to C at the same time.

2.3 Parameterized computational complexity

We use the textbook by Cygan et al. [44] as our main source of information on this topic since it provides arguably the most complex and comprehensive overview of the theory of parameterized complexity and, in many cases, presents the state of the art in the field.

As we already mentioned in Section 1.2, parameterized complexity gives us tools that help us distinguish between different difficulty levels in places where NP-hardness fails to do so. To achieve this, parameter complexity introduces a notion of parameterization of the input instance, where the parameter is just some secondary measurement of the input instance. Such parameter is then used together with the input instance size when describing a computational complexity of NP-hard problems. Given a problem, there are typically many parameters we can parameterize the problem by. This leads us to the importance of choosing the right parameter.

When parameterizing some problem, we typically look for parameters that are small in real-world applications, as one of the main goals when studying parameterized problems is to find algorithms for these problems in which the running time is exponential only in the parameter, leaving the running time polynomial in the, potentially enormous, input size. However, not every choice of parameter lets us design such an efficient algorithm, as it seems that many problems with certain parameters admit no efficient algorithm at all. That means the hardness, or tractability, of the problem depends on the parameter we use. Typically, the more information a parameter carries about the input instance, the higher are chances for us to exploit this information and design a faster algorithm.

The typical parameter we often try is the size, or other property, of the solution we are looking for. The other type of parameter is a measure of some property of the input instance. For example the maximum degree, regularity or treewidth measurings of the input graph. For non-graph instances, it may be the maximum length of a string when the input instance consists of a set of strings, or the number of variables when working with Boolean formulas.

Given a problem parameterized by p, algorithm designers commonly talk about some function of p, $f(p)$. Let us note that it is also possible to use more than one parameter. Having parameters k and l, we then talk about a function of k and l, $f(k,l)$. However, we can (and we do) express the parameterization by k and l by using just one parameter $k+l$, $f(k+l)$.

The two most fundamental complexity classes of this theory, used to reason about the complexity of parameterized problems, are FPT and XP. To distinguish between NP-hard problems that are in XP but not in FPT, we can use another set of complexity classes, the W-hierarchy. There is also a complexity class called para-NP. What follows are the formal definitions of some of these classes, together with definitions of the parameterized problem and parameterized reduction. The hierarchy of the classes can be seen in Figure 2.1.

▶ **Definition 2.15** (Parameterized problem). *A parameterized problem is a language $L \subseteq \Sigma^* \times \mathbb{N}$,*

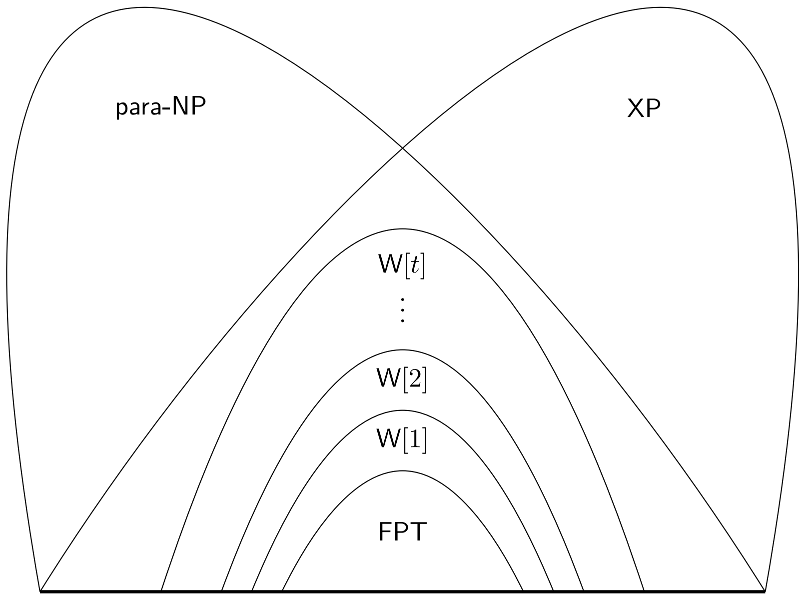


Figure 2.1 Relations among the parameterized complexity classes. Inspired by the depiction from Flum and Grohe [41, p. 97].

where Σ is a fixed, finite alphabet and Σ^* is a set of all strings over Σ. For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the parameter.

One particular problem parameterized by different parameters leads to different parameterized problems.

▶ **Definition 2.16** (Parameterized reduction)**.** *Let $A, B \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems. A parameterized reduction from A to B is an algorithm that, given an instance (x, k) of A, outputs an instance (x', k') of B such that*

- *(x, k) is a yes-instance of A if and only if (x', k') is a yes-instance of B,*
- *$k' \leq g(k)$ for some computable, non-decreasing function g, and*
- *the running time is $f(k) \cdot |x|^{\mathcal{O}(1)}$ for some computable, nondecreasing function f.*

▶ **Definition 2.17** (FPT)**.** *A parameterized problem L is called fixed-parameter tractable (FPT) if there exists*

- *an algorithm $\mathcal{A}$, called fixed-parameter algorithm, or FPT algorithm,*
- *a computable, non-decreasing function $f : \mathbb{N} \to \mathbb{N}$*
- *and a constant c*

such that, given $(x, k) \in \Sigma^* \times \mathbb{N}$, the algorithm $\mathcal{A}$ correctly decides whether $(x, k) \in L$ in time bounded by $f(k) \cdot |(x, k)|^c$.

The complexity class containing all fixed-parameter tractable problems is called FPT. An example of a problem from FPT is n-variable SAT parameterized by n. Indeed, the problem can be solved in time $\Theta(2^n)$ by simply trying each of the 2^n possible evaluations for the variables and checking if the evaluation is correct in time $\Theta(n)$. Moreover, we know from the Exponential Time Hypothesis, formulated by Impagliazzo and Paturi [49], that the problem cannot be solved in time $2^{o(n)}$

The typical goal when designing FPT algorithms is to make factor $f(k)$ and constant exponent c in the running time boundary as small as possible. We can see that if the parameter is equal to the input size, then FPT becomes exponential and, on the other hand if $k = 1$, then FPT becomes P [50].

The only difference between FPT and para-NP problems is that algorithms for para-NP problems are nondeterministic in general. We do not provide the exact definition of para-NP problems as they are not important to us. From the para-NP-hardness perspective, we can say that a problem is para-NP-hard if it is NP-hard already for a constant value of the parameter.

▶ **Definition 2.18** (XP)**.** *A parameterized problem L is called slice-wise polynomial (XP) if there exists*

- *an algorithm $\mathcal{A}$*
- *and two computable, nondecreasing functions $f, g : \mathbb{N} \to \mathbb{N}$*

such that, given $(x, k) \in \Sigma^* \times \mathbb{N}$, the algorithm $\mathcal{A}$ correctly decides whether $(x, k) \in L$ in time bounded by $f(k) \cdot |(x, k)|^{g(k)}$.

The complexity class containing all slice-wise polynomial problems is called XP. An example of a problem from XP is k-Independent Set, in which the question is if there exists a set K of at most k vertices of graph G such that $G[K]$ has no edges. A solution to this problem can be found by trying each of the $\binom{n}{k} \leq n^{\mathcal{O}(k)}$ k-element subsets of vertices and checking in time $\mathcal{O}(n)$ whether or not it is a solution.

