FACULTY
OF INFORMATION
TECHNOLOGY
CTU IN PRAGUE

Assignment of master's thesis

Title:	Target Set Selection in Sparse and Geometric Networks
Student:	Bc. Michal Dvořák
Supervisor:	doc. RNDr. Dušan Knop, Ph.D.
Study program:	Informatics
Branch / specialization:	Computer Science
Department:	Department of Theoretical Computer Science
Validity:	until the end of summer semester 2023/2024

Instructions

The Target Set Selection problem can be motivated as follows.
In the beginning, all individuals are either infected or healthy.
Next, in discrete rounds, the disease spreads in the network from infected to healthy individuals such that a healthy individual gets infected if and only if a sufficient number of its direct neighbors are already infected.
We represent the social network as a graph.

The task is to find (parameterized) algorithms or present hardness results for networks that either possess some geometric structure (such as unit-disc graphs) or are sparse (such as those with bounded tree-cut width).

Electronically approved by doc. Ing. Jan Janoušek, Ph.D. on 2 February 2023 in Prague.

Master's thesis

TARGET SET SELECTION IN SPARSE AND GEOMETRIC NETWORKS

Bc. Michal Dvořák

Faculty of Information Technology
Department of Theoretical Computer Science
Supervisor: doc. RNDr. Dušan Knop, Ph.D.
May 4, 2023

Citation of this thesis: Dvořák Michal. *Target Set Selection in Sparse and Geometric Networks.* Master's thesis. Czech Technical University in Prague, Faculty of Information Technology, 2023.

Contents

Chtěl bych poděkovat především svému vedoucímu doc. RNDr. Dušanovi Knopovi Ph.D. za nekonečnou podporu a trpělivost při spolupráci se mnou nejen na této práci. Dále bych rád poděkoval všem kamarádům a blízkým, kteří mi během tvorby této diplomové práce jakkoliv pomáhali, motivovali mě nebo mě psychicky podporovali. Díky, že jste na mě nezanevřeli i přesto, že jsem během psaní této práce na vás neměl tolik času, kolik bych si představoval.

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.

In Prague on May 4, 2023 ....

Abstract

We study the following model of opinion spread in a social network. In the beginning, some individuals in the network adopt a concrete opinion (for example, they are bribed to adopt it). Next, in discrete rounds, the opinion spreads throughout the network in the following way. An individual that has not yet adopted the opinion, adopts it if a sufficient number of its direct neighbors possess the opinion. The task is to initially influence a small number of individuals such that the opinion floods the entire network. This model corresponds to a notorious hard problem called TARGET SET SELECTION. In this work, we address geometric graphs, in particular, unit disk graphs. We show that even in this class TARGET SET SELECTION remains NP-hard, even if maximum degree of the underlying graph is 4 and thresholds are at most 2. We also show NP-hardness of TARGET SET SELECTION in the majority and unanimous threshold settings. En route, we show similar hardness results for related classes of graphs such as disk contact or grid graphs.

Keywords influence spread, TARGET SET SELECTION, geometrical graph classes, unit disk graphs, intersection graphs, computational complexity

Abstrakt

V této práci se zabýváme následujícím modelem šíření názoru v sociální síti. Na začátku někteří jedinci přijmou jistý názor (například tak, že jsou uplaceni). Poté se názor v síti šíří v diskrétním smyslu podle následujících pravidel. Jedinec, který ještě tento názor nemá, jej přijme, pokud dostatečné množství jeho přímých sousedů už názor má. Úkolem pak je ovlivnit malé množství jedinců tak, aby názor zaplavil celou síť. Tento model odpovídá notoricky těžkému problému TARGET SET SELECTION. V této práci řešíme tento problém v geometricky motivovaných grafových třídách, konkrétně ve třídě unit disk grafů. Ukazujeme, že i pro tuto třídu je problém TARGET SET SELECTION NP-těžký i když má vstupní graf maximální stupeň 4 a hodnota prahové funkce je nanejvýš 2. Také ukazujeme NP-těžkost v případě, kdy je prahová funkce nastavena na majoritu. Po cestě ukazujeme podobné výsledky pro související třídy grafů jako jsou disk contact grafy nebo mřížkové grafy.

Klíčová slova síření názoru, TARGET SET SELECTION, geometrické grafové třídy, unit disk grafy, průnikové grafy, výpočetní složitost

Chapter 1

Introduction

Imagine you are working in a marketing department of a company. Suppose your firm created a brand new product and you want the public to adopt it. A natural question arises: Which individuals in the population should one target (for instance, these individuals could be given the product for free or with a discount) such that a sufficiently large portion of the whole society adopts the new product? The answer to this question necessarily depends on the structure of the society (i.e., the underlying social network) we are dealing with.

The previous scenario is an example of *diffusion*, which turns out to be a natural phenomenon in many real-world networks. Except for the spread of influence, ideas, or opinion, one could model the spread of rumors in an online social network [6], the propagation of a virus, the spread of diseases in a human contact network [63], wormhole in a computer network [40], and many more real-world scenarios.

A simple way of modeling these situations is to assign each individual v in the network a *threshold* $t(v)$. Each individual is either *active* (influenced, infected) or *inactive* (not influenced, healthy). Each individual v in the network has some set of individuals (e.g., close friends, family, his boss at work, etc.) who directly influence him. Those are his *neighbors*. If at least $t(v)$ of his neighbors become active, then he will become active. The activation is, for simplicity, assumed to be symmetrical for all individuals. That is, if u can influence v, then v can influence u. We stress out, that an individual, after he becomes active, remains active for the rest of the diffusion process. For example, there is no notion of getting healthy after being infected by a virus.

Returning to the original motivation, if a threshold of an individual v is equal to 2, it is sufficient that only 2 of his neighbors adopt the new product, and this automatically convinces v to adopt it too. On the other hand, if a threshold of an individual is large (say, half of the population), it might be of consideration to convince him to adopt the new product in a different way rather than forcing too many of his direct neighbors to adopt it. Suppose we insist that everyone in the population adopts our new product. We could, for example, give out the product for free to individuals with large threshold values. Well, we could as well give out the product for free to all individuals. However, this is impractical, since we can't afford to give out the product for free to everyone. Instead, we would like to look for the *smallest* number of individuals that we have to influence in a different way (say, by giving them the product for free) such that the entire population adopts our new product. Another approach could be to fix the number of products we want to give out for free and look to *maximize* the number of individuals that will eventually buy our new product. We could as well combine these two approaches and look for a set of (at most) k individuals to whom we will give the product for free such that at least ℓ individuals eventually buy the product. In our work, we deal with the first scenario – i.e., we aim to minimize the initial set and influence the whole population.

We represent the network as an undirected graph G in a natural way. The vertices of G correspond to the individuals and two individuals share an edge if they can influence each other in the modeled process. The threshold values of the individuals correspond to a function $t\colon V(G) \to \mathbb{N}$. The task then translates to activating a (small) set of vertices, such that all vertices in G eventually become active in the activation process (refer to Section 1.2 for a formal definition). We refer to the sets that cause activation of the entire network as *target sets*.

This model corresponds to the TARGET SET SELECTION problem (or TSS for short), introduced by Domingos and Richardson [62] in the context of viral marketing on social networks. In their work [62], they originally came up with a probabilistic model and heuristic solutions. Kempe, Kleinberg, and Tardos [45, 46, 47] later refined the model in terms of thresholds, which is the model we follow in this work. They also showed that the problem is NP-hard. They originally considered a model with probabilistic thresholds, called the *linear threshold model* and focused on a slightly different task. For a fixed k, they wanted to find an initial set S of size (at most) k that would maximize the expected number of active individuals at the end of the process.

There are several other notions equivalent to target sets. The term *dynamic monopolies* (or *dynamos*) is used by several authors [10, 33, 65]. Dynamic monopolies are motivated by the spread of fault in majority-based network systems arising in the study of distributed computing and communication networks [1]. These networks have been used to model the spread of fault to a certain node in a distributed system by checking for faults within a majority of its direct neighbors. This corresponds to TARGET SET SELECTION where the threshold of an individual v is equal to $\left\lceil \frac{\deg v}{2} \right\rceil$.

Another similar notions include *irreversible k-conversion processes* or *irreversible k-threshold processes* [29, 51, 64, 65] which correspond to TARGET SET SELECTION with threshold values set to k for all individuals. An equivalent notion to the latter is that of *k-neighborhood bootstrap percolation*, as studied in [7, 57].

Known results

TARGET SET SELECTION is known to be computationally very hard from both exact computation and approximation points of view. In the decision variant, the task is to decide whether the input graph admits a target set of size at most k (see Section 1.2 for a formal definition). In the optimization variant, the task is to minimize the size of the target set. The decision variant is NP-hard in the general case as it generalizes the well-known VERTEX COVER problem (we include a proof of this fact in Section 1.2).

Restriction of the threshold function

As TARGET SET SELECTION is NP-hard in the general case, it was first attempted to tackhle the problem's complexity by restricting the threshold function. There are three main settings or restrictions of the threshold function commonly studied:

Unanimous thresholds – $t(v) = \deg v$ for all vertices v,

Constant thresholds – $t(v) \leq c$ for all vertices v, and some fixed constant c,

Majority thresholds – $t(v) = \left\lceil \frac{\deg v}{2} \right\rceil$ for all vertices v.

The unanimous threshold setting resembles the well-known VERTEX COVER problem.

Dreyer and Roberts [29] showed that TARGET SET SELECTION remains NP-hard even if the threshold function is bounded by a constant $c \geq 3$. Chen [17] extended this result and showed NP-hardness even in the case where thresholds are at most 2. The problem is trivial when the thresholds are at most 1 since it suffices (and is also necessary) to target exactly one vertex in each connected component (not containing a vertex of threshold 0) of the input graph. Optimization variant of TARGET SET SELECTION cannot be approximated within a polylogarithmic factor, unless $\mathsf{NP} \subseteq \mathsf{DTIME}(n^{\mathrm{polylog}\, n})$, even in constant degree graphs with thresholds equal to 2 [17].

NP-hardness of the majority setting is originally due to Peleg [61].

One could also consider exact thresholds, i.e., not $t(v) \leq c$ for some constant c, but $t(v) = c$ for all v. In Chapter 3, we show that there is a slight difference in the latter two settings for $c = 2$. A generalization of the three settings (unanimous, exact, and majority) is to make the threshold function only depend on the degree of a vertex (also known as degree-dependent thresholds [60]). In other words, $t(v) = f(\deg v)$ for some (computable) function f. The unanimous setting corresponds to the case when f is the identity function, exact corresponds to the case when f is constant, and majority corresponds to the case when $f(x) = \left\lceil \frac{x}{2} \right\rceil$. This generalization of the threshold function was considered in multiple works [19, 30].

Restriction of the graph structure

As natural restrictions of the threshold function turned out to be either trivial or still NP-hard, attempts were made to restrict the underlying graph structure.

The problem is solvable in polynomial time if the underlying graph has diameter one (i.e., it is a complete graph) but becomes NP-hard on graphs of diameter two [60]. Diameter of a graph G is the length of the longest path among all shortest paths between any two vertices in G. In graphs of diameter d every vertex can be reached from any other vertex by a path of length at most d. It is quite common that large social networks tend to have small diameter [60].

The polynomial-time algorithm on complete graphs can be described as follows. Sort the vertices according to their thresholds from largest to smallest, and for each $i \geq 0$ activate the first i vertices and simulate the activation process. It is not hard to see that the smallest i for which the set of first i vertices activates the entire graph is an optimal target set. The sorting part can be accomplished in linear time because we can assume $t(v) \leq \deg(v) < n$ (see Lemma 3.1), and the second part can be optimized to run in linear time using dynamic programming [60].

Another tractability result is due to Chen [17] who showed a simple linear-time algorithm for trees, also based on dynamic programming. This result was further generalized to the class of block-cactus graphs. A *cactus graph* is a graph in which all 2-connected components consist of a cycle or a single edge. Another equivalent characterization is that any two cycles share at most 1 vertex. A *block-cactus graph* is a graph in which each 2-connected component is either a cycle or a complete graph. Chiang et al. [18] proposed a linear-time algorithm for the class of block-cactus graphs, generalizing the linear-time algorithm for trees. In the same work, they also proposed a linear-time algorithm for the class of chordal graphs and thresholds at most 2. A graph is *chordal* if and only if it does not contain a cycle on $k \geq 4$ vertices as an induced subgraph. They also determined the size of an optimal target set for Hamming graphs. A graph is a *Hamming graph* if and only if it is a cartesian product (see Section 1.1 for a formal definition) of nontrivial complete graphs. In this context, a nontrivial complete graph is a complete graph on at least 2 vertices.

Parameterized complexity

Restriction of the underlying graph structure was further investigated in several works using the framework of parameterized complexity. Parameterized complexity aims to study the computational complexity of problems according to their inherent difficulty with respect to one or more *parameters* of the input or output. This allows for a finer analysis of NP-hard problems than in classical complexity. In classical complexity the input of a problem is a string $x \in \Sigma^*$, whereas in parameterized complexity the input is a pair (or more generally an r-tuple for $r-1$ parameters) $(x, \kappa) \in \Sigma^* \times \mathbb{N}$, where κ is referred to as the *parameter*. A parameterized problem is thus a subset of $\Sigma^* \times \mathbb{N}$. It turns out that many NP-hard problems are efficiently solvable if a suitable parameterization is found. A parameterized problem L is said to be *fixed-parameter tractable* or (in the class) FPT if it can be solved in $f(\kappa) \cdot n^{O(1)}$ time for some computable function f, where κ is the parameter and n is the size of the input. A parameterized problem is (in the class) XP if it can be solved in $n^{f(\kappa)}$ time for some computable function f. The classes FPT and XP are analogs to the classes P and EXP from classical complexity. Central to parameterized complexity is the W-hierarchy of classes, defined by the closure of particular parameterized problems under fpt-reductions (see for instance the monograph of Cygan et al. [25] for formal definitions and other details). These classes are:

$$\mathsf{FPT} \subseteq \mathsf{W[1]} \subseteq \mathsf{W[2]} \subseteq \ldots \subseteq \mathsf{W[P]} \subseteq \mathsf{XP}.$$

All inclusions are believed to be strict, however, we only know that FPT $\neq$ XP by a diagonalization argument. The statement FPT $\neq$ W[1] is true, assuming the so-called exponential-time hypothesis. Exponential-time hypothesis (ETH) asserts that the 3-Sat problem cannot be solved in time $O(2^{\varepsilon n})$ for any $\varepsilon > 0$ [42]. Similarly, as we assume P $\neq$ NP, we assume ETH to be true. However, ETH is a stronger claim than P $\neq$ NP. In other words, ETH implies P $\neq$ NP but not vice versa. The class W[1] is a parameterized analog of the class NP from classical complexity. The major goal in the theory of parameterized complexity is to distinguish between parameterized problems that are in FPT and those which are W[1]-hard. If a problem is shown to be W[1]-hard, it is very unlikely that it admits an FPT algorithm.

Parameterizations

A very natural parameterization for many computational problems is parameterization by the size of the solution. For example, while the Vertex Cover problem is NP-hard, deciding whether a graph has a vertex cover of size at most k can be accomplished in FPT time with respect to k. For Target Set Selection, parameterization by the size of the target set is, unfortunately, W[2]-hard even on graphs with diameter 2 [60].

Another natural parameterization is the *distance* to a particular class of graphs. More precisely, the parameter is the minimal number of vertices (or edges) that must be removed from the input graph to obtain a graph from a particular class. More generally, one could also consider addition operations, or contractions, and so on. If a problem is solvable for a particular class $\mathcal{G}$, there is a hope for an efficient algorithm if the graph is "not too far" from being in the class $\mathcal{G}$ (e.g., the number of vertices or edges to be removed or added to obtain a graph from class $\mathcal{G}$ is small). This approach is also known as parameterization by *distance to triviality* [2, 35]. These parameterizations with known results about Target Set Selection include:

- the *feedback vertex set number* and the *feedback edge set number* – the minimal number of vertices (edges) that one has remove from the graph to obtain a graph without cycles (i.e., a forest),
- the *cluster edge deletion number* – the minimal number of edges that must be deleted from the graph to obtain a cluster graph (i.e., a graph which is a disjoint union of cliques),
- the *vertex cover number* – the minimal number of vertices that have to be removed from the graph to obtain a graph without edges,
- the *size of minimal degree-k modulator* – a subset M of vertices in a graph G is called a *degree-k modulator* if removing M from G results in a graph with maximum degree k. Degree 0-modulators are precisely the vertex covers of the graph.

Many of the above parameters coincide with parameters that are optimal solutions to particular optimization problems. For example, the vertex cover number is the size of a minimum vertex cover (see Section 1.1 for a precise definition) of the input graph, i.e., an optimal solution to the optimization variant of the Vertex Cover problem. Another such parameter is the *twin cover number* which is the size of a minimum *twin cover* of the input graph (see [30] for a definition of this parameter).

It turns out that parameterization of Target Set Selection by the feedback vertex set number is W[1]-hard [9]. On the other hand, Target Set Selection is fixed-parameter tractable when parameterized by any of the following: vertex cover number [8, 60], cluster edge deletion number [60], feedback edge number [60], and degree-1 modulator [8]. In addition, Target Set Selection admits a linear kernel (see [25] for a formal definition of kernels) with respect to the parameter feedback edge number [60]. In the majority and constant threshold settings Target Set Selection is fixed-parameter tractable parameterized by the twin cover number but W[1]-hard when the threshold function is unrestricted [30].

Notice that it makes a huge difference if the distance to a particular class is measured via removal of the vertices or the edges. For example, there are graph classes with bounded feedback vertex set number, while having unbounded feedback edge set number. The parameterized complexity of Target Set Selection with respect to these two parameters also differs drastically. While Target Set Selection is W[1]-hard parameterized by the former parameter, it has a linear kernel with respect to the latter parameter.

Among other parameterizations, it is important to mention *structural parameterizations*. Structural parameterizations capture various structural properties of the graph. For example, a very simple structural parameter can be the maximum degree of a graph. For Target Set Selection this parameterization cannot admit even an XP algorithm, unless P = NP, because the problem is NP-hard even on graphs with maximum degree 3. We include a proof of this result in Chapter 3.

A very famous structural parameter is *treewidth* (denoted by tw). Roughly speaking, the treewidth of a graph expresses its "tree-likeness". Refer to [25] for a formal definition. Many NP-hard optimization problems can be solved even in linear time if the input graph has bounded treewidth. These include Dominating Set, Vertex Cover, Independent Set, and Longest Path, to name a few. We refer the reader to [25] for formal definitions of these problems and the corresponding algorithms.

Ben-zwi et al. [9] showed that Target Set Selection can be solved in $n^{O(\mathrm{tw})}$ or $c^{O(\mathrm{tw})}n^{O(1)}$ time, where c is an upper bound on all threshold values. In other words, they showed that Target Set Selection is fixed-parameter tractable in the constant threshold setting and in XP if the threshold function is unrestricted. They also showed that Target Set Selection is W[1]-hard parameterized by treewidth, and there is no algorithm with running time $n^{o(\sqrt{\mathrm{tw}})}$, unless FPT = W[1]. Another way of interpreting the result of Ben-zwi et al. is to say that the Target Set Selection problem is FPT with respect to the combined parameter treewidth and maximum threshold.

An interesting structural parameterization for Target Set Selection is the parameter *neighborhood diversity* (first introduced by Lampis [53]) which, roughly speaking, measures how many distinct neighborhoods the vertices have in the graph (see [30, 53] for definitions). In the context of the Target Set Selection problem, two vertices u and v with the same neighborhood, i.e., $N(u) = N(v)$, should be indistinguishable in the activation process if their threshold t is the same i.e., $t(u) = t(v)$. Dvořák et al. [30] showed that Target Set Selection admits an FPT algorithm with respect to the parameter neighborhood diversity if the threshold function satisfies $t(v) = f(\deg v)$ (they call such setting Uniform Target Set Selection). In fact, they consider the scenario where the threshold function is a function of the neighborhood (i.e., if $t(v) = f(N(v))$) and not necessarily of the degree, which is a slightly weaker assumption. To see this, note that any function f depending only on the degree of a vertex v can be turned into a function f' depending on the set of neighbors of v. The function f' is given by $f'(N(v)) = f(|N(v)|)$. On the other hand, if the threshold function is unrestricted, they show that the problem becomes W[1]-hard parameterized by neighborhood diversity.

The structural parameter *cliquewidth* (denoted by cw) captures the complexity of the input graph in terms of certain algebraic expressions (see [22, 23]). It is similar to treewidth in some sense, but unlike treewidth, cliquewidth may be bounded even for some dense graph classes (see Section 1.1 for a definition of a dense graph class). It is known that $\mathrm{cw} \leq 2^{\mathrm{tw}+1} + 1$ [24]. Hartmann [36] showed that Target Set Selection is fixed-parameter tractable when parameterized by the combined parameter cliquewidth and the maximum threshold of the input graph, generalizing the previous result for the parameters treewidth and maximum threshold of Ben-zwi et al. [9]. As stressed by Hartmann in [36], the dependence of the parameters on the complexity of their algorithm is surprisingly well-behaved.

Other restrictions of graphs

When attempting to restrict an input for a graph problem there are several other natural approaches to consider. One option is to forbid certain structures inside the graph. For example, *trees* are defined as connected graphs that do not contain any cycles. *Chordal graphs* are defined as graphs that do not contain any induced cycle on 4 or more vertices. Another interesting graph class arising from forbidding a structure is the class of *claw-free graphs*. In claw-free graphs, there is no induced star with three leaves, i.e., a $K_{1,3}$. This very simple restriction yields surprising results concerning tractability of otherwise NP-hard problems [59].

Another way of restricting the structure of a graph is to impose bounds on some structural parameters. We could, for example, upper-bound or lower-bound the maximum or minimum degree of a vertex in the graph or impose restriction on the treewidth of the graph.

Further restrictions include forbidding *minors*. A graph H is a minor of a graph G if it can be obtained from some subgraph of G by contracting edges (refer to [28] for a formal definition). Trees can be equivalently characterized by specifying that G is a tree if and only if the cycle graph C_3 is not a minor of G.

Decompositions

Instead of forbidding a certain structure, one could consider a slightly opposite approach and prescribe the building blocks of the graph. An example could be the algebraic expression related to the parameter cliquewidth mentioned in the previous section. Another known graph class defined in such a way is the class of *cographs*. Cographs are defined recursively by the following three rules (refer to Section 1.1 for formal definitions of used terms):

i) The graph K_1 is a cograph.

ii) If G is a cograph, then its complement is also a cograph.

iii) If G and H are cographs, then their disjoint union is a cograph.

When the graph belongs to a certain class defined like this it usually comes with a corresponding *decomposition* or other similar notion which captures the building blocks of the graph. For example, there is a notion of *tree decompositions* related to the parameter treewidth. A *cliquewidth decomposition* would be equivalent to the algebraic formula describing the graph. A cograph is usually represented by its corresponding *cotree*. These decompositions of graphs often yield interesting algorithmic applications (refer to [25] for algorithms on tree decompositions and refer to [21] for a survey about cographs).

Intersection graph classes

Another way of specifying the building blocks of a graph is the concept of *intersection graphs* or *intersection graph classes.* Given a family of sets $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$, one can construct a graph with vertex set $\mathcal{A}$ and connect two sets A_i and A_j $(i \neq j)$ with an edge if and only if $A_i \cap A_j \neq \emptyset$ (see Definition 1.2). It is not hard to observe that any graph can be constructed as an intersection graph (see Observation 1.3). The key idea is to impose restrictions on the family $\mathcal{A}$ from which the graph is built.

This gives rise to a well-known class of *interval graphs*, where the family $\mathcal{A}$ is restricted to be a family of closed intervals with real endpoints. Bessy et al. [10] showed that Target Set Selection can be solved in polynomial time in the class of interval graphs when the threshold function is bounded by a constant.

It turns out to be interesting when the intersection graphs arise from a family of geometrical objects [5, 16, 34, 50]. Recognition of some classes may even be NP-hard. On the other hand, if the representation of the graph is given as an input, the geometric structure of the graph provides interesting algorithmic results for some computationally hard problems. We have seen one such example in the previous paragraph – the interval graphs. Apart from Target Set Selection, many other NP-hard problems can be solved efficiently (even in linear time) in interval graphs. These include Dominating Set [15] and Hamiltonian Cycle [44], to name a few. Interval graphs may also be recognized in linear time [12].

Central to our work are the classes of disk and unit disk graphs. A *disk graph* is an intersection graph arising from a family of closed disks in the Euclidean plane, and *unit disk graph* arises from a family of closed disks in the Euclidean plane with equal diameter. We discuss basic properties of these classes in Section 1.1.

To the best of our knowledge, there are very few results on Target Set Selection regarding intersection graph classes, in particular, classes that possess a geometric structure. We focus on the class of (unit) disk graphs, which further includes a real-world motivation in the context of the Target Set Selection problem.

Unit disk graphs

Unit disk graphs were initially used as a natural model for a topology of ad-hoc wireless communication networks [41]. Interestingly, it is NP-hard to recognize this graph class. More precisely, for a given graph G, the problem of deciding whether G is a unit disk graph is NP-hard [14, 37, 43]. The same hardness result holds for unit disk contact graphs [13], where the disks may only touch at one point, as well as for disk graphs [49]. However, if the disks only touch and are allowed to have arbitrary diameter (i.e., the graph is a disk contact graph), the recognition problem is solvable in linear time [38]. This is because the latter class is equivalent to the class of planar graphs (see Theorem 1.4). Refer to Section 1.1 for formal definitions of unit disk, disk contact, and disk graphs. On the other hand, many computationally hard problems, such as Independent Set or Fractional Coloring, can be efficiently approximated for the class of unit disk graphs [58] even without the disk representation on the input. In contrast, Independent Set cannot be approximated in general graphs within a factor of $n^{1-\varepsilon}$ for any $\varepsilon > 0$, unless $\mathsf{P} = \mathsf{NP}$ [69], and Fractional Coloring cannot be approximated in general within a factor of $n^{1-\delta}$ for some $\delta > 0$, unless $\mathsf{P} = \mathsf{NP}$ [55]. The Clique problem (which essentially shares the same approximation hardness as Independent Set in the general case) can be solved even exactly in polynomial time if the disk representation is given as part of the input [20]. As Target Set Selection is NP-hard in general graphs even with restricted threshold function, it is natural to explore whether restricting the input graph to be a (unit) disk graph provides any tractability result.