Note that definitions of parameterized problems FTP and XP differ only in a running time boundary of algorithm $\mathcal{A}$ and so FPT $\subseteq$ XP. Although both FPT and XP algorithms run in polynomial time for every fixed value of the parameter, the underlying difference between them is that FPT algorithms have the combinatorial explosion restricted to the parameter only, leaving the exponent of the instance size constant, in other words, FPT algorithms are more efficient than XP algorithms.

2.3.1 W-hierarchy

As stated by Cygan et al. [44, p. 423], there are thousands of natural problems which are NP-complete and which are reducible to each other, meaning that in this sense, they are equally hard and thus we can say that they occupy the same level of hardness. However, we cannot say the same when talking about parameterized problems as it seems there are different levels of hardness for such problems and, in this sense, even basic problems seem to be differently hard as they occupy different hardness levels. For this reason, W-hierarchy was introduced by Downey and Fellows [39] as an attempt to shed light on these apparent differences in the hardness of parameterized problems.

The levels of W-hierarchy are marked as $\mathsf{W}[t]$ for $t \in \mathbb{N} \land t \geq 0$, where each level represents its own complexity class. The most important level for us will be W[1], as well as the fact that the k-Clique problem is W[1]-complete [39].

We do not provide the exact definition and further description of W-hierarchy because it is not important for the purposes of this work. On the other hand, what is important is the fact that we interpret the W-hardness as evidence that a problem is not fixed-parameter tractable. This interpretation is based on a general assumption that $\mathsf{FPT} \neq \mathsf{W}[1]$. To show that a certain problem is W[1]-hard, it is typically shown a parameterized reduction from some other problem, which is already known to be W[1]-hard, to it.

At the end of this section, let us note that problems that are W[1]-hard cannot be in FPT but still can be in XP, however, if we show that some problem is para-NP-hard, then it cannot be either in FPT or in XP.

Chapter 3

Problem statement and analysis

Now we define the central problem of this thesis, the Hiding Leaders problem. In this chapter, we also provide an introduction to the notation we use, a detailed description of the problem, a motivation behind its formulation, an explanation of its hardness and a review of the literature on this topic with a description of the differences between problem definitions and with a presentation of the literature results.

▶ **Definition 3.1** (Hiding Leaders)**.** *Let (G, L, b, c, d) be the problem instance, then*

- *$G = (V, E)$ is a network,*
- *$L \subseteq V$ are leaders and the remaining vertices $F = V \setminus L$ are followers,*
- *b is the maximum number of edges that we are allowed to add in G,*
- *$c : G \times V \to \mathbb{R}$ is a centrality measure,*
- *d is a safety margin – the number of followers whose final centrality should be at least as high as of any leader.*

Given this instance, the goal is to identify a set of maximum b edges between followers $W \subseteq F \times F$ such that the resulting network $G' = (V, E \cup W)$ contains at least d followers $F' \subseteq F$ whose centrality must be at least as high as the centrality of any leader, that is,

$$|W| \leq b$$

and

$$\exists_{F' \subseteq F} |F'| \geq d \wedge \forall_{f \in F'} \forall_{l \in L} c(G', f) \geq c(G', l).$$

In other words, we want to *"identify a set of edges to be added between the followers so that the ranking of the leaders (based on some centrality measure) drops below a certain threshold"* [2]. Adding such a set of edges into a given network lets leaders increase their security while not affecting their influence in the network, although, new members with great influence may arise.

We survey HL only with respect to the degree centrality measure, that is, with $c = c_{deg}$. We denote this version of the problem as HL_{deg}. We also use symbol λ to denote a maximum degree among leaders, this is the minimum degree that all followers from F' have to reach. Lastly, we use symbol $\hat{A}$ to denote a set of edges between followers that are not present in G and thus can be added, $\hat{A} = E(\overline{G[F]})$.

Among the first authors who surveyed the topic of evading social network analysis tools, rather than developing new ones, were Waniek et al. [33]. This topic was further researched by the same authors in the subsequent work [2] in which the Hiding Leaders problem was first formulated. Note that the paper from 2017 was only a preliminary version of the work and the completed paper was published later in 2021 [51]. There was also formulated an optimization variant of the Hiding Leaders problem in the completed paper, called Minimum Hiding Leaders, in which there is no budget specified, as the goal is to find the smallest possible set of connections between the followers sufficient to solve the problem. In another work from 2021 [4], Waniek et al. studied a problem similar to the Hiding Leaders problem, which is called Local Hiding. The Hiding Leaders problem was then also studied by Dey and Medya [3].

There are two definitions of the Hiding Leaders problem in the literature and they differ from each other in some details. Namely, the definition[1] from Waniek et al. [2] and the definition from Dey and Medya [3]. The definition from Waniek et al. [2] allows leaders L to be equal to V, $L \subseteq V$, which is a small difference since for $L = V$ the problem is trivial, but the more important difference is that it requires the centrality measure of followers F' to be strictly greater than that of any leader, $\forall_{f \in F'} \forall_{l \in L} c(G', f) > c(G', l)$. We decided to stick to the definition from Dey and Medya [3] because it seems more natural to us to let followers to be part of a solution as soon as their centrality is at least the same as of all the leaders – if no leader has a centrality measure greater than any follower from F', then we consider the leaders hidden and safe from detection. On the other hand, the name "Hiding Leaders" comes from the original definition by Waniek et al. as the name is more indicative of the possibility that there are many leaders in the network. Later in Section 4.1, we show that complexity results for degree centrality from Waniek et al. [2] still hold even with this slightly different definition of HL.

The motivation behind only adding edges, with no deletion involved, is that we want leaders to maintain their existing influence in the network. Indeed, deleting edges incident to leaders would decrease their number of connections and thus potentially reduce their reach within the network. Also, deleting edges between followers would only decrease the values of all degree, betweenness and closeness centrality measures for affected followers [33], which is the exact opposite of what we try to achieve in the Hiding Leaders problem. From this, we can see why it makes sense to only allow adding edges into a given network – because deleting edges incident with leaders violates our requirements on keeping the influence of leaders, whereas deleting edges incident with followers violates our requirements on making leaders more hidden.

Also, note that decreasing the value of the degree centrality of any given member is a straightforward task (unlike decreasing other centralities) as it only consists in cutting edges [33]. On the other hand, HL_{deg} is NP-complete [2]. The core difference between these two problems is that HL_{deg}, and HL in general, is not about decreasing the centrality value of a given member but rather about decreasing their ranking, the relative position among other nodes with respect to the centrality measure. To decrease member's ranking, we must increase the centrality of some other members. This fact, together with budget constraint b and safety margin constraint d, is what stands behind the hardness of the Hiding Leaders problem, because, as Waniek et al. [2] shown, and as we will also show in Chapter 4, there exist instances of HL where finding a solution corresponds to solving certain NP-hard problem.

3.1 State of the art

Here we present literature results for the Hiding Leaders problem. We first take a look at results for the degree centrality measure as it is our main concern, then we complement it with

[1] There is actually also a generalized form of this definition in Waniek's dissertation thesis [52], similar to the definition of Local Hiding, where he also considers a set of edges that can be added and a set of edges that can be removed. However, we consider the first-mentioned set to always contain all possible edges and the second-mentioned set to be empty, which is also the typical situation in the literature. For this reason, we do not distinguish between these two definitions.

results for other centralities. For a presentation of our results, please refer to Section 4.2.

3.1.1 Results for degree centrality

Waniek et al. [2] show that the Hiding Leaders problem is NP-complete for the degree centrality. In more detail, they show that the problem is W[1]-hard when parameterized by $b + d$. We review this proof for our definition of HL in Section 4.1 and present a similar result as Theorem 4.2.