Except for purely theoretical motivation, there is also a real-world scenario where Target Set Selection restricted to the class of disk graphs may be identified. Imagine that the activation process simulates a spread of a virus or infection through the network. Initially, the set S is infected by the virus, and threshold values $t(v)$ correspond to *immunity values* of the individuals. The lower the immunity value of an individual v, the higher the chance that the individual v will become infected (i.e., only $t(v)$ of his infected neighbors will make him infected). The task in the Target Set Selection problem then translates to infecting a group of size at most k such that the whole network becomes eventually infected. Inspired by the actual restrictions in the epidemic of coronavirus, especially by social and physical distancing requirements, it is reasonable to model each individual with their personal space which can be, for simplicity, a closed disk. In this sense, two individuals exchange the virus with each other whenever they are “too close” to each other. In other words, their personal spaces intersect. As we model the personal spaces with disks, the underlying graph representing the network will be a disk graph.

In fact, we are specifically addressing *unit* disk graphs. As we will show, even if all disks have the same diameter (i.e., the graph is a unit disk graph), Target Set Selection remains NP-hard. Thus, the problem cannot be easier on disk graphs as disk graphs is a superclass of unit disk graphs.

Our contribution

Our main focus is on the class of unit disk graphs. We extend the hardness results for the constant threshold setting of Target Set Selection and show that Target Set Selection is NP-hard even when restricted to the class of grid graphs and the thresholds are at most 2. This implies NP-hardness for the class of unit disk graphs even with $\Delta G \leq 4$ and thresholds at most 2. En route, we also prove NP-hardness for the class of planar graphs with $\Delta G \leq 4$ and thresholds at most 2.

By a simple modification of our reductions, we also obtain NP-hardness for the majority threshold setting in the classes of grid, planar, and unit disk graphs even when $\Delta G \leq 4$.

We complete the complexity picture by observing how the unanimous threshold setting behaves in these graph classes, namely, grid graphs, planar graphs, and unit disk graphs. We show a tractability result for the class of grid graphs and observe NP-hardness for planar and unit disk graphs.

Lastly, we show that Target Set Selection is NP-hard in the general case when thresholds are at most 2 and $\Delta G \leq 3$. This complements the known tractability result when thresholds are exactly 2 and $\Delta G \leq 3$ [51]. We demonstrate that the tractability result cannot be extended even in the class of unit disk graphs. We show that Target Set Selection remains NP-hard in the classes of unit disk and planar graphs even when $\Delta G \leq 4$ and the thresholds are exactly 2.

We complete the analysis for bounded degree graphs by showing how to compute an optimal target set in graphs with $\Delta G \leq 2$, i.e., cycles and paths.

As an auxiliary result, we show that for $r \in \mathbb{N}$ satisfying $r = -1 \mod 4$ or $r = -1 \mod 5$, the Independent Set problem is NP-hard even when restricted to the class of r-regular unit disk graphs.

Note

This work includes detailed proofs and extensions of results included in a paper co-authored by the author of this thesis. The paper, called *Establishing Herd Immunity is Hard Even in Simple Geometric Networks*, was accepted to the WAW 2023 conference (18th Workshop on Algorithms and Models for the Web Graph) held at the Fields Institute for Research in Mathematical Sciences, Toronto, Canada on May 23-26, 2023.

Organization of the work

The rest of this work is organized as follows. In the following sections, we introduce basic notation and terminology. We provide an extended introduction to graph theory and complexity theory and all necessary definitions and theorems we build upon.

In Chapter 2, we give proofs of our main results. We address the class of unit disk graphs and prove the promised hardness and tractability results.

In Chapter 3, we address bounded degree graphs and the exact threshold setting. We provide the proofs of results regarding gengeral graphs with maximum degrees 3 and 2. We also include proofs regarding the exact threshold setting in the classes of planar and unit disk graphs.

1.1 Preliminaries

We denote by $\mathbb{N}$ the set $\{0, 1, 2, \ldots\}$, $\mathbb{Z}$ is the set of all integers, i.e., $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$, and $\mathbb{Q}$ and $\mathbb{R}$ stand for the sets of rational and real numbers, respectively. For a set A, we denote by $\binom{A}{2}$ the set of all 2-element subsets of A. For a nonnegative integer k, we denote the set $\{1, \ldots, k\}$ by $[k]$. In particular, $[0] = \emptyset$.

1.1.1 Graph Theory

We assume that the reader is familiar with basic concepts from graph theory. We state the most important definitions and results related to our work. For further reading about graph theory and other related topics, we refer the reader to the monograph of Diestel [28].

Finite simple undirected graph is an ordered pair $G = (V, E)$, where V is a finite set of *vertices*, and $E \subseteq \binom{V}{2}$ is a set of *edges*. For a graph G, we denote by $V(G)$ and $E(G)$, respectively, the set of vertices and the set of edges of G. Traditionally, if no confusion can occur, we denote by n the number of vertices and by m the number of edges of a graph.

Throughout this work, we will deal exclusively with graphs that are finite, simple, and undirected. We will thus refer to finite simple undirected graphs simply as *graphs*.

A graph H is a *subgraph* of a graph G if $V(H) \subseteq V(G)$ and $E(H) \subseteq E(G)$. A graph H is an *induced subgraph* of a graph G if $V(H) \subseteq V(G)$ and $E(H) = E(G) \cap \binom{V(H)}{2}$. Let $X \subseteq V(G)$. We denote by $G[X]$ the subgraph of G with vertex set X and edge set $E(G) \cap \binom{X}{2}$. We refer to $G[X]$ as the *subgraph induced by* X. The *complement of a graph* G, denoted by $\overline{G}$, is the graph $\overline{G} \coloneqq \left(V, \binom{V(G)}{2} \setminus E(G)\right)$.

Let $G = (V, E)$ and $G' = (V', E')$ be two graphs. We say that G and G' are *isomorphic*, or that G *is isomorphic to* G', if there exists a bijection $f \colon V \to V'$ such that $\{u, v\} \in E$ if and only if $\{f(u), f(v)\} \in E'$ for all vertices $u, v \in V$. If two graphs G and H are isomorphic, we write $G \simeq H$.

Let $G = (V, E)$ be a graph and $e = \{u, v\} \in E$ its edge. We say that e is *incident* to the vertices u and v, and we refer to u and v as *endpoints* of e. We further call u a *neighbor* of v and vice versa v a neighbor of u. The set of all neighbors of a vertex $v \in V$ is denoted by $N_G(v)$. The number of edges incident to a vertex v is called the *degree* of the vertex v and is denoted by $\deg_G v$. A vertex $v \in V(G)$ with $\deg_G v = 1$ is called a *leaf*.

We refer to the set $N(v)$ as the *open neighborhood* of the vertex v. The *closed neighborhood* of a vertex v is defined to be $N_G[v] \coloneqq N_G(v) \cup \{v\}$. As we are dealing with simple graphs, we have $\deg_G v = |N_G(v)|$. If no confusion can occur, we omit the index G and write just $N(v), N[v]$ or $\deg v$. We extend the neighborhood notation $N_G(v), N_G[v]$ to sets of vertices as follows. For a set of vertices X we define $N_G(X) \coloneqq \bigcup_{v \in X} N(v)$ and similarly $N_G[X] \coloneqq \bigcup_{v \in X} N[v]$.

The *maximum degree* of a graph G is $\Delta G \coloneqq \max\{\deg_G v \mid v \in V(G)\}$. We say that a graph G is *c-regular* if $\deg_G v = c$ for all vertices $v \in V(G)$. We call a graph *regular* if it is c-regular for some constant $c \in \mathbb{N}$.

The *path graph* on n vertices is the graph $P_n \coloneqq ([n], \{\{i, i+1\} \mid i \in [n-1]\})$. The *cycle graph* on n vertices, or simply a *cycle* (of length n), is defined to be the graph $C_n \coloneqq ([n], E(P_n) \cup \{\{n, 1\}\})$. The *complete graph* on n vertices, or a *clique*, is denoted by K_n and is defined by $K_n \coloneqq \left([n], \binom{[n]}{2}\right)$, i.e., it contains all possible edges. The *complete bipartite graph* with partitions of size $m, n \in \mathbb{N}$ (not both of them zero) is the graph $K_{m,n}$ where $V(K_{m,n}) \coloneqq [m] \times \{0\} \cup [n] \times \{1\}$ and $E(K_{m,n}) \coloneqq \{\{(a, 0), (b, 1)\} \mid a \in [m], b \in [n]\}$.

We say that a graph $G = (V, E)$ is *bipartite* if and only if there is a partition of V into two sets A and B such that for all edges $e = \{u, v\} \in E$ either $u \in A$ and $v \in B$ or vice versa. We refer to the sets A and B as *parts* of the graph G. Observe that a graph G is bipartite if and only if it is a subgraph of $K_{m,n}$ for some m, n. There is a folklore characterization of bipartite graphs in terms of odd cycles. See [28, Proposition 1.6.1.] for a proof.

▶ **Theorem 1.1.** *A graph is bipartite if and only if it does not contain a cycle of odd length as a subgraph.*

In our work, we deal with *planar graphs*. Informally speaking, a graph is planar if and only if it can be embedded in $\mathbb{R}^2$ such that vertices occupy distinct points and edges $\{u, v\}$ are curves connecting points corresponding to the vertices u, v. No two curves are allowed to cross except at the endpoints. We refer the reader to [28, Chapter 4] for a more rigorous introduction.

We now introduce the notion of *intersection graphs*. An intersection graph is a natural way to build a graph from a family of sets. The definition is as follows.

▶ **Definition 1.2.** Let $\mathcal{A} = \{A_1, A_2, \dots, A_n\}$ be a family of sets. The *intersection graph* arising from $\mathcal{A}$, denoted by $G(\mathcal{A})$, is defined as follows. $V(G(\mathcal{A})) \coloneqq \mathcal{A}$ and $E(G(\mathcal{A})) \coloneqq \{\{A_i, A_j\} \mid i \neq j \wedge A_i \cap A_j \neq \emptyset\}$. The family $\mathcal{A}$ is called the *intersection model* or the *representation* for the graph G.

▶ **Observation 1.3.** *Every graph is an intersection graph for some family of sets $\mathcal{A}$.*

Proof. Let $G = (V, E)$ be a graph. Let $A_v = \{e \mid v \in e\}$ and $\mathcal{A} = \{A_v \mid v \in V\}$. In other words, for each vertex v take the set of edges incident to v. By definition, two vertices u, v are neighbors if and only if there is an edge $\{u, v\} \in E$, which is equivalent to the fact that $A_u \cap A_v = \{u, v\} \neq \emptyset$. ■

Note that we are explicitly talking about $\mathcal{A}$ as a *family* of sets, not as a *set* of sets. We permit repetitions inside $\mathcal{A}$. More specifically, for some $A_i, A_j \in \mathcal{A}$ we can have $A_i = A_j$ and $i \neq j$.

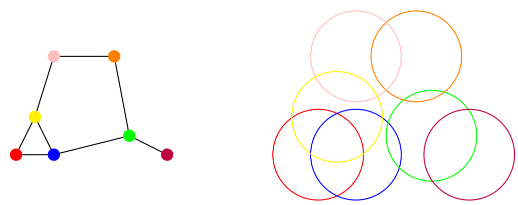
Defining the notion of intersection graphs might initially seem uninteresting, as it is true that any graph can be represented as an intersection graph. However, by imposing restrictions on the intersection model, one can obtain interesting and non-trivial graph classes.

The class of *disk graphs* is the class of intersection graphs for which the intersection model is a set of closed disks in the euclidean plane. The class of *interval graphs* is the class of intersection graphs for which the intersection model is a set of closed interval with real endpoints. The class of disk graphs has several interesting subclasses that we will specifically deal with in this work. These include the classes of disk contact graphs, unit disk graphs and grid graphs.

A *unit disk graph* is a disk graph in which the corresponding intersection model consists of closed disks in the Euclidean plane with equal diameter. An example of a unit disk graph and its corresponding intersection model is given in Figure 1.1. A *disk contact graph* (also known as *coin graph*) is a disk graph $G(\mathcal{D})$ in which the corresponding intersection model further satisfies: For any two distinct disks $D_i, D_j \in \mathcal{D}$ we have $|D_i \cap D_j| \leq 1$. In other words, disks corresponding to adjacent vertices may only touch in exactly 1 point.

There is a known characterization of disk contact graphs which we state here without proof. This result is also known as *Koebe-Andreev-Thurston Theorem* [48].

▶ **Theorem 1.4** (Circle Packing Theorem, folklore)**.** *The class of disk contact graphs is exactly the class of planar graphs.*



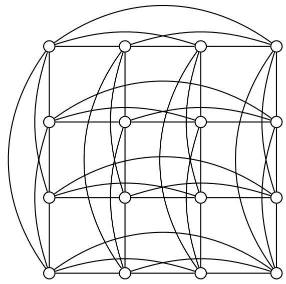
■ **Figure 1.1** An example of a unit disk graph. The graph is depicted on the left, and its corresponding unit disk representation is on the right.

We now define the class of grid graphs. One could define the grid graphs geometrically as follows.

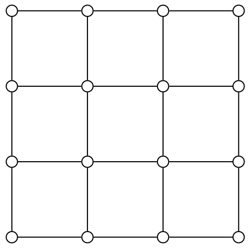
▶ **Definition 1.5.** Let $m, n \in \mathbb{N}, m, n \geq 1$. An $m \times n$ grid is a graph $G = (V, E)$ where $V = [m] \times [n]$ and (x_1, y_1) is a neighbor of (x_2, y_2) if and only if the manhattan distance of the points (x_1, y_1) and (x_2, y_2) is equal to 1. That is, if $|x_1 - x_2| + |y_1 - y_2| = 1$. In other words, if (x_1, y_1) and (x_2, y_2) are two neighboring grid points, they share an edge.

A more graph-theoretical definition is as follows.

▶ **Definition 1.6.** Let G and H be graphs. The *cartesian product* of G and H, denoted by $G \square H$, is the graph $G' = (V', E')$, where $V' = V(G) \times V(H)$, and $\{(u, u'), (v, v')\} \in E'$ if and only if $u = v \wedge u' \in N_H(v')$ or $u' = v' \wedge u \in N_G(v)$.



■ **Figure 1.2** The 4×4 rook graph, or $K_4 \square K_4$



■ **Figure 1.3** The 4×4 grid graph, or $P_4 \square P_4$

▶ **Example 1.7.** $K_2 \square K_2 \simeq C_4$. Cartesian product with K_1 does not mean anything. I.e., $G \square K_1 \simeq G$ for any G.

▶ **Example 1.8.** A *rook graph* is a graph, whose vertices are positions on an $m \times n$ chess board. Two board positions u and v are adjacent if and only if the rook piece can move from u to v. See Figure 1.2. It turns out that the $m \times n$ rook graph is isomorphic to the cartesian product $K_m \square K_n$.

The alternative definition of a grid is as follows.

▶ **Definition 1.9.** An $m \times n$ *grid* is the graph $P_m \square P_n$.

It is not hard to observe that Definitions 1.5 and 1.9 define the same thing. If we don't specifically care about the dimensions of the grid, we refer to an $m \times n$ *grid* simply as *grid.* A *grid graph* is a graph G that is an induced subgraph of a grid.

▶ **Example 1.10.** C_4 is the 2×2 grid as $K_2 \square K_2 = P_2 \square P_2 = C_4$. A 4×4 grid is shown on Figure 1.3

We shall now see the relationship between the graph classes introduced so far. This is the content of Observation 1.11.

▶ **Observation 1.11.** *The classes of disk graphs, unit disk graphs, disk contact graphs, and grid graphs are related in the following way (see also Figure 1.4):*

i) Every unit disk graph is a disk graph, and every disk contact graph is a disk graph.

ii) The class of unit disk graphs and disk contact graphs are unrelated by inclusion.

iii) Every grid graph is also a unit disk graph and also a disk contact graph.

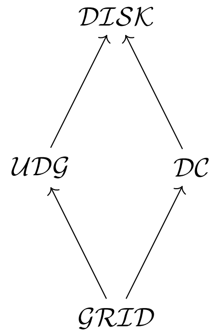
Proof. Claim i) is clear from the definition.

For ii), we make use of the Circle Packing theorem. Observe that clique K_p for $p \geq 5$ is a unit disk graph but not a planar graph (thus not a disk contact graph). On the other hand, for $p \geq 6$, the graph $K_{1,p}$ is not a unit disk graph[1] but is indeed planar (i.e., disk contact).

Finally, for iii) consider the grid graph H and let G be the grid $P_m \square P_n$ for some m, n where H is an induced subgraph of G. Embed G in an obvious way into the integer grid $[m] \times [n] \subseteq \mathbb{Z}^2$. We

[1] Unit disk graphs do not contain $K_{1,p}$ as an induced subgraph for all $p \geq 6$.

first obtain the obvious disk representation for G. We place a disk with diameter 1 on each grid point $(i, j) \in [m] \times [n]$. To obtain disk representation for H, just erase all disks corresponding to vertices in $V(G) \setminus V(H)$. This disk representation also shows that H is also a disk contact graph.[2] ■



■ **Figure 1.4** Relationship between the considered graph classes. An arrow from a class $\mathcal{G}$ to a class $\mathcal{H}$ indicates that $\mathcal{G}$ is a subclass of $\mathcal{H}$. $\mathcal{UDG}$ stands for the class of unit disk graphs, $\mathcal{DC}$ is the class of disk contact graphs, $\mathcal{DISK}$ is the class of disk graphs and $\mathcal{GRID}$ is the class of grid graphs.

When talking about a graph class there is a notion of *sparsity* or *density* which essentially captures how many edges the graph has.

Complete graphs have a large number of edges in the sense that $|E(K_n)| = \Omega(|V(K_n)|^2)$ as $n \to \infty$. Complete graphs are *dense*. The Ω notation is not a precise definition since we also want to call a graph class $\mathcal{G}$ dense if it contains graphs with quadratically many edges. However, not all graphs from G need to have at least $c|V|^2$ edges for some constant c only depending on $\mathcal{G}$. With this definition, we consider the class of bipartite graphs to be dense.

On the other hand, it can be shown (see [28]), that for a planar graph G we have $|E(G)| \leq 3|V(G)| - 6$, asymptotically $|E(G)| = O(|V(G)|)$ as $|V(G)| \to \infty$. The class of planar graphs is an example of a *sparse* class.

Obviously, the notion of sparsity and density only makes sense if the size of the graphs in the class tends to infinity, i.e., we are only interested in infinite graph classes. A formal definition follows.

▶ **Definition 1.12.** Let $\mathcal{G}$ be an infinite class of graphs. $\mathcal{G}$ is said to be *sparse* if there exists a constant c such that for all graphs $G \in \mathcal{G}$, we have $|E(G)| \leq c|V(G)|$. On the other hand, $\mathcal{G}$ is said to be *dense* if there is a constant c such that for every n, there is a graph $G \in \mathcal{G}$ with $|V(G)| = n$ and $|E(G_n)| \geq c|V(G)|^2$.

With this definition, there cannot be a class that is both dense and sparse. However, there can be a class that is neither dense or sparse.

▶ **Observation 1.13.** *The classes of grid and disk contact graphs are sparse, while the classes of unit disk and disk graphs are dense.*

In our work, we make use of a standard graph construction called *graph subdivision*. Let $G = (V, E)$ be a graph and $e \in E$ an edge. The process of *subdividing* the edge e constitutes of replacing e in G by a path graph P and connecting the original endpoints of e with the leaves of P. We will often use the phrase "we subdivide an edge $e \in E(G)$ once, twice, or k times (respectively)" which means we are replacing the concrete edge e by a path P_1, P_2, P_k (respectively) in G.

▶ **Example 1.14.** Take $G = C_3$. By subdividing any edge e once, we obtain the cycle C_4. More generally, taking $G = C_n$, then, by subdividing any edge k times, we obtain the cycle C_{n+k}.

With the notion of edge subdividision we can define the concept of a graph subdivision.

▶ **Definition 1.15.** A *subdivision* of a graph G is a graph G' which is created from G by subdividing some edges $e \in E(G)$.

[2]This is also a consequence of the Circle Packing theorem because grid graphs are planar.

Independent Set and Vertex Cover

We turn our attention to two important problems that we use in our reductions. Before delving into the complexity theory perspective, we define the notions of vertex covers and independent sets and observe a few relations between them and other problems, namely matching.

▶ **Definition 1.16.** Let $G = (V, E)$ be a graph. A set $S \subseteq V$ is a *vertex cover* (of G) if and only if it covers all edges[3]. More precisely, for all edges $e \in E$ we have $S \cap e \neq \emptyset$. A vertex cover S^* is a *minimum* vertex cover if and only if for any other vertex cover S of G we have $|S^*| \leq |S|$.

We first observe an alternative definition of a vertex cover which will be useful later.

▶ **Lemma 1.17.** *Let $G = (V, E)$ be a graph. A set $S \subseteq V$ is a vertex cover of G if and only if for every vertex $v \in V$, we have $v \in S$ or $N(v) \subseteq S$.*

Proof. Suppose that S is a vertex cover and for the sake of contradiction, let v be a vertex with $v \notin S$ and $u \in N(v)$ with $u \notin S$. Notice that this implies $S \cap \{u, v\} = \emptyset$, contradicting the assumption that S was a vertex cover.

On the other hand, suppose that every vertex satisfies $v \in S$ or $N(v) \subseteq S$ and let $\{u, v\} \in E$ be any edge. By assumption, for the vertex v, we have $v \in S$ or $N(v) \subseteq S$. If $v \in S$, we are done. Otherwise, if $v \notin S$, we have some $u \in N(v) \subseteq S$, thus $u \in S$ in this case. Thus, S is a vertex cover. ■

We also have the notion of *independent sets.*

▶ **Definition 1.18.** Let $G = (V, E)$ be a graph. A set $S \subseteq V$ is independent (in G) if and only if for all vertices $u, v \in S$, we have $\{u, v\} \notin E$. In other words, any two vertices inside S are not adjacent.

Vertex covers and independent sets are related by the following lemma.

▶ **Lemma 1.19.** *Let $G = (V, E)$ be a graph. Then $S \subseteq V$ is a vertex cover of G if and only if $V \setminus S$ is an independent set in G.*

Proof. Let S be a vertex cover and consider the set $I = V \setminus S$. Let $u, v \in I$ be two distinct vertices. If $\{u, v\} \in E(G)$, this means that $\{u, v\} \in V \setminus S$, in other words, $S \cap \{u, v\} = \emptyset$, contradicting the assumption that S was a vertex cover. Thus, $\{u, v\} \notin E(G)$ and thus I is an independent set.

On the other hand, let $V \setminus S$ be an independent set. We show that $S = V \setminus (V \setminus S)$ is a vertex cover. Let $\{u, v\} \in E(G)$, and suppose, for the sake of contradiction, that $S \cap \{u, v\} = \emptyset$. This in turn implies that $\{u, v\} \in V \setminus S$. But since $\{u, v\} \in E(G)$, and $V \setminus S$ is, by assumption, an independent set, this is impossible. Thus, S must be a vertex cover. ■

As we already observed in the proof of Lemma 1.19, for $S \subseteq V$ we have $V \setminus (V \setminus S) = S$. Thus, we also get that $S \subseteq V$ is an independent set if and only if $V \setminus S$ is a vertex cover.

Related to the vertex cover is the notion of *matching.* Given graph $G = (V, E)$, a *matching* is a set of edges $M \subseteq E$ satisfying $\forall e, f \in M : e \cap f = \emptyset$. In other words, no two edges share a vertex. A *maximum matching* is a matching M^* such that for any other matching M, we have $|M^*| \geq |M|$.

Interestingly, in the class of bipartite graphs, the size of a maximum matching is equal to the size of a minimum vertex cover. This is a standard result of König [52]. See [28, Theorem 2.1.1.] for a reasonable proof.

▶ **Theorem 1.20** (Kőnig, 1916)**.** *If G is a bipartite graph, then the size of a maximum matching of G is equal to the size of a minimum vertex cover of G.*

Observe that this does not hold for non-bipartite graphs. For example, take the graph C_3. A minimum vertex cover is of size 2, while a maximum matching is of size 1.

[3]That makes sense, right? *Vertex* cover covers *edges.*

1.1.2 Complexity Theory

In our work, we utilize reductions from the Independent Set, the Vertex Cover, and the Sat problems, which we introduce in the following paragraphs.

We assume that the reader is familiar with standard complexity theory concepts such as classes P and NP, NP-hardness, polynomial reductions, etc. For a brief introduction to complexity theory, we refer the reader to the monograph of Arora and Barak [4].

The notions of vertex cover (Definition 1.16) and independent set (Definition 1.18) give rise to the Vertex Cover and Independent Set decision problems. These are defined as follows.

Vertex Cover

Input:	Graph $G = (V, E)$, $k \in \mathbb{N}$
Task:	Is there a vertex cover $C \subseteq V$ of G with $\lvert C\rvert \le k$?

Independent Set

Input:	Graph $G = (V, E)$, $k \in \mathbb{N}$
Task:	Is there an independent set $I \subseteq V$ in G with $\lvert I\rvert \ge k$?

▶ **Theorem 1.21** (folklore, see for instance [4, Theorem 2.16]). *Vertex Cover and Independent Set are NP-complete.*

We will often talk about a problem restricted to a particular class of inputs. This restriction can significantly alter the nature of the problem. Consider the Independent Set and Vertex Cover problems and consider the extreme case where we restrict the input graphs to be cliques. Then the Independent Set and Vertex Cover problems become trivial and rather uninteresting. Largest independent set is of size 1 since any set of size at least 2 contains an edge, and smallest vertex cover is of size $n - 1$ (as a consequence of Lemma 1.19). However, if the problem remains NP-hard even with some restrictions of the input, it could be useful for future hardness reductions. For example, the Independent Set problem remains NP-hard even when restricted to the class of 3-regular planar graphs [32].