From the work of Dey and Medya [3], we know that HL_{deg} is polynomial-time solvable if the degree of every leader is bounded by some constant. That means that HL_{deg} admits an XP algorithm when parameterized by the degree of leaders, or, in other words, when parameterized by λ. In addition to this, the authors present a 2-approximation algorithm for HL_{deg} which optimizes the number of edges added. They empirically evaluate the algorithm in synthetic networks and conclude that it produces near optimal solutions in practice. They complement that by proving that if there exists a $(2 - \epsilon)$-approximation algorithm for the above problem for any constant $0 < \epsilon < 1$, then there exists a $(\frac{\epsilon}{2})$-approximation algorithm for the Densest k-Subgraph problem.

3.1.2 Results for other centralities

Waniek et al. next show that Hiding Leaders is NP-complete for the closeness [2] and betweenness [51] centralities. In more detail, they show that the problem is W[1]-hard when parameterized by b for both the closeness and the betweenness centralities. In addition to this, the authors show that Hiding Leaders cannot be approximated within a ratio of $(1-\epsilon)\cdot\ln(|F|)$ for any $\epsilon > 0$, unless $\mathsf{P} = \mathsf{NP}$, with respect to both the closeness and the betweenness centralities.

For the core centrality measure, Dey and Medya [3] next show that HL is NP-complete even if the core centrality of every leader is exactly 3. Meaning that HL is para-NP-hard when parameterized by the maximum core centrality of leaders. In addition to this, the authors prove that HL is polynomial-time solvable if the core centrality of every leader is at most 1. They complement that by proving that there does not exist any $((1 - \alpha) \cdot \ln(n))$-approximation algorithm for any constant $\alpha \in (0, 1)$ which optimizes the number of edges that one needs to add even when the core centrality of every leader is 3.

Chapter 4

Contribution

This is the main chapter of the thesis. On this place, we present our contribution to the topic of the Hiding Leaders problem. We first review a proof of a theorem about HL_{deg} from Waniek et al. [2] and show that the proof can be adapted for our definition of Hiding Leaders. Then, we present our own proofs for HL_{deg} regarding various parameters and complexity classes. Lastly, we put our results in the context of the work of other authors, visualizing the parameterized complexity results for HL_{deg} discovered both by us and by authors from the literature.

4.1 Proof revision

Waniek et al. [51] proposed three hardness results for the Hiding Leaders problem given degree, closeness and betweenness centralities. In this section, we review the first aforementioned result with respect to our definition of HL. The original theorem reads:

▶ **Theorem 4.1.** *The Hiding Leaders problem is NP-complete given the degree centrality.*

The proof that follows is almost identical to the original proof from Waniek et al. [51], with only small adjustments done. The adjustments are pretty straightforward but to better understand them, we start the proof by briefly describing the idea of the original proof.

Proof. A common way of proving NP-hardness is to propose a polynomial reduction from another NP-hard problem to the problem we are proving NP-hardness of. In the original proof, the authors show a reduction from the k-Clique problem defined on graph G. The idea is following: In a polynomial time, construct a new graph H from G in a way that in order to solve Hiding Leaders in H we must solve k-Clique in G in the first place. To achieve that, the authors, as a starting point of H, first create a complement of G where all vertices are marked as followers. These will be the only followers in the final H. This initial construction assures that every possible solution of HL will only consist of edges from G, that is, $W \subseteq E(G)$ (and $F' \subseteq V(G)$). Next, they add two types of leaders into H, each type with a different purpose. Leaders of the first type act like "regularizators" as their only job is to connect with followers (in some way) and, by doing so, make the degree of every follower the same and equal to n. These leaders are denoted as X and y in the original proof. The task of the second type, having no connection with followers at all, is then to set a maximum degree across all leaders, which followers from F' will have to reach. These leaders are denoted as L' and are practically the only concern of our adjustments to which we now proceed.

We allow equality of centralities between followers from F' and leaders in our definition of Hiding Leaders. To compensate it, we just need to create one extra leader of the second type, that is, $|L'| = n + k$ as opposed of $|L'| = n + k - 1$ in the original proof, leading to all leaders from L' having degree of $n+k-1$, as opposed of $n+k-2$ in the original proof. Now note that this is the highest degree among the leaders and that all the followers have degree n, so, in order to be part of F', they must obtain at least $(n+k-1)-n = k-1$ new neighbors. We got into exactly the same situation as in the original proof and since, as stated above, leaders from L' do not have any other purpose than setting the maximum degree of all leaders, the rest of the arguments presented in the proof remain unchanged.

We have just shown that allowing equality of centralities between followers from F' and leaders does not break the proof presented by Waniek et al. [2]. The other change in our definition of Hiding Leaders is that we allow all members to be leaders, $L = V$, but since such instances are trivial as they are *yes*-instances if and only if $d = 0$, there is nothing to change in the proof. □

4.2 Our results

The next proof is built on the same idea Waniek et al. [2] presented in their proof of NP-completeness of HL for the degree centrality measure. We reviewed the proof from Waniek et al. [2] and its core ideas in Section 4.1. We present a proof of a stronger theorem, with a simpler reduction graph and in the language of parameterized complexity. These results are possible mainly because of a stronger assumption in the proof where we assume a regular graph as the input.

The core idea of the proof is that we construct a network in such a way that finding a solution W of the Hiding Leaders problem corresponds to finding a solution to the problem we started with, essentially having the starting network present in the constructed network in some manner and then finding a solution inside of it.

▶ **Theorem 4.2.** *Hiding Leaders parameterized by $b+d$ is W[1]-hard even if $|L| = 1$.*

Proof. To prove this theorem, we give a parameterized reduction from the k-Clique problem on regular graphs, which is known to be W[1]-hard with respect to k [53].

Suppose we have a parameterized k-Clique instance (G, k), where $|V(G)| = n$, G is a r-regular graph and $n-2 \geq r \geq k \geq 3$ (for $r > n-2$ or $r < k$ or $k < 3$ the problem is trivial). Next, let us take a parameterized problem $(\mathcal{I}, k')$, where $\mathcal{I} = (H, \{\ell\}, \frac{k\cdot(k-1)}{2}, c_{deg}, k)$ is an instance of HL_{deg} and $k' = b + d = \frac{k\cdot(k-1)}{2} + k$ is the parameter. Graph H is then constructed from graph G in the following way:

1. Start with graph H containing one vertex ℓ, *i.e.*, $H = (\{l\}, \{\})$.
2. Add all vertices from G, *i.e.*, $V(H) \leftarrow V(H) \cup V(G)$, mark these vertices as F_1.
3. Add all edges from $\overline{G}$, *i.e.*, $E(H) \leftarrow E(\overline{G})$.
4. Add an edge between vertex ℓ and $n - r + (k-1) = \lambda$ arbitrarily chosen vertices $X \subset F_1$, *i.e.*, $E(H) \leftarrow E(H) \cup \{(\ell, x) \mid x \in X\}$.
5. For each vertex $v \in V(H) \setminus X =: Y$, introduce a new vertex w_v and add edge $\{v, w_v\}$, *i.e.*, $V(H) \leftarrow V(H) \cup \{w_v \mid v \in Y\}$ and $E(H) \leftarrow E(H) \cup \{(v, w_v) \mid v \in Y\}$, mark the set of w_v for each $v \in Y$ as F_2.

An example of such construction can be seen in Figure 4.1.