One approach to measuring the hardness of a problem in classical complexity theory is to impose restrictions on the problem that narrow the scope of possible inputs, yet still encompass a diverse range of instances that can be considered. If a problem is NP-hard in the general case (i.e., there is tiny hope of a polynomial-time algorithm), the ultimate goal is to find a reasonable restriction of the problem which permits an efficient (polynomial-time) algorithm for the problem, and yet the range of inputs remains broad enough to be meaningful. What exactly is meant by "broad enough" is rather a philosophical question.

For example, consider the restriction of the input graphs to cliques for the Vertex Cover and Independent Set problems. This restriction is rather uninteresting because not many[4] graphs are cliques. Also, the provided algorithm provides minimal insight into the nature of the problem. In contrast, restricting input graphs to trees or, more generally, graphs with bounded treewidth, we obtain a reasonably large class of graphs while still maintaining the polynomial-time[5] solvability. Additionally, the algorithm provides nontrivial insight into the problem's nature.

From this point of view, Independent Set and Vertex Cover are equivalent in the sense that no matter what restriction we impose on the inputs, in any class $\mathcal{G}$, the problems are either

a) both NP-hard, or

b) both polynomial-time solvable.

[4] What exactly is "many" is precisely the question about "broad enough" mentioned in the previous paragraph.
[5] in this case, even linear-time

As both problems are in NP, if neither of a),b) were true, this would be equivalent to $\mathsf{P} \neq \mathsf{NP}$. Assuming $\mathsf{P} \neq \mathsf{NP}$, exactly one of a) and b) applies for each particular class of inputs. We get the following claim as a corollary of Lemma 1.19. We leave the part with polynomial-time solvability of the two problems for the reader.

▶ **Corollary 1.22.** *Let $\mathcal{G}$ be a class of graphs. The following claims are equivalent:*

i) Vertex Cover *is NP-hard restricted to $\mathcal{G}$.*

ii) Independent Set *is NP-hard restricted to $\mathcal{G}$.*

Proof. For i)⇒ii), we reduce from Vertex Cover restricted to $\mathcal{G}$. Given an instance (G, k) of Vertex Cover, we construct an instance (G', k') of Independent Set as follows. Set $G' = G$ and $k' = |V(G)| - k$. By Lemma 1.19, a set $S \subseteq V(G)$ is a vertex cover of size at most k if and only if $V(G) \setminus S$ is an independent set for $G = G'$ of size at least $|V(G)| - k$. The direction ii)⇒i) is proven analogously. ■

It makes sense that if a problem is polynomial-time solvable, then imposing restrictions on the input won't break the polynomial-time solvability. This can be seen the other way around from the hardness side. That is, if a problem is NP-hard under some restrictions, then it is NP-hard even if the restrictions are relaxed. We formalize this in Lemma 1.23.

▶ **Lemma 1.23.** *Let X be a decision problem and let $\mathcal{G} \subseteq \mathcal{H}$ be two classes of inputs for X. Then the following holds:*

i) If X restricted to $\mathcal{G}$ is NP-hard, then X restricted to $\mathcal{H}$ is also NP-hard.

ii) If X restricted to $\mathcal{H}$ is polynomial-time solvable, then X restricted to $\mathcal{G}$ is also polynomial time solvable.

Proof. For claim i) suppose that X restricted to $\mathcal{G}$ is NP-hard. We reduce from X restricted to $\mathcal{G}$. Let x be an instance of X and let $x \in \mathcal{G}$. Since $\mathcal{G} \subseteq \mathcal{H}$, also $x \in \mathcal{H}$. Thus, we provided a reduction from NP-hard instance to an instance $x \in \mathcal{H}$. Thus, X restricted to $\mathcal{H}$ is also NP-hard.

For claim ii) consider a polynomial-time algorithm $A_{\mathcal{H}}$ solving inputs to X restricted to $\mathcal{H}$. We show how to solve instances of X restricted to $\mathcal{G}$ in polynomial time. Let $x \in \mathcal{G}$. By assumption, also $x \in \mathcal{H}$. Thus, we may apply the polynomial-time algorithm $A_{\mathcal{H}}$ to solve the instance $x \in \mathcal{G}$. ■

Consider the graph classes from Figure 1.4. As a consequence of Lemma 1.23, if we show NP-hardness of some problem restricted to the class of grid graphs, we immediately get NP-hardness result for all the superclasses, in this particular case for unit disk, disk contact, and disk graphs.

Satisfiability (SAT)

In the following paragraphs, we introduce the well-known satisfiability problem. A *Propositional formula over variables* $X = \{x_1, \ldots, x_n\}$ is a boolean expression containing the variables and three logical connectives: disjunction ($\vee$), conjunction ($\wedge$), and negation ($\neg$). A *literal* is either a variable or a negation of a variable. If a literal is a variable, it is referred to as *positive*, whereas if it is a negation of a variable, it is referred to as *negative*. A *clause* is a disjunction of literals. A formula φ is in *conjunctive normal form* (*CNF*) if and only if it is a conjunction of clauses. We often shorten this and refer to formulae in CNF simply as *CNF formulae.*

▶ **Example 1.24.** Formula $\varphi = (x \vee \neg z) \wedge (\neg x \vee y \vee z)$ is a CNF formula over variables $\{x, y, z\}$ with two clauses $x \vee \neg z$ and $\neg x \vee y \vee z$. The first clause consists of two literals x and $\neg z$. The literal x is positive, while the literal $\neg z$ is negative.

A *(truth) assignment* for a formula φ over variables $X = \{x_1, \ldots, x_n\}$ is a function $f \colon X \to \{0, 1\}$. We can regard f as a function to the boolean algebra $(\{0, 1\}, \wedge, \vee, \neg)$. This gives rise to a unique extension of f to a homomorphism of boolean algebras $\widetilde{f} \colon FX \to (\{0, 1\}, \wedge, \vee, \neg)$, where FX is the free boolean algebra generated by the set X. From now on, by abuse of notation, we write f instead of $\widetilde{f}$.

In simpler terms, FX corresponds to the set of all propositional formulae over the variables X and $\widetilde{f}$ evaluates the formula in the standard[6] way given the values of the variables.

Assignment f for a formula φ is *satisfying* if and only if $f(\varphi) = 1$. Formula φ is satisfiable if and only if there exists a satisfying assignment for φ.

▶ **Example 1.25.** Going back to Example 1.24, consider again the formula $\varphi = (x \vee \neg z) \wedge (\neg x \vee y \vee z)$. The assignment f_1, given by $f_1(x) = 0, f_1(y) = 1, f_1(z) = 1$, is not a satisfying assignment for φ, because $f_1(\varphi) = (0 \vee 0) \wedge (1 \vee 1 \vee 1) = 0 \wedge 1 = 0$. However, φ is satisfiable, as witnessed by the assignment f_2, given by $f_2(x) = 1, f_1(y) = 1, f_1(z) = 0$. We let the reader verify that $f_2(\varphi) = 1$.

Any CNF formula φ may be viewed as a set of clauses $C_1, \ldots, C_m$, and every clause can be thought of as a set of literals. Note that repeated literals inside a clause, repeated clauses, or the order of the clauses or literals in the formula make no difference on the value $f(\varphi)$ for any assignment f. To see this, note that the connectives $\wedge$ and $\vee$ are associative and commutative, so the order does not matter. The connectives also satisfy the *idempotence law*. That is, $\psi \vee \psi = \psi$ and $\psi \wedge \psi = \psi$ for any formula ψ. It follows that repeated clauses or literals make no difference.

▶ **Observation 1.26.** *Let φ be a CNF formula with clauses $C_1, \ldots, C_m$. An assignment f is satisfying for φ if and only if for each clause C_j we have $f(C_j) = 1$ and that is if and only if there exists a literal $\ell \in C_j$ such that $f(\ell) = 1$.*

In the SAT problem, the input is a propositional CNF formula φ, and the task is to decide whether φ is satisfiable.

SAT

Input:	Propositional CNF formula φ.
Task:	Is φ satisfiable?

▶ **Fact 1.27** (Cook-Levin Theorem)**.** *SAT is NP-complete.*

The SAT problem is an inherent NP-hard problem that is widely studied. Many constrained variants of the SAT problem were shown to be NP-hard and utilized to show NP-hardness for other problems of interest (see for instance [26, 27, 31, 54]).

We now proceed to define one of the variants used in our reductions. The variant is called RESTRICTED PLANAR 3-SAT, and the definition and hardness of this setting comes from the work of Dahlhaus et al. [26], where they used it to prove NP-hardness of the MULTITERMINAL CUT problem.

▶ **Definition 1.28.** Let φ be a propositional CNF formula with clauses $C_1, \ldots, C_m$ and variables $x_1, \ldots, x_n$. The *incidence graph* for φ, denoted by G_φ, has the vertex v_{x_i} for each variable x_i, and the vertex v_{C_j} for each clause C_j. There is an edge between variable vertex v_{x_i} and clause vetex v_{C_j} if and only if the variable x_i occurs in C_j.

It is not hard to observe that the incidence graph of any formula φ is always bipartite with parts $\{v_{x_1}, \ldots, v_{x_n}\}, \{v_{C_1}, \ldots, v_{C_m}\}$.

3-SAT is a restricted variant of SAT problem where all clauses of the input formula contain at most 3 literals. In the PLANAR 3-SAT, the incidence graph of the input formula is required to be planar. In the RESTRICTED PLANAR 3-SAT, it is further assumed that each variable occurs in exactly 3 clauses. Moreover, each variable occurs twice as a positive literal and exactly once as a negated literal. RESTRICTED PLANAR 3-SAT remains NP-hard (see [26, pf. Theorem 2a]).

1.2 Target Set Selection

In this section, we finally define the problem central to our work.

[6] That is, recursively, $f(\neg\varphi) := 1 - f(\varphi), f(\varphi \vee \psi) := \max\{f(\varphi), f(\psi)\}$, and $f(\varphi \wedge \psi) := \min\{f(\varphi), f(\psi)\}$ for all formulae φ and ψ.

▶ **Definition 1.29.** Let $G = (V, E)$ be a graph, $t: V \to \mathbb{N}$ a function (which we call the *threshold function*), and $S \subseteq V$. Consider the following activation process arising from S defined as follows:

$$S_0 = S,$$
$$S_{i+1} = S_i \cup \{v \in V \mid |N(v) \cap S_i| \geq t(v)\}.$$

We call S a *target set* (for G) (with respect to t) if and only if $S_n = V$.

It is not hard to see that for any $S \subseteq V$, the sets $S_0, S_1, S_2 \ldots$ form an ascending chain with respect to inclusion, i.e., $S_0 \subseteq S_1 \subseteq S_2 \subseteq \cdots$. Moreover, for all indices $i \geq 0$, either $S_{i+1} \setminus S_i \neq \emptyset$, or $S_{i+j} = S_i$ for all $j \geq 0$. In other words, in every iteration, either there is at least one new active vertex, or the process stabilizes and no new vertices become active in all subsequent iterations. As G has n vertices, the process stabilizes after at most n iterations. With this definition at hand, we can define decision variant of the Target Set Selection problem.

Target Set Selection (TSS)

Input: Graph $G = (V, E)$, a threshold function $t: V \to \mathbb{N}$, $k \in \mathbb{N}$

Task: Is there a target set $S \subseteq V$ for G with respect to t satisfying $|S| \leq k$?

We observe that Target Set Selection generalizes the Vertex Cover problem, in other words, Vertex Cover is a special case of Target Set Selection. This implies that Target Set Selection is necessarily NP-hard in the general case. This is a standard result about TSS and can be found in many works about Target Set Selection (see for instance [17]), but we include a proof for the sake of completeness.

▶ **Lemma 1.30.** *Let $G = (V, E)$ be a graph, $t: V \to \mathbb{N}$ given by $t(v) = \deg v$ and let $S \subseteq V$. Then S is a vertex cover of G if and only if S is a target set for G with respect to t.*

Proof. Suppose that S is a vertex cover. By Lemma 1.17, this is equivalent to the fact that for all vertices $v \in V$, either $v \in S$ or $N(v) \subseteq S$.

Now, consider the activation process arising from S. All vertices outside $S = S_0$ get activated in the first round of the process, i.e., $S_1 = S_2 = \ldots = S_n = V$, so S is also a target set for G with respect to t.

On the other hand, suppose that S is a target set for G with respect to t, and, for the sake of contradiction, suppose that it is not a vertex cover of G. This implies existence of an edge $\{u, v\} \subseteq V \setminus S$. Observe that u and v will never be activated as both of them have their threshold equal to their degree, thus u has to be activated before v and vice versa, but that is impossible. It follows that S must be a vertex cover. ∎

▶ **Corollary 1.31.** *Target Set Selection is NP-complete.*

Proof. First, we show that the problem belongs to the class NP. Given the underlying graph $G = (V, E)$, threshold function t, and $S \subseteq V$, we can verify, by simulating the process, that $S_n = V$ in polynomial time. Hence, Target Set Selection belongs to the class NP.

For the hardness, we reduce from the Vertex Cover problem. Given an instance (G, k) of the Vertex Cover problem, we construct an instance (G, t, k) of Target Set Selection as follows. We set $t(v) = \deg v$ for each $v \in V$. By Lemma 1.30, instances (G, k) and (G, t, k) are equivalent. ∎

In all future statements, we only deal NP-hardness as all our considered problems (including Target Set Selection) are in NP.

As already said in the introduction, due to NP-hardness of Target Set Selection in the general case, attempts were made to restrict the threshold function. Let us remind ourselves of the three commonly studied settings.

Unanimous thresholds – $t(v) = \deg v$ for all vertices v

Constant thresholds – $t(v) \leq c$ for all vertices v, and some fixed constant c

Majority thresholds – $t(v) = \left\lceil \frac{\deg v}{2} \right\rceil$ for all vertices v.

All three settings are NP-hard in the general case, as we will in fact prove in the next chapter. This is not a new result as already Dreyer and Roberts [29] showed hardness for all constants $c \geq 3$ and Chen [17] then extended this result to all constants $c \geq 2$. The proof for $c \geq 3$ can be carried out the same way as in the proof of Corollary 1.31, but instead reduce from Vertex Cover restricted to c-regular graphs. To one's disappointment, this reduction does not work for $c = 2$ as Vertex Cover is trivially solvable in the class of 2-regular graphs which is nothing else but a disjoint union of cycles. Hardness of Target Set Selection under the majority setting was first shown by Peleg [61]. We in fact include a different hardness proof for the majority setting in Section 2.3.

Although it seems that all three settings share the same complexity, there are graph classes where the complexity of these three settings varies significantly. Specifically, we demonstrate this for the class of grid graphs. In the unanimous threshold setting, Target Set Selection can be solved in polynomial time, whereas the other two settings remain NP-hard.

Chapter 2
Unit Disk Graphs

In this chapter, we provide the main results regarding the Target Set Selection problem restricted to the class of unit disk graphs. As a by-product of our theorems, we also obtain the full complexity picture in related classes, namely disk contact and grid graphs. The results are summarized in Table 2.1.

The chapter is organized into three parts, each of which addresses one of the three settings of Target Set Selection. In Section 2.1, we discuss the unanimous threshold setting. Section 2.2 delves into the constant threshold setting and finally, in Section 2.3, we cover the majority threshold setting.

2.1 Unanimous Thresholds

In this section, we provide the results about Target Set Selection in the unanimous threshold setting. That is, for each vertex v, we have $t(v) = \deg v$. We strongly rely on the equivalence between Target Set Selection and the Vertex Cover problem established in Lemma 1.30.

▶ **Theorem 2.1.** *Target Set Selection is NP-hard even if the underlying graph is a unit disk graph and all thresholds are unanimous.*

Proof. It is known that the Vertex Cover problem is NP-hard even on unit disk graphs [20]. In the same way as in proof of Corollary 1.31, we reduce from the Vertex Cover problem but restricted to unit disk graphs. Therefore, the theorem holds. ■

▶ **Theorem 2.2.** *Target Set Selection is NP-hard even if the underlying graph is planar and all thresholds are unanimous.*

Proof. It is known that Independent Set problem is NP-hard restricted to the class of 3-regular planar graphs [32]. By Corollary 1.22, the same applies for the Vertex Cover problem. In the same way as in proof of Corollary 1.31, we reduce from the Vertex Cover problem but restricted to the class of planar graphs. Therefore, the theorem holds. ■

■ **Table 2.1** Overview of our results. The first row contains individual restrictions of the threshold function, and the first column contains assumed graph classes. In the table, "NP-h" stands for "NP-hard", "P" stands for polynomial-time solvable cases. All results are complemented by the reference to the appropriate statement.

	constant	majority	unanimous	unrestricted
unit disk graphs	NP-h (Cor 2.28)	NP-h (Cor. 2.35)	NP-h (Thm. 2.1)	NP-h (Thm. 2.1)
grid graphs	NP-h (Thm. 2.27)	NP-h (Cor. 2.34)	P (Thm. 2.3)	NP-h (Thm. 2.27)
planar graphs (disk contact graphs)	NP-h (Thm. 2.18)	NP-h (Thm. 2.32)	NP-h (Thm. 2.2)	NP-h (Thm. 2.2)

NP-hardness of Target Set Selection in the unanimous threshold setting in the classes of planar and unit disk graphs reflects the NP-hardness of the Vertex Cover problem in these graph classes. However, the situation is different in the class of grid graphs. We make use of Kőnig's theorem, which makes the Vertex Cover problem tractable in the class of bipartite graphs. As grid graphs are also bipartite, we can use the same idea to find Vertex Cover efficiently in the class of grid graphs.

▶ **Theorem 2.3.** *Target Set Selection can be solved in time $O(n^{3/2})$ if the underlying graph is a grid graph and the thresholds are unanimous.*

Proof. By Lemma 1.30, Target Set Selection with unanimous thresholds is equivalent to the Vertex Cover problem so this reduces to deciding whether the input instance has a vertex cover of size at most k. Let $G = (V, E)$ be the input graph, which is a grid graph. Observe that grid graphs are also bipartite, so G is bipartite. We start by finding a maximum matching of G. This can be accomplished in $O(m\sqrt{n})$ time using the Hopcroft-Karp algorithm [39]. Now, by Kőnig's theorem, the size of a maximum matching in G is equal to the size of a minimum vertex cover of G. We can thus decide whether G has a vertex cover of size at most k, in other words, a target set of size at most k. Since grid graphs are sparse, the total running time is $O(m\sqrt{n}) = O(n\sqrt{n}) = O(n^{3/2})$. ∎

Note that the above algorithm can be easily modified to also output an optimal target set. To see this, notice that we are specifically finding a maximum matching in G, whose size is, by Kőnig's theorem, the same as the size of a minimum vertex cover of G. Any minimum vertex cover then corresponds to an optimal target set. It is possible to recover a minimum vertex cover from a given maximum matching in $O(n + m)$ time in bipartite graphs. See [11, Lemma 5.3.]. Thus, the time complexity remains $O(n^{3/2})$.

2.2 Constant Thresholds

In this section, we deal with the constant threshold setting. That is, all thresholds are bounded by a constant c. Recall that the problem is trivial when all thresholds are bounded by $c = 1$ since it suffices (and is also necessary) to target one vertex per each connected component (not containing a vertex of threshold 0) of the input graph.

We show that Target Set Selection is NP-hard when all thresholds are bounded by a constant $c \geq 2$ even if the underlying graph is a grid graph.

The section is divided into two parts. The first part establishes the complexity for the case where thresholds are bounded by a constant $c \geq 3$. The second part provides a complete picture of the problem's complexity by resolving the scenario even when the thresholds are bounded by a constant $c \geq 2$. We, in fact, show the hardness when the thresholds are at most 3 (and at most 2). Obviously, if the thresholds are bounded by a constant c, they are also bounded by all constants $c' \geq c$, so the hardness result for all constants is an obvious corollary to our theorems.

Note that the result for constant thresholds would hold solely by the proof for thresholds at most 2, although the part dealing with thresholds at most 3 includes an additional finding regarding the hardness of the Independent Set problem restricted to the class of regular unit disk graphs, which may be of independent interest.

Rectilinear embedding of planar graphs

Throughout the section, we will employ reductions from problems that involve planar graphs. To effectively apply such reductions, having some kind of "nice" representation of the planar graph will be essential. One such representation that is useful for our purposes is the so-called *rectilinear embedding.*

▶ **Definition 2.4.** Given a planar graph $G = (V, E)$, a *rectilinear embedding (of G)* is a planar drawing of G such that vertices occupy integer coordinates, and all edges are made of (possibly more) line segments of the form $x = i$ or $y = j$ (i.e., the line segments are parallel to the coordinate axes).

▶ **Example 2.5.** Recilinear embeddings of K_4 and $K_{2,3}$ are shown in Figure 2.1.

■ **Figure 2.1** A rectilinear embedding of K_4 (on the left) and $K_{2,3}$ (on the right).

It is obvious that if a planar graph admits a rectilinear embedding, its maximum degree must be at most 4. Due to the theorem of Valiant (Theorem 2.6), this condition is also sufficient. Moreover, the area of the embedding is polynomial in the size of the graph and the embedding can be computed in polynomial time with respect to the size of G. By the *area* of such embedding we mean the minimal area of a rectangle R with sides parallel to the coordinate axes such that the drawing is entirely inside R. For example, the area of the embedding of both graphs in Figure 2.1 is $3 \cdot 3 = 9$.

▶ **Theorem 2.6** (Valiant [68]). *Given a planar graph $G = (V, E)$ with maximum degree $\Delta G \leq 4$, there exists a rectilinear embedding of G with area at most $O(|V|^2)$. Moreover, this embedding can be computed in polynomial time with respect to the size of G.*

2.2.1 Case of thresholds bounded by 3

Note that we have already proved the hardness for Target Set Selection for planar graphs in the case of constant thresholds.

▶ **Corollary 2.7.** *Target Set Selection is NP-hard even if the underlying graph is planar and all thresholds are at most 3.*

Proof. Apply the same proof as in Theorem 2.2. Note that the input graph was 3-regular, so the thresholds are exactly 3, thus at most 3. ■

Our attention now turns to unit disk graphs. We begin by showing an auxiliary result that the Independent Set problem is NP-hard in the class of 3- and 4-regular unit disk graphs. We don't know anything about 5-regular unit disk graphs, see Remark 2.16. We then combine Corollary 1.22 together with Lemma 1.30 to finish the hardness proof for Target Set Selection.

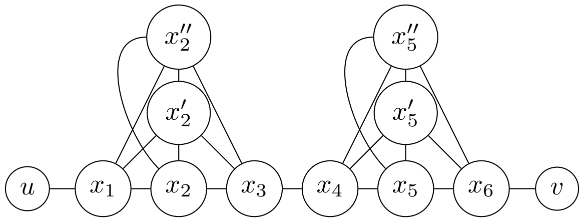
▶ **Theorem 2.8.** *For $r \in \{3, 4\}$ Independent Set is NP-hard even if the underlying graph is an r-regular unit disk graph.*

Proof. We reduce from Independent Set on r-regular planar graphs. As already noted in the proof of Theorem 2.2, this restriction of Independent Set remains NP-hard [32]. Let (G, k) be an instance of the Independent Set problem where $G = (V, E)$ is planar r-regular graph, where $r \in \{3, 4\}$. We first construct a rectilinear embedding of G, which exists by Theorem 2.6. We construct a new instance (G', k') of Independent Set where G' will be a unit disk graph.

We start from the graph G and we subdivide each edge $e = \{u, v\} \in E(G)$ exactly $6q_e$ times, creating a path $u, x_1, x_2, \ldots, x_{6q_e}, v$. The constant q_e is to be explained later. Next, for all $i \in [2q_e]$, we replace every vertex x_{3i-1} with a clique K_{r-1} and connect all its neighbors to the clique (see Figure 2.2). In other words, we create $r-2$ additional copies of the vertex x_{3i-1} and connect these copies into a complete graph (independently for each i). Note that the number q_e depends on the edge e. Let G' be the resulting graph. It is not hard to see that G' remains r-regular.

We now provide the unit disk representation for G' and show how to compute the numbers q_e in polynomial time for all edges $e \in E(G)$. We also show a polynomial upper bound on the numbers q_e. This is the content of the following technical lemma.

▶ **Lemma 2.9.** *The graph G' can be represented by unit disks and for each edge $e \in E(G)$ there exists a number q_e such that the number of vertices on the path from u to v created by the subdivision of e is equal to $6q_e$. Moreover, the number q_e can be computed in polynomial time, and it can be choosen small enough such that it is polynomially bounded by the size of G.*



■ **Figure 2.2** An example of a construction of path subdivision in the case of 4-regular graphs. In this case, $q_e = 1$.

Proof. We let $d = \frac{1}{7}$ be the diameter of the disks in the representation. Fix a rectilinear embedding for G. From now on, we will refer to this embedding simply as *the drawing*. Recall that the drawing has vertices in integral grid points and edges are made of (possibly more) line segments parallel to the coordinate axes. For an edge $e \in E(G)$, we denote by P_e the simple polygonal chain representing e in the drawing.

First, the vertices $v \in V(G')$ corresponding to vertices of G will have their disk at the corresponding grid point in the drawing.

We now show how to construct the subdivisions of the edges. We proceed independently for each edge $e = \{u, v\} \in E(G)$.[1] Let g denote the number of grid points contained in the polygonal chain P_e. Let these be $p_1, \dots, p_g$ as seen when walking from u to v along P_e. In particular, the point p_1 corresponds to the vertex u, and the point p_g corresponds to the vertex v. The remaining points are are refered to as *internal* points of the chain P_e. We place a disk centered at each of the internal points $p_2, \dots, p_{g-1}$. Let D_1 and D_g denote the disks corresponding to u and v, respectively, and $D_2, \dots, D_{g-1}$ the new disks centered at $p_2, \dots, p_{g-1}$. Our task is now to insert a certain number of disks in between D_i, D_{i+1} for all $i \in [g-1]$. Note that D_i and D_{i+1} are centered at neighboring grid points, i.e., their centers are at distance 1. Let w_i denote the number of inserted disks between D_i, D_{i+1}. We specify the numbers w_i later. Our task is to make the total number of disks in between D_1 nd D_r a multiple of 6, i.e., the number of disks shall be equal to $6q_e$. The total number of disks between $D_1, \dots, D_r$ is given by $y_e = g - 2 + \sum_{i=1}^{g-1} w_i$. Our aim is now to choose the numbers w_i such that $y_e = 6q_e$. To achieve this, we do the following. We first learn how to insert $\ell \in \{6, 7, 8, 9\}$ disks between a single pair of adjacent disks. We prove this in Lemma 2.10. We simplify the scenario and assume that D_i and D_{i+1} are centered at $p_i = (0, 0)$ and $p_{i+1} = (0, 1)$, respectively. It is not hard to generalize the idea to general points p_i, p_{i+1}.