Note that step 4 can always be done because $r \geq k$, so $|F_1| = n > n - r + (k-1)$ and since $\deg(\ell) = 0$ (before step 4), there are enough vertices for ℓ to connect with. Considering G is r-regular, its complement, constructed and added into H in steps 2 and 3, must be $(n-r-1)$-regular. After connecting all the F_1 vertices either with ℓ, or the corresponding w_v in steps 4 and 5, they all end up with degree $n-r$; in other words, graph $H[F_1]$ is $(n-r)$-regular. Also, note that the construction of H is done in a time polynomial with n.

Next, notice that the degree of the only leader ℓ is λ, as presented above. The other vertices $V(H) - \ell$ are naturally followers of which we can recognize two types, F_1 and F_2, where $\forall_{f_1 \in F_1} \deg(f_1) = n - r$ and $\forall_{f_2 \in F_2} \deg(f_2) = 1$. Whereas F_1 are vertices of the original graph G, F_2 plays the role of "partners" of vertices Y, since their only job is to substitute a missing connection with ℓ.

Because there is just one leader in $\mathcal{I}$, it is clear that for any follower $f' \in F'$ applies that $\deg(f') \geq \lambda = n - r + (k-1)$. Also, because $n > r$, $\max\limits_{\forall f_1 \in F_1} \deg(f_1) = n - r > 1 = \max\limits_{\forall f_2 \in F_2} \deg(f_2)$.

We now show how finding a k-clique in graph G corresponds to finding a solution W of HL in H, more precisely, we show that (G, k) is a yes-instance of the k-Clique problem if and only if $\mathcal{I}$ is a yes-instance of HL_{deg}.

Let us start with the left-to-right direction, assuming that (G, k) is a *yes*-instance of k-Clique. We denote a set of vertices that form the clique in G as S, $|S| = k$. We can see from the construction above that $S \subseteq F_1$ in H and also, because S form a clique in G, there will be no edge connecting two vertices from S in H. We now make $H[S]$ a complete graph by adding a set of new edges E_S, $|E_S| = \frac{k \cdot (k-1)}{2} = b$, which rise the degree of every follower from S from $n - r$ to $n - r + (k-1) = \lambda$. Because $S \subseteq F_1$, $|S| = k$ and $\forall s \in S\colon \deg(s) = \lambda$, we get that E_S is the solution W of the HL instance $\mathcal{I}$ and S is the corresponding F'.

Next, we show the right-to-left direction of the proof, for which we assume that $\mathcal{I}$ is a yes-instance of HL_{deg}, with newly added edges W and the corresponding set of followers F' as the solution. For that, let us first prove three useful lemmas.

▶ **Lemma 4.3.** *F' can contain no vertex from F_2, i.e., $F' \cap F_2 = \emptyset$.*

Proof. Assume, for sake of contradiction, that $F' \cap F_2 = \{f_2\}$.

Because $|F'| \geq k$, there is at least $k - 1$ vertices from F_1 in F'. Every vertex from F_1 have degree $n - r$ and every vertex from F_2 have degree 1, so, because degree of each vertex from F' is at least λ, each vertex from $F' \cap F_1$ must have gotten at least $\lambda - (n-r) = k-1$ new neighbors and vertex f_2 must have gotten at least $\lambda - 1 = n - r + k - 2$ new neighbors, for which we must have added at least $\frac{(k-1)^2 + (n-r+k-2)}{2} = b + \frac{n-r-1}{2}$ new edges into H. Because $n - 1 \geq r$, we see that $\frac{n-r-1}{2} > 0$, so we must have added more edges than there are in the budget, which is a contradiction. ■

We have shown that we can only add edges between followers F_1 when looking for solution W. We now show that there are exactly k followers in F' and that once we add a new edge between two followers, they must become part of F'.

▶ **Lemma 4.4.** *There are exactly k followers in F', i.e., $|F'| = k$,*

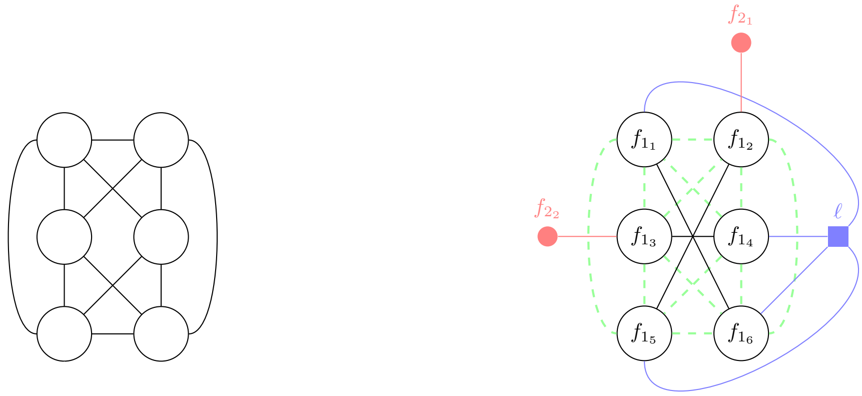
Proof. There cannot be less than k followers in F' because by the definition, there are at least $d = k$ followers in F', so, assume, for sake of contradiction, that $|F'| > k$.

Each vertex from F' must have gotten at least $k - 1$ new neighbors, for which we must have added at least $\frac{|F'| \cdot (k-1)}{2} > \frac{k \cdot (k-1)}{2} = b$, which is a contradiction. ■

▶ **Lemma 4.5.** *W can contain no edge incident to a vertex that is not in F', i.e., $\forall (u, v) \in W\colon u \notin F_1 \setminus F' \wedge v \notin F_1 \setminus F'$.*

Proof. Assume, for sake of contradiction, that $(f', f_1) \in W$, where $f' \in F'$ and $f_1 \in F_1 \setminus F'$. Also assume that (f', f_1) is the only edge from W which is incident to someone from $F_1 \setminus F'$.

We know that each of the $k-1$ vertices from $F' \setminus \{f'\}$ has only gotten new neighbors from F'. Because each vertex from F' must have gotten at least $k - 1$ new neighbors, all of the vertices from $F' \setminus \{f'\}$ must have become neighbors with all the other vertices from F', which means that $\frac{k \cdot (k-1)}{2} = b$ new edges have been added. However, (f', f_1) is also in W, meaning that there are $b + 1$ edges in W, which is a contradiction. ■



(a) A 4-regular ($r = 4$) graph G with $n = 6$ vertices, playing a role of the input graph for the k-Clique problem and is the starting point of our construction.

(b) A graph H constructed from graph G where $H[F_1]$ is 2-regular. The black vertices are followers from F_1, the red vertices are followers from F_2 and the blue vertex is a leader. The black edges are edges $E(\overline{G})$ and the green, dashed lines represent edges from G, which are not present in H, but from which a potential solution W of HL would be picked.

Figure 4.1 A sample construction of graph H as presented in Proof 4.2.

Putting it together, we can see that there are exactly k followers from F_1 in F' and that there are only edges between followers from F' in W.

To get at least $k-1$ new neighbors, each follower from F' has connected with all the other followers from F', meaning that $H[F']$ is a complete graph. Also note that every edge from W must be an edge from the original graph G because $W \subset F_1 \times F_1$ and $F_1 = V(G)$ and also all the edges from $E(\overline{G})$ are already in $E(H)$ so they cannot be in W, since it only contains newly added edges. For this reason, the fact that $H[F']$ is a complete graph means that vertices F' form a k-clique in G.

Because a solution W for HL in H exists if and only if a k-clique in G exists, we can conclude that finding a solution for the Hiding Leaders problem in graph H is at least as hard as finding a solution for the k-Clique problem in graph G.

The reduction presented in this proof is a valid parameterized reduction because; (G, k) is a yes-instance of the k-Clique problem if and only if $\mathcal{I}$ is a yes-instance of HL_{deg}, the construction of H is done in polynomial time; and there is a function of k, $g(k) = \frac{k\cdot(k-1)}{2} + k$, upper-bounding the parameter k' of $\mathcal{I}$. ☐

We have just shown that Hiding Leaders parameterized by $b + d$ is W[1]-hard even with a constant number of leaders. We can easily see that HL remains W[1]-hard even when parameterized by $|L| + b + d$. This is because the problem is W[1]-hard already for $|L|$ bounded by a constant function, hence, bounding it by a nonconstant function will not make the problem easier to solve, leaving it W[1]-hard.

▶ Remark 4.6. Hiding Leaders parameterized by $|L| + b + d$ is W[1]-hard.

Following the argument, we can see that Hiding Leaders is W[1]-hard for any variation of parameters $|L|$, b, d. For example, if we omit b from the parameter, then b is no longer bounded by a function of the parameter and become bounded by a function of the input size, making the problem possibly harder and so leaving it W[1]-hard.

▶ **Corollary 4.7.** *Hiding Leaders parameterized by any variation of $|L|$, b, d is W[1]-hard.*

The parameterized reduction shown in Proof 4.2 is done in polynomial time with respect to input size, that means that it is also a polynomial reduction. That is, there is a polynomial reduction from k-Clique on regular graphs to Hiding Leaders, where $|L| = 1$. Because k-Clique on regular graphs is NP-hard, we get from the reduction that Hiding Leaders is NP-hard as well, even if the number of leaders is constant. From this, we get the following.

▶ **Corollary 4.8.** *Hiding Leaders is para-NP-hard with respect to $|L|$.*

We now look at the problem from the opposite end. Let us recall that we interpret W-hardness as an evidence that the given problem is not in FPT. Hence, there is no use in trying to find FPT algorithms for Hiding Leaders parameterized by $|L|$, b, d, or any of their variation. However, there is still a chance for XP algorithms. We now show that there are indeed such algorithms for parameters b and d, meaning that Hiding Leaders parameterized either by budget b or by safety margin d belongs to XP. From this, we then show that HL is in XP even for the other parameter variations. Let us present a proof of the variant with parameter b first.

▶ **Theorem 4.9.** *Hiding Leaders parameterized by b is in XP.*

Proof. Given parameterized problem $(\mathcal{I}, b)$, where $\mathcal{I} = (G, L, b, c_{deg}, d)$ is a HL_{deg} instance and b is the parameter, we show an algorithm which decides whether or not $(\mathcal{I}, b)$ is a *yes*-instance in time bounded by $f(b) \cdot n^{g(b)}$ for some computable functions f, g.

We start by picking gradually each of the $\binom{|\hat{A}|}{b}$, $\binom{|\hat{A}|}{b} \leq \binom{n^2}{b} \leq \frac{n^{2b}}{b!} \leq n^{\mathcal{O}(b)}$, b-element subsets of edges that can be added between followers, we denote the current subset as B. Because we take all the possible b-element subsets of $\hat{A}$, we eventually find a subset B such that $B = W$; if $(\mathcal{I}, b)$ is a *yes*-instance, then such W exists. We then try to add those edges B into G and after each such addition, we check whether there are at least d followers with degree greater than λ. The addition can be done in time $\mathcal{O}(b)$, and the check can be done in time $\mathcal{O}(n)$.

If instance $(\mathcal{I}, b)$ is not identified as a *yes*-instance for any choice of B, then we conclude that $(\mathcal{I}, b)$ is a *no*-instance because if it was a *yes*-instance, then we would have found it by the procedure above.

The algorithm we have just described runs in time $n^{\mathcal{O}(b)}$. From this, we can conclude that Hiding Leaders parameterized by b is in XP. ☐

Notice that although the just presented algorithm is not possibly the best one we could find, there surely cannot be so much faster algorithm since, as we already know, there can be no FPT algorithm for Hiding Leaders parameterized by b.

Now we proceed to a proof of the variant with parameter d. This proof will be led in a similar manner as the previous proof.

▶ **Theorem 4.10.** *Hiding Leaders parameterized by d is in XP.*

Proof. Given parameterized instance $(\mathcal{I}, d)$, where $\mathcal{I} = (G, L, b, c_{deg}, d)$ is a HL_{deg} instance and d is the parameter, we show an algorithm which decides whether or not $(\mathcal{I}, d)$ is a *yes*-instance in time bounded by $f(d) \cdot n^{g(d)}$ for some computable functions f, g.

We start by picking gradually each of the $\binom{|F|}{d}$, $\binom{|F|}{d} \leq \binom{n}{d} \leq \frac{n^d}{d!} \leq n^{\mathcal{O}(d)}$, d-element subsets of followers F, we denote the current subset as D and a set of edges that can be added between followers from D as $\hat{A}_D \subseteq \hat{A}$, $\hat{A}_D = E(\overline{G[D]}) \leq d^2$. Because we take all the possible d-element subsets of F, we eventually find a subset D such that, after the addition of new edges, $D = F'$; if such F' exists, that is, if $(\mathcal{I}, d)$ is a *yes*-instance. We then try to add $t =: \min(|\hat{A}_D|, b)$ edges between followers from D in each of the $\binom{|\hat{A}_D|}{t} \leq 2^{d^2}$ possible ways. Note that t can be equal to b because we are not allowed to add more than b edges into G. The addition can be done in time $\mathcal{O}(t)$. From now, only one of these three situations can occur:

First, we added t edges into G and all D vertices have a degree greater than λ. Because $|D| = d$, $(\mathcal{I}, d)$ is a *yes*-instance. We can check this in time $\mathcal{O}(n)$.

Second, we added t edges into G but not all D vertices have a degree greater than λ. Moreover, $|\hat{A}_D| < b$ and thus we know that $G[D]$ is (after adding new edges) already a complete graph but we still can add some edges into G before reaching the budget b. In this situation, $D = F'$ if and only if there is at least $\lambda - \min_{\forall \delta \in D} deg(\delta)$ followers outside of D and a number of edges we are still allowed to add before reaching budget b, $b - |\hat{A}_D|$, is smaller than the sum of the number of connections each vertex from D needs to reach λ, $\sum_{\delta \in D} \min(0, \lambda - \deg(\delta))$. This is true because in order for instance $(\mathcal{I}, d)$ to be a *yes*-instance, we must ensure that each follower from D has degree at least λ, for what we connect them to some followers outside D (because D is already complete), for which there must be enough followers outside D and we must have enough edges left before reaching b. Note that if $D = F'$, then $(\mathcal{I}, d)$ in a *yes*-instance. We can check this in time $\mathcal{O}(n)$.

Third, we added t edges into G but not all D vertices have a degree greater than λ. Moreover, $|\hat{A}_D| \geq b$, so $t = b$ and thus the current set D can never be F'. We can check this in time $\mathcal{O}(n)$.

Situations other than one of the three cannot occur. If instance $(\mathcal{I}, d)$ is not identified as a *yes*-instance for any choice of D, then we conclude that $(\mathcal{I}, d)$ is a *no*-instance because if it was a *yes*-instance, then we would have found it by the procedure above.