▶ **Lemma 2.10.** *Let L be a line segment with endpoints $(0, 0), (1, 0)$ and let $\ell \in \{6, 7, 8, 9\}$. We can always place ℓ disks $E_1, \dots, E_\ell$ with diameters $d = \frac{1}{7}$ and centers $s_1, \dots, s_\ell$ all lying on L such that:*

i) $s_1 = (d, 0)$,

ii) $s_\ell = (1 - d, 0)$,

iii) any disk E_j intersects precisely its neighbors E_{j-1} and E_{j+1} (if they exist) and no other disks.

Proof. We prove this by construction and specify the centers of the ℓ disks. As all centers shall be on the line L, they are of the form $s_j = (a_j, 0)$. For a fixed ℓ and $j \in [\ell]$, the points a_j are given by the formula:

$$a_j = \frac{5j + \ell - 6}{7(\ell - 1)}.$$

It can be verified by a straightforward calculation that the properties i),ii) and iii) hold. To verify iii), it is enough to check that $a_{j+1} - a_j \leq d$ and $a_{j+2} - a_j > d$ for appropriate j. ■

Now we know how to insert $\ell \in \{6, 7, 8, 9\}$ disks. We show how many disks we have to insert such that $g - 2 + \sum_{i=1}^{g-1} w_i$ is a multiple of 6, given $g \geq 2$. In other words, we are now in the situation to choose the corresponding w_i, given $g \geq 2$. We prove this in Lemma 2.11.

[1]The variables introduced from this point should have another index e to indicate that they are in fact dependent on the edge e, however, for the sake of readability, we omit it.

▶ **Lemma 2.11.** *For any $g \geq 2$ we can pick $g-1$ numbers $w_1, \dots, w_{g-1}$ from the set $\{6,7,8,9\}$ such that*

$$g - 2 + \sum_{i=1}^{g-1} w_i = 0 \mod 6$$

.

Proof. We divide the proof into six cases according to the residue class of g modulo 6.

Case 1 If $g = 0 \mod 6$, we set $w_1 = 8$ and $w_i = 6$ for $i \in \{2, \dots, g-1\}$.

Case 2 If $g = 1 \mod 6$, we set $w_1 = 7$ and $w_i = 6$ for $i \in \{2, \dots, g-1\}$.

Case 3 If $g = 2 \mod 6$, we set $w_i = 6$ for all $i \in [g-1]$.

Case 4 If $g = 3 \mod 6$, we set $w_1 = 9, w_2 = 8$ and $w_i = 6$ for all $i \in \{3, \dots, g-1\}$. Note that $g \geq 3$ in this case.

Case 5 If $g = 4 \mod 6$, we set $w_1 = w_2 = 8$ and $w_i = 6$ for all $i \in \{3, \dots, g-1\}$. Note that $g \geq 3$ in this case as well.

Case 6 If $g = 5 \mod 6$, we set $w_1 = 8$ and $w_i = 6$ for all $i \in \{2, \dots, g-1\}$.

It is straightforward computation to verify that the chosen numbers w_i work in every case. ■

After this step, we know how to compute the number q_e for each edge. The formula for q_e is given by

$$q_e = \frac{1}{6}\left(g - 2 + \sum_{i=1}^{g-1} w_i\right).$$

The number g is given by the polygonal chain P_e and Lemma 2.11 tells us how to choose the numbers w_i. This computation can indeed be done in polynomial time. We now establish the promised polynomial upper bound on the number q_e. Observe that the number of grid points contained in P_e is at most the area of the drawing. i.e., by using Theorem 2.6, $g \leq c|V|^2$ for some constant c. Further, by the construction, we have $w_i \leq 9$ for all any $g \geq 2$ and $i \in [g-1]$. Thus, we have

$$q_e \leq \frac{1}{6}\left(g - 2 + 9(g-1)\right) \leq \frac{1}{6}\left(10g - 11\right) \leq \frac{10}{6} g \leq \frac{10}{6} c|V|^2 = O(|V|^2),$$

as promised.

We thus constructed the subdivision of the edges. What is left is to show how to represent the cliques at the vertices x_{3i-1} for $i \in [2q_e]$. We simply create $r-2$ copies for the case of K_{r-1}. It is not hard to see that this precisely corresponds to replacing a vertex with a clique.

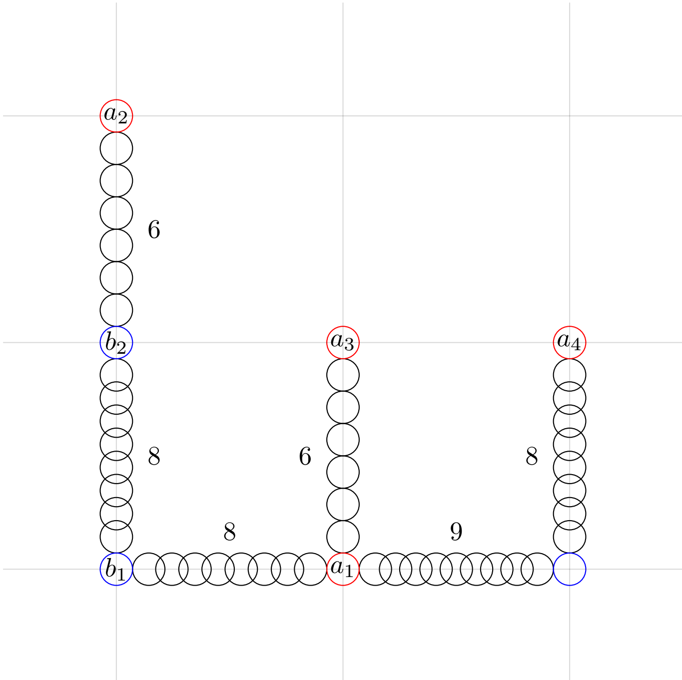
Note that in Lemma 2.10 we placed the disks in between D_i and D_{i+1} starting from $s_1 = (d, 0)$ and ending at $s_\ell = (1-d, 0)$. This implies that for any edge $e = \{u, v\}$, the disks adjacent to D_1 and D_r will not intersect any other disks representing other subdivided edges (in particular, those with endpoints u or v)

This completes the construction of unit disk representation for G'. Refer to Figure 2.3 for an example of the construction. ■

Finally, to finish the construction of the instance (G', k'), we set $k' = k + \sum_{e \in E} 3q_e$. We now establish the equivalence between the instances (G, k) and (G', k').

▶ **Claim 2.12.** *If (G, k) is a yes-instance, then (G', k') is a yes-instance.*

Proof. Assume that (G, k) is a yes-instance of Independent Set, and let S be an independent set in G of size at least k, i.e., $|S| \geq k$. We build an independent set S' of size at least k'. We first add all vertices from S to S'. Note that G' contains all original vertices of G by construction. Next, process all edges independently in arbitrary order as follows. Let $e = \{u, v\} \in E(G)$ be an edge. Since S was an independent set, we have $u \notin S$ or $v \notin S$ (i.e., in S'). We distinguish these two cases.



■ **Figure 2.3** Example of a construction of the unit disk representation of the graph G' from the proof of Theorem 2.8. The original graph G was a star $K_{1,3}$ with vertices embedded at a_1, a_2, a_3, a_4. The proof of Theorem 2.8 originally started with an r-regular graph, however, for the sake of simplicity, we show the reduction on a simpler graph. The blue disks correspond to the internal grid points contained in the polygonal chains representing the edges. These are the disks $D_2, \dots, D_{g-1}$ for the corresponding edges. Consider the edge $e = \{a_1, a_2\} \in E(G)$. The red disks at a_1 and a_2 are the disks D_1 and D_4, respectively, and the blue disks at b_1 and b_2 are the disks D_2 and D_3, respectively. The black disks correspond to the disks E_j from Lemma 2.11. The numbers correspond to the numbers w_i (and are equal to the number of black disks E_j between blue and red disks). For the edge e we have $g = 4$ grid points contained in the polygonal chain, thus we are in the case $g = 4 \mod 6$ in Lemma 2.11. Thus, we are setting $w_1 = w_2 = 8$ and $w_3 = 6$ for this particular edge. The total number of disks on the subdivided edge e is thus $2 + 8 + 8 + 6 = 24 = 0 \mod 6$ and we have $q_e = 4$.

Case 1 If $u \notin S$, we add the vertices $\{x_{2i-1} \mid i \in [3q_e]\}$ to S'.

Case 2 If $v \notin S$, we add the vertices $\{x_{2i} \mid i \in [3q_e]\}$ to S'.

It is not hard to see that S' is independent after each step. For each edge, we added exactly $3q_e$ vertices, so $|S'| \geq k + \sum_{e \in E} 3q_e = k'$. It follows that (G', k') is a yes-instance. ■

▶ **Claim 2.13.** *If (G', k') is a yes-instance, then (G, k) is a yes-instance.*

Proof. Assume that (G', k') is a yes-instance and let S' be an independent set in G' of size at least k', i.e., $|S'| \geq k' = k + \sum_{e \in E} 3q_e$. We create an independent set $S \subseteq V(G)$ as follows. We start with S' and we will be removing some vertices. We process the edges $e \in E(G)$ in arbitrary order and we do the following. Let $e = \{u, v\} \in E$ be an edge in the original graph and let $u, x_1, x_2, \dots, x_{6q_e}, v$ be its subdivision. There are two cases to consider.

Case 1 Both u and v are in S'. In this case, $|S' \cap \{x_1, \dots, x_{6q_e}\}| \leq 3q_e - 1$ by the pigeonhole principle. We choose u or v and remove one of them from S' to make the resulting set independent in G. We also remove all the x_i's that were in S'.

Case 2 At least one of u and v is not in S'. In this case, $|S' \cap \{x_1, \dots, x_{6q_e}\}| \leq 3q_e$ by the pigeonhole principle. In this case, we only remove all the x_i's from S'.

Note that in both cases, we removed at most $3q_e$ vertices for each edge. The new set consist exclusively of the vertices of the original graph G and is of size at least $k' - \sum_{e \in E(G)} 3q_e = k$. Note that if $\{u, v\} \in E(G)$

is an edge and both u and v were in S', we removed one of u, v from S', so $|S \cap \{u, v\}| \leq 1$. It follows that S is independent, thus (G, k) is a yes-instance. ■

It remains to say that the reduction is polynomial since the rectilinear embedding can be computed in polynomial time by Theorem 2.6 and we further added at most $O(\sum_{e \in E} 3q_e)$ new vertices and edges. However, the numbers q_e are polynomially bounded in the size of G and can be computed in polynomial time by Lemma 2.9. This completes the proof of Theorem 2.8. ■

We can extend this result for infinitely many r. Given instance of INDEPENDENT SET where the underlying graph is r-regular we replace each vertex v by a clique K_q and join all its neighbors to all the vertices of the K_q. In other words, create $q-1$ additional copies $v_1, \ldots, v_{q-1}$ of the vertex v and connect all pairs v_i, v_j with $i \neq j$ by an edge. This is the same construction as in proof of Theorem 2.8. In this way we get a $(q(r+1)-1)$-regular graph. Denote this new graph by $\widehat{G}_q$. Note that for each independent set in G there is an independent set of the same in $\widehat{G}_q$ and vice versa. This implies that the instances (G, k) and $(\widehat{G}_q, k)$ of INDEPENDENT SET are equivalent. In the language of (unit) disk graphs, replacing each vertex with a clique K_q is the same as creating $q-1$ additional copies of every disk in the original representation. We thus proved the following interesting corollary.

▶ **Corollary 2.14.** *If INDEPENDENT SET is NP-hard on the class of r-regular graphs, then INDEPENDENT SET is NP-hard in the class of $(q(r+1)-1)$-regular graphs for any positive integer q.*

We explicitly showed NP-hardness of INDEPENDENT SET in r-regular unit disk graphs for $r \in \{3, 4\}$ in Theorem 2.8. As a by-product, we get the following result.

▶ **Corollary 2.15.** *INDEPENDENT SET is NP-hard even if the underlying graph is an r-regular unit disk graph where r is positive integer and $r \equiv -1 \mod 4$ or $r \equiv -1 \mod 5$.*

▶ **Remark 2.16.** We remark that we don't know if INDEPENDENT SET is NP-hard for all r-regular unit disk graphs, $r \geq 3$. Certainly, for $r = 1, 2$ the problem is in P. The first r unknown to us is $r = 5$. It is known that INDEPENDENT SET on 5-regular planar graphs is NP-hard [3], however, our reduction is not applicable, since we need $\Delta G \leq 4$ to apply the rectilinear embedding.