The algorithm we have just described runs in time $\mathcal{O}(2^{d^2}) \cdot n^{\mathcal{O}(d)}$. From this, we can conclude that Hiding Leaders parameterized by d is in XP. ☐

We have just shown that Hiding Leaders parameterized by b or d is in XP. If we now add some other parameter or parameters to either b or d, then the problem might become easier but surely stays in XP. For example, take $b + |L|$ as the parameter; when not a part of the parameter, |L| was bounded by a function of input size, because it is now a part of the parameter, |L| is bounded by a function of the parameter, hence the problem is possibly easier to solve. From this, we get the following.

▶ **Corollary 4.11.** *Hiding Leaders is in XP when parameterized by at least b or d.*

Last, following the same logic and considering the results from Dey and Medya [3] presented in Subsection 3.1.1, we can state the following.

▶ **Corollary 4.12.** *Hiding Leaders is in XP when parameterized by at least λ.*

4.3 Overview

In this last section, we visualize the results for HL_{deg} presented in Section 4.2, together with the result for parameter λ presented in Subsection 3.1.1. The visualization can be seen in Figure 4.2.

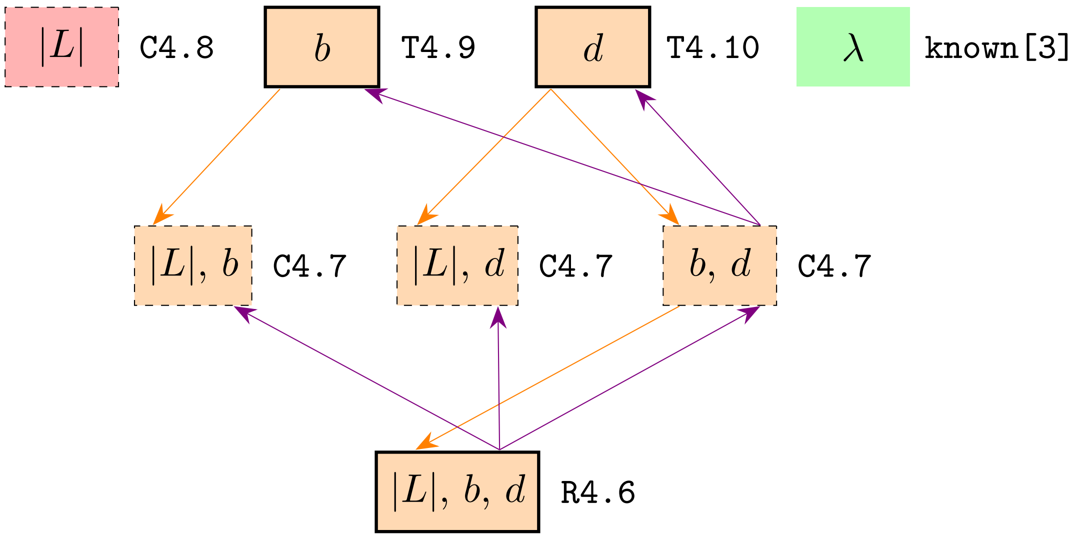


Figure 4.2 The complexity picture for the Hiding Leaders problem given the degree centrality measure, HL_{deg}. Combinations of parameters for which HL_{deg} is in XP but we do not know if they are W[1]-hard are highlighted in green, while combinations for which HL_{deg} is in XP and also W[1]-hard are highlighted in orange, and para-NP-hard combinations are highlighted in red. Results explicitly proven in this work are represented by a black solid border, their corollaries are represented by a black dashed border, and results from the literature do not have any border. Each node has a reference to the corresponding theorem, remark, corollary or bibliography record. Arrows represent how complexity results spread. Orange arrows represent XP class and violet arrows represent W[1]-hardness. When multiple arrows of the same color would aim at the same node, then only one of them is displayed.

Chapter 5

Conclusion and future work

In this thesis, we introduced the reader to the topic of covert networks and their analysis, with emphasis on the Hiding Leaders problem for which we provided a detailed description, we motivated its study and explained its hardness. We mostly followed up on the work from Waniek et al. and Dey with Medya [2, 3]. We pointed out the authors use definitions that are different from each other and described our motivation for picking the one from Dey and Medya. We then reviewed the result for the degree centrality from Waniek et al. with respect to our definition.

The results from the literature are described in a language of classical complexity theory, so we put them into a perspective of the parameterized complexity framework, which we also briefly introduced. We then focused on the problem with respect to the degree centrality measure and presented our own parameterized complexity results in this domain. Namely, we showed that Hiding Leaders is W[1]-hard when parameterized by $b+d$ even if $|L| = 1$. Because the reduction in the proof was done in polynomial time, we got that HL is para-NP-hard with respect to $|L|$. We then showed that HL is in XP when parameterized by b or d. Last, we put together both our results and results from the literature for the degree centrality and parameters $|L|$, b, d, λ and visualized them in an overviewing graph.

Some of the future work on the topic of the Hiding Leaders problem may involve revisioning the results for the closeness and betweenness centralities from Waniek et al. [51] with respect to our definition, or inspecting the problem for various combinations of centralities and parameters for which there are no results yet.

For instance, it would be interesting to see some parameterized complexity results for the core centrality as the problem seems much harder for the core centrality than for the degree centrality [3]. For the variant with the core centrality, one could also survey the computational complexity of HL when the core centrality of every leader is at most 2, as Dey and Medya [3] inspect only the situation where the core centrality of every leader is exactly 3.

For the variant with the degree centrality, which we studied the most in this thesis, it would be interesting to see the parameterization which gives rise to an FPT algorithm as there is known no such algorithm yet. It seems that the parameterization by the *vertex cover number* could, in some cases, bring promising results in this direction. Vertex cover number is the size of a *minimum vertex cover*, minimum vertex cover is the smallest possible number of vertices V of a given graph G such that each edge from $E(G)$ is incident to at least one vertex from V. For the parameter λ, HL_{deg} is in XP and we think that it is also W[1]-hard, so, another immediate future work is to prove or disprove this conjecture.

The Hiding Leaders problem has not yet been studied parameterized by some structural limitation of the input graph. One such structural limitation is the aforementioned *minimum vertex cover*. Another interesting structural parameter to expose the problem to is a *treewidth*. Treewidth is, roughly speaking, a measurement of how well a graph can be decomposed into pieces that are connected in a tree-like fashion [44, p. 151]. The notion of treewidth was described by Robertson and Seymour in the series of papers [54, 55, 56]. Treewidth has vast applications in the design of parameterized graph algorithms and in the design of graph algorithms in general.

Another open problem in this topic is exploring the average case computational complexity of Hiding Leaders for various centrality measures, rather than doing the worst-case analysis. The results from the literature show that the problem is intractable only in the worst case so some heuristics may exist that efficiently solve most of the instances typical in some real-world application. If true, then the apparent complexity of hiding in networks by manipulating their structure could be overcome in many real-world cases. [3]

Bibliography

1. BAVELAS, Alex. A MATHEMATICAL MODEL FOR GROUP STRUCTURES. *Applied Anthropology* [online]. 1948, vol. 7, no. 3, pp. 16–30 [visited on 2023-04-12]. ISSN 00932914. Available from: http://www.jstor.org/stable/44135428.
2. WANIEK, Marcin; MICHALAK, Tomasz P.; RAHWAN, Talal; WOOLDRIDGE, Michael. On the Construction of Covert Networks. In: *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 1341–1349. AAMAS '17.
3. DEY, Palash; MEDYA, Sourav. Covert Networks: How Hard is It to Hide? In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 628–637. AAMAS '19. ISBN 978-1-450-36309-9.
4. WANIEK, Marcin; WO'ZNICA, Jan; ZHOU, Kai; VOROBEYCHIK, Yevgeniy; RAHWAN, Talal; MICHALAK, Tomasz P. Strategic Evasion of Centrality Measures. In: *AAMAS '21*. 2021.
5. KARRA, Mohan. *Parameterized Complexity of the Hiding Leader problem*. India, 2023. MA thesis. Indian Institute of Science Education & Research.
6. MORSELLI, Carlo. *Inside Criminal Networks*. Vol. 8. New York: Springer, 2009. Studies of Organized Crime. ISBN 978-0-387-09526-4.
7. FICARA, Annamaria; CURRERI, Francesco; FIUMARA, Giacomo; DE MEO, Pasquale; LIOTTA, Antonio. Covert Network Construction, Disruption, and Resilience: A Survey. *Mathematics*. 2022, vol. 10, no. 16. ISSN 2227-7390. Available from DOI: 10.3390/math10162929.
8. RAAB, Jörg; MILWARD, H. Brinton. Dark Networks as Problems. *Journal of Public Administration Research and Theory*. 2003, vol. 13, no. 4, pp. 413–439. ISSN 1053-1858. Available from DOI: 10.1093/jpart/mug029.
9. LINDELAUF, Roy; BORM, Peter; HAMERS, Herbert. The influence of secrecy on the communication structure of covert networks. *Social Networks*. 2009, vol. 31, no. 2, pp. 126–137. ISSN 0378-8733. Available from DOI: https://doi.org/10.1016/j.socnet.2008.12.003.
10. XU, Jennifer; CHEN, Hsinchun. Criminal Network Analysis and Visualization. *Communications of the ACM*. 2005, vol. 48, no. 6, pp. 100–107. ISSN 0001-0782. Available from DOI: 10.1145/1064830.1064834.
11. RESSLER, Steve. Social Network Analysis as an Approach to Combat Terrorism: Past, Present, and Future Research. *Homeland Security Affairs*. 2006, no. 2. Available also from: https://www.hsaj.org/articles/171.

12. SAAVEDRA–NIEVES, Alejandro; CASAS–MÉNDEZ, Balbina. On the centrality analysis of covert networks using games with externalities. *European Journal of Operational Research*. 2023, vol. 309, no. 3, pp. 1365–1378. ISSN 0377-2217. Available from DOI: https://doi.org/10.1016/j.ejor.2023.02.023.

13. OLIVER, Kathryn; EDWARDS, Gemma; CROSSLEY, Nick; KOSKINEN, Johan; EVERETT, Martin. *Covert network data: a typology of effects, processes, practices and structures* [online]. 2014. [visited on 2023-04-27]. Tech. rep. Available from: https://hummedia.manchester.ac.uk/schools/soss/research/mitchell/covertnetworks/presentations/Sunbelt2014.pdf.

14. BULDÚ, J. M.; BUSQUETS, J.; ECHEGOYEN, I.; SEIRUL.LO, F. Defining a historic football team: Using Network Science to analyze Guardiola's F.C. Barcelona. *Scientific Reports*. 2019, vol. 9, no. 1, article no. 13602. ISSN 2045-2322. Available from DOI: 10.1038/s41598-019-49969-2.

15. CRESCENZI, Pierluigi; D'ANGELO, Gianlorenzo; SEVERINI, Lorenzo; VELAJ, Yllka. Greedily Improving Our Own Closeness Centrality in a Network. *ACM Transactions on Knowledge Discovery from Data*. 2016, vol. 11, no. 1, pp. 1–32. ISSN 1556-4681. Available from DOI: 10.1145/2953882.

16. SHAW, Marvin E. Group Structure and the Behavior of Individuals in Small Groups. *The Journal of Psychology*. 1954, vol. 38, no. 1, pp. 139–149. Available from DOI: 10.1080/00223980.1954.9712925.

17. BEAUCHAMP, Murray A. An improved index of centrality. *Behavioral Science*. 1965, vol. 10, no. 2, pp. 161–163. Available from DOI: https://doi.org/10.1002/bs.3830100205.

18. ANTHONISSE, Jac M. The rush in a directed graph. *Journal of Computational Physics*. 1971, pp. 1–10.

19. FREEMAN, Linton C. A Set of Measures of Centrality Based on Betweenness. *Sociometry* [online]. 1977, vol. 40, no. 1, pp. 35–41 [visited on 2023-04-12]. ISSN 00380431. Available from: http://www.jstor.org/stable/3033543.

20. SEIDMAN, Stephen B. Network structure and minimum degree. *Social Networks*. 1983, vol. 5, no. 3, pp. 269–287. ISSN 0378-8733. Available from DOI: https://doi.org/10.1016/0378-8733(83)90028-X.

21. MOCANU, Decebal Constantin; EXARCHAKOS, Georgios; LIOTTA, Antonio. Decentralized dynamic understanding of hidden relations in complex networks. *Scientific Reports*. 2018, vol. 34, no. 4. ISSN 2045-2322. Available from DOI: 10.1038/s41598-018-19356-4.

22. DE MEO, Pasquale; FERRARA, Emilio; FIUMARA, Giacomo; PROVETTI, Alessandro. Enhancing community detection using a network weighting strategy. *Information Sciences*. 2013, vol. 222, pp. 648–668. ISSN 0020-0255. Available from DOI: https://doi.org/10.1016/j.ins.2012.08.001. Including Special Section on New Trends in Ambient Intelligence and Bio-inspired Systems.

23. FICARA, Annamaria; SAITTA, Rebecca; FIUMARA, Giacomo; DE MEO, Pasquale; LIOTTA, Antonio. Game of Thieves and WERW-Kpath: Two Novel Measures of Node and Edge Centrality for Mafia Networks. In: *Complex Networks XII*. Ed. by TEIXEIRA, Andreia Sofia; PACHECO, Diogo; OLIVEIRA, Marcos; BARBOSA, Hugo; GONÇALVES, Bruno; MENEZES, Ronaldo. Cham: Springer, 2021, pp. 12–23. ISBN 978-3-030-81854-8.

24. EASLEY, David; KLEINBERG, Jon. *Networks, Crowds, and Markets: Reasoning about a Highly Connected World*. Cambridge University Press, 2010. Available from DOI: 10.1017/CBO9780511761942.

25. ZHOU, Tao; LÜ, Linyuan; ZHANG, Yi-Cheng. Predicting missing links via local information. *The European Physical Journal B*. 2009, vol. 71, no. 4, pp. 623–630. Available from DOI: 10.1140/epjb/e2009-00335-8.

26. WANG, Peng; XU, BaoWen; WU, YuRong; ZHOU, XiaoYu. Link prediction in social networks: the state-of-the-art. *Science China Information Sciences*. 2015, vol. 58, no. 1, pp. 1–38. ISSN 1869-1919. Available from DOI: 10.1007/s11432-014-5237-y.

27. ZHOU, Kai; MICHALAK, Tomasz P.; WANIEK, Marcin; RAHWAN, Talal; VOROBEYCHIK, Yevgeniy. Attacking Similarity-Based Link Prediction in Social Networks. In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 305–313. AAMAS '19. ISBN 9781450363099.

28. HUANG, Zan; LI, Xin; CHEN, Hsinchun. Link Prediction Approach to Collaborative Filtering. In: *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*. Denver, CO, USA: Association for Computing Machinery, 2005, pp. 141–142. JCDL '05. ISBN 1581138768. Available from DOI: 10.1145/1065385.1065415.

29. TALASU, Nitish; JONNALAGADDA, Annapurna; PILLAI, S; JAMPANI, Rahul. A link prediction based approach for recommendation systems. In: 2017, pp. 2059–2062. International Conference on Advances in Computing, Communications and Informatics (ICACCI). Available from DOI: 10.1109/ICACCI.2017.8126148.

30. DEMIROZ, Fatih; KAPUCU, Naim. Anatomy of a dark network: the case of the Turkish Ergenekon terrorist organization. *Trends in Organized Crime*. 2012, vol. 15, pp. 271–295. Available from DOI: https://doi.org/10.1007/s12117-012-9151-7.

31. MORSELLI, Carlo; GIGUÈRE, Cynthia; PETIT, Katia. The efficiency/security trade-off in criminal networks. *Social Networks*. 2007, vol. 29, no. 1, pp. 143–153. ISSN 0378-8733. Available from DOI: https://doi.org/10.1016/j.socnet.2006.05.001.

32. CROSSLEY, Nick; EDWARDS, Gemma; HARRIES, Ellen; STEVENSON, Rachel. Covert social movement networks and the secrecy-efficiency trade off: The case of the UK suffragettes (1906–1914). *Social Networks*. 2012, vol. 34, no. 4, pp. 634–644. ISSN 0378-8733. Available from DOI: https://doi.org/10.1016/j.socnet.2012.07.004.

33. WANIEK, Marcin; MICHALAK, Tomasz P.; WOOLDRIDGE, Michael J.; RAHWAN, Talal. Hiding individuals and communities in a social network. *Nature Human Behaviour*. 2018, vol. 2, no. 2, pp. 139–147. Available from DOI: 10.1038/s41562-017-0290-3.

34. DEY, Palash; MEDYA, Sourav. Manipulating Node Similarity Measures in Networks. In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 2020, pp. 321–329. AAMAS '20. ISBN 9781450375184.

35. DOWNEY, Rod G.; FELLOWS, Michael R. Fixed-Parameter Tractability and Completeness I: Basic Results. *SIAM Journal on Computing*. 1995, vol. 24, no. 4, pp. 873–921. Available from DOI: 10.1137/S0097539792228228.

36. DOWNEY, Rodney G.; FELLOWS, Michael R. Fixed-Parameter Tractability and Completeness II: On Completeness for W[1]. *Theoretical Computer Science*. 1995, vol. 141, pp. 109–131.

37. DOWNEY, Rodney; FELLOWS, Michael. Fixed-parameter tractability and completeness III: Some structural aspects of the IF-hierarchy. 1993.

38. ABRAHAMSON, Karl R.; DOWNEY, Rodney G.; FELLOWS, Michael R. Fixed-Parameter Tractability and Completeness IV: On Completeness for W[P] and PSPACE Analogues. *Annals of Pure and Applied Logic*. 1995, vol. 73, pp. 235–276.

39. DOWNEY, R.G.; FELLOWS, M.R. *Parameterized Complexity*. New York: Springer, 1999. Monographs in Computer Science. ISBN 9780387948836. Available from DOI: `https://doi.org/10.1007/978-1-4612-0515-9`.

40. DOWNEY, Rodney; FELLOWS, Michael. *Fundamentals of Parameterized Complexity*. London: Springer, 2013. Texts in Computer Science. ISBN 978-1-4471-5558-4. Available from DOI: `10.1007/978-1-4471-5559-1`.

41. FLUM, Jörg; GROHE, Martin. *Parameterized Complexity Theory*. Heidelberg: Springer, 2006. ISBN 978-3-540-29953-0. Available from DOI: `https://doi.org/10.1007/3-540-29953-X`.

42. NIEDERMEIER, Rolf. *Invitation to Fixed-Parameter Algorithms*. Vol. 31. OUP Oxford, 2006. Oxford Lecture Series in Mathematics and Its Applications. ISBN 9780198566076. Available from DOI: `10.1093/acprof:oso/9780198566076.001.0001`.

43. BODLAENDER, Hans; DOWNEY, Rod; FOMIN, Fedor; MARX, Dániel. *The Multivariate Algorithmic Revolution and Beyond*. Vol. 7370. Heidelberg: Springer, 2012. Lecture Notes in Computer Science. Available from DOI: `https://doi.org/10.1007/978-3-642-30891-8`.

44. CYGAN, Marek; FOMIN, Fedor V.; KOWALIK, Łukasz; LOKSHTANOV, Daniel; MARX, Dániel; PILIPCZUK, Marcin; PILIPCZUK, Michał; SAURABH, Saket. *Parameterized Algorithms*. Cham: Springer, 2015. ISBN 978-3-319-21274-6.

45. HAAN, Ronald. *Parameterized Complexity in the Polynomial Hierarchy*. Vol. 11880. Heidelberg: Springer, 2019. Lecture Notes in Computer Science. Available from DOI: `https://doi.org/10.1007/978-3-662-60670-4`.

46. FOMIN, Fedor V.; LOKSHTANOV, Daniel; SAURABH, Saket; ZEHAVI, Meirav. *Kernelization: Theory of Parameterized Preprocessing*. Cambridge University Press, 2019. Available from DOI: `10.1017/9781107415157`.

47. DIESTEL, Reinhard. *Graph Theory*. Vol. 173. Fifth edition. Heidelberg: Springer, 2018. Graduate Texts in Mathematics. ISBN 978-3-662-53622-3.

48. ARORA, S.; BARAK, B. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. ISBN 9781139477369.

49. IMPAGLIAZZO, Russell; PATURI, Ramamohan. On the Complexity of k-SAT. *Journal of Computer and System Sciences*. 2001, vol. 62, no. 2, pp. 367–375. ISSN 0022-0000. Available from DOI: `https://doi.org/10.1006/jcss.2000.1727`.

50. KOUTENSKÝ, František. *Kernels for the Max Cut problem*. Czech Republic, 2020. Bachelor's Thesis. Faculty of Information Technology, Czech Technical University in Prague.

51. WANIEK, Marcin; MICHALAK, Tomasz P.; WOOLDRIDGE, Michael; RAHWAN, Talal. How Members of Covert Networks Conceal the Identities of Their Leaders. *ACM Transactions on Intelligent Systems and Technology*. 2021, vol. 13, no. 1. ISSN 2157-6904. Available from DOI: `10.1145/3490462`.

52. WANIEK, Marcin. *Hiding in Social Networks*. Poland, 2017. PhD thesis. University of Warsaw.

53. MATHIESON, Luke; SZEIDER, Stefan. The Parameterized Complexity of Regular Subgraph Problems and Generalizations. In: *Proceedings of the Fourteenth Symposium on Computing*. Wollongong, NSW, Australia: Australian Computer Society, Inc., 2008, pp. 79–86. CATS '08. ISBN 9781920682583.

54. ROBERTSON, Neil; SEYMOUR, P.D. Graph minors. III. Planar tree-width. *Journal of Combinatorial Theory, Series B*. 1984, vol. 36, no. 1, pp. 49–64. ISSN 0095-8956. Available from DOI: `https://doi.org/10.1016/0095-8956(84)90013-3`.

55. ROBERTSON, Neil; SEYMOUR, P.D. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*. 1986, vol. 7, no. 3, pp. 309–322. ISSN 0196-6774. Available from DOI: https://doi.org/10.1016/0196-6774(86)90023-4.

56. ROBERTSON, Neil; SEYMOUR, P. D. Graph minors. V. Excluding a planar graph. *Journal of Combinatorial Theory. Series B*. 1986, vol. 41, no. 1, pp. 92–114. ISSN 0095-8956. Available from DOI: 10.1016/0095-8956(86)90030-4.