To prove hardness for *all* constants r, one can't apply the approach we used here to show that for infinitely many constants the problem is hard. That is, prove it explicitly for finitely many base cases $r_1, \ldots, r_k$, and then apply Corollary 2.14 (we proved explicitly $r_1 = 3, r_2 = 4$). To see this, note that for any such base cases $r_1, \ldots, r_k$, we can pick a large enough prime number p with $p > r_i + 1$ for all $i \in [k]$. Observe that explicitly proving hardness for r_i-regular graphs implies the hardness for r-regular graphs with $r = -1 \mod r_i + 1$. Now, the NP-hardness for $r = p - 1$ is not implied by the base cases. To see this, note that this would imply that for some j we have $p - 1 = -1 \mod r_j + 1$, which in turn implies $p = 0 \mod r_j + 1$. But that is impossible since p was a prime number and $p > r_j + 1$.

We now return to the TARGET SET SELECTION problem and utilize the previous results to show NP-hardness of TARGET SET SELECTION in the class of unit disk graphs in the constant threshold setting for $c \geq 3$.

▶ **Theorem 2.17.** *TARGET SET SELECTION is NP-hard even if the underlying graph is a unit disk graph and all thresholds are at most 3.*

Proof. By Theorem 2.8, INDEPENDENT SET is NP-hard when restricted to the class of 3-regular unit disk graphs. By Corollary 1.22, the same hardness result holds for VERTEX COVER. As in the proof of Corollary 1.31 we reduce from VERTEX COVER but restricted to 3-regular unit disk graphs. Therefore, the theorem holds. ■

2.2.2 Case of thresholds bounded by 2

Proof for thresholds at most 2 relies on a reduction from the RESTRICTED PLANAR 3-SAT problem. First, we establish the hardness for planar graphs with $\Delta G \leq 4$ (Theorem 2.18). We then utilize this reduction to show NP-hardness for the classes of grid graphs and unit disk graphs.

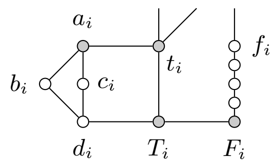
Planar Graphs

▶ **Theorem 2.18.** *Target Set Selection is NP-hard even when the underlying graph is planar with maximum degree $\Delta G \leq 4$ and all thresholds are at most 2.*

Proof. We reduce from the Restricted Planar 3-Sat problem. Let φ be the input formula with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$. The reduction consists of two types of gadgets:

Variable gadget Given a variable x_i, the variable gadget for x_i is the planar graph depicted in Figure 2.4. We refer to this gadget as X_i. The notable vertices of the gadget are T_i, F_i, t_i, and f_i. The idea is that the vertices T_i and F_i stand for the truth assignment of this particular variable, while the vertices t_i and f_i represent the positive and negative literals, respectively, and serve to connect the variable gadgets with the respective clause gadgets. Note that by the definition of Restricted Planar 3-Sat we have $\deg t_i = 4$ and $\deg f_i = 2$.

Clause gadget Given a clause C_j, the clause gadget for C_j consists of a single vertex y_j which is connected to the corresponding literal vertices that are contained in the clause C_j. We refer to this gadget as Y_j.



■ **Figure 2.4** Schematic representation of the variable gadget X_i for a variable x_i. The gray vertices have threshold 2, while the white vertices have threshold 1. Note also that the half-edges illustrate the fact that the gadget is connected with the rest of the graph only via t_i and f_i.

We are now ready to construct an instance (G, t, k) of Target Set Selection. Start with the incidence graph G_φ. For every variable x_i, we replace the vertex v_{x_i} by the variable gadget X_i and we identify each clause vertex v_{C_j} with the vertex y_j, i.e., with the gadget Y_j. Next, we connect all literal vertices of X_i with the corresponding clause gadgets. More precisely, we add an edge $\{t_i, y_j\}$ into $E(G)$ if x_i occurs as a positive literal in the clause C_j, and we add an edge $\{f_i, y_j\}$ to $E(G)$ if x_i occurs as a negative literal in the clause C_j.

It remains to set the thresholds and k. For the variable gadget, the gray vertices have threshold equal to 2, while the white vertices have threshold equal to 1. In the clause gadget, we set $t(y_j) = 1$. Finally, we set $k = n$.

Observe that G is a planar graph. To see this, note that we can start with a planar drawing of G_φ and replace the vertices of G_φ with the gadgets. Note that the only problem could be with the edges coming from the vertices t_i and f_i. However, for a variable x_i occuring in the clauses $C_{j_1}, C_{j_2}, C_{j_3}$, no matter what the order[2] of the vertices $y_{j_1}, y_{j_2}, y_{j_3}$ is (with respect to the planar drawing of G_φ), it is always possible to draw the edges from t_i and f_i to the corresponding clause gadgets in such a way that we don't create any crossings. For example, the edges going from t_i can encircle the entire gadget in the drawing and leave the gadget to the right of the edge coming from f_i.

Moreover, we have $\Delta G \leq 4$, and thresholds are at most 2, as promised.

Before we show equivalence of the instances (G, t, k) and φ, we establish some basic properties of the variable gadget. Properties of the clause gadget are clear, since it is a single vertex with threshold 1.

▶ **Lemma 2.19.** *The gadget X_i has following properties:*

i) If the vertex T_i is active, then after 4 rounds, the vertices a_i, b_i, c_i, d_i, t_i are necessarily active.

ii) If the vertex F_i is active, then after 4 rounds, all vertices on the f_i-F_i-path inside X_i are necessarily active.

[2] More precisely, the order of the points corresponding to the vertices $y_{j_1}, y_{j_2}, y_{j_3}$ given by radially sorting them around the point corresponding to the vertex v_{x_i} in the original drawing.

iii) Even if the vertices in $N[V(X_i)] \setminus V(X_i)$ are active and no other vertex inside $V(X_i)$ is active, then vertices T_i and F_i will never be active.

iv) If the vertex F_i is active and t_i becomes active, then all vertices in X_i are eventually active.

v) If the vertex T_i is active and f_i becomes active, then all vertices in X_i are eventually active.

Proof. The claims i) and ii) are clear from the construction of the gadget.

For claim iii), observe that if the neighbors of t_i and f_i outside $V(X_i)$ are active, then t_i becomes active. However, $t(a_i) = t(T_i)$, so vertices a_i and T_i never become active. If f_i is also active, it only activates the neighbor of F_i with threshold 1 but never F_i itself, since $t(F_i) = 2$ and T_i never become active.

For claim iv), suppose that F_i is active and t_i becomes active in round r. Then, in round $r+1$ the vertex T_i becomes active. In round $r+2$, the vertex d_i becomes active. Next, in round $r+3$, the vertices b_i and c_i become active. Finally, in round $r+4$, the vertex a_i becomes active. Also, all vertices on the path from F_i to f_i become active during these rounds (if they are not already activated). Thus, all vertices in X_i are active.

For claim v), suppose that T_i is active and f_i becomes active in round r. After 4 rounds, the vertex F_i becomes active and similarly as in the proof for iv), observe that the remaining vertices a_i, b_i, c_i, d_i and t_i become active. Thus, all vertices in X_i are active. ■

We now establish the equivalence between the formula φ and the constructed instance (G, t, k).

▶ **Claim 2.20.** *If φ is satisfiable, then (G, t, k) is a yes instance of TSS*

Proof. Let φ be satisfiable and let f be a satisfying assignment. We create a target set S as follows. For each variable x_i we add either T_i if $f(x_i) = 1$ or F_i if $f(x_i) = 0$. Observe that $|S| = n = k$. It remains to show that S is a target set.

To see this, observe that by the properties i) and ii) every T_i and F_i activates the corresponding t_i or f_i (respectively) in 4 rounds. In the fifth round, all clauses become active. Indeed, because f is a satisfying assignment, all of them become active. In the sixth round, the vertices f_i or t_i become active. More precisely, if $T_i \in S$, then f_i becomes active in the sixth round (and vice versa if $F_i \in S$, then t_i becomes active in the sixth round). By the properties iv) and v), the remaining vertices of all the variable gadgets become active. Thus, S is a target set. It follows that (G, t, k) is a yes-instance. ■

▶ **Claim 2.21.** *If (G, t, k) is a yes-instance of TSS, then the formula φ is satisfiable.*

Proof. Suppose that (G, t, k) is a yes-instance of TARGET SET SELECTION and let $S \subseteq V(G)$ be a target set for G of size at most k. We first make several claims about the structure of S.

▶ **Claim 2.22.** *For every variable gadget X_i we have $S \cap V(X_i) \neq \emptyset$.*

Proof. Suppose otherwise, i.e., let X_i be a variable gadget such that $S \cap V(X_i) = \emptyset$. Note that by the property iii) of the variable gadget, the vertices T_i and F_i never become active even if the vertices in $N[V(X_i)] \setminus V(X_i)$ are active. This contradicts the assumption that S is a target set. ■

By Claim 2.22, S must contain at least one vertex from the variable gadget. By the definition of k, there is at most one vertex of the gadget X_i inside S. Putting this together, we have the following claim.

▶ **Claim 2.23.** *For all $i \in [n]$ we have $|S \cap V(X_i)| = 1$.*

Let $u_i \in S \cap V(X_i)$ be the unique vertex for the i-th variable gadget. We now argue that we can assume without loss of generality that $u_i \in \{F_i, T_i\}$.

▶ **Claim 2.24.** *There is a target set S' satisfying: for all $i \in [n]$ we have $S' \cap \{T_i, F_i\} \neq \emptyset$ and $|S'| = |S|$.*

Proof. Process the variable gadgets independently one by one. Formally,[3] start with $S^0 = S$ and inductively build the sets S^i for $i \in [n]$ by processing the gadgets. We let $S' := S^n$. Let $i \geq 1$.

For the i-th gadget, if $S^{i-1} \cap \{T_i, F_i\} \neq \emptyset$, there is nothing to do, i.e., we set $S^i = S^{i-1}$. Otherwise, observe that $S \cap V(X_i) = S_{i-1} \cap V(X_i)$. Recall that u_i was the unique vertex in $S \cap V(X_i)$. We distinguish two cases.

Case 1 The vertex u_i lies on the f_i-F_i-path in X_i. Note that we can replace u_i by F_i and this does not change the fact that u_i eventually becomes active by the property ii). I.e., in this case, we set $S^i = S^{i-1} \setminus \{u_i\} \cup \{F_i\}$.

Case 2 The vertex u_i satisfies $u_i \in \{a_i, b_i, c_i, d_i, t_i\}$. Note that we can replace u_i by T_i and this does not change the fact that u_i eventually becomes active by the property i). I.e., in this case, we set $S^i = S^{i-1} \setminus \{u_i\} \cup \{T_i\}$

This finishes the proof. ∎

By Claim 2.24 we have, without loss of generality, $u_i \in \{F_i, T_i\}$ for all $i \in [n]$. We proceed to construct a satisfying assignment f for φ in the obvious way. For a variable x_i we set $f(x_i) = 0$ if $u_i = F_i$, and $f(x_i) = 1$ otherwise (i.e., if $u_i = T_i$). What is left is to show that f is indeed a satisfying assignment for φ.

▶ **Claim.** *f is a satisfying assignment for φ.*

Proof. For the sake of contradiction, suppose that that f is not a satisfying assignment. By Observation 1.26, this is equivalent to the existence of a clause C_j with $f(C_j) = 0$. More precisely, for all literals $\ell \in C_j$ we have $f(\ell) = 0$.

Without loss of generality, assume that $|C_j| = 2$ and that C_j contains one positive and one negative literal. Any other case can be proven analogously. Without loss of generality, let $C_j = \neg x_1 \vee x_2$ (i.e., let C_j consist of the first two variables, otherwise permute the names of variables accordingly). By assumption, we have $f(x_1) = 1$ and $f(x_2) = 0$. That is, $S \cap V(X_1) = \{T_1\}$ and $S \cap V(X_2) = \{F_2\}$.

Note that $y_j \notin S$ because otherwise there is a variable gadget X_i with $S \cap X_i = \emptyset$, which is impossible by Claim 2.22. Since S is a target set and $y_j \notin S$, there must be a round r in which one of the neighbors of y_j becomes active. We have $N(y_j) = \{f_1, t_2\}$. We show that this is impossible. More precisely, we show that neither of f_1 and t_2 become active.

We have $S \cap V(X_1) = \{T_1\}$. Observe that in order to make the vertex f_1 active, it is necessary to have at least one vertex from the path from f_1 to F_1 in the target set S. This implies that $|V(X_1) \cap S| \geq 2$, which contradicts Claim 2.23. Thus f_1 is never active.

Analogously, we have $S \cap V(X_2) = \{F_2\}$. Observe that in order to have t_2 active, then, since $t(t_2) = 2$ and one edge from t_2 outside $V(X_2)$ leads to y_j, we need at least one of $\{a_i, T_i\}$ to be active. Observe that this implies that one of $a_2, b_2, c_2, d_2, T_2, t_2$ must be in the initial target set S, otherwise t_2 is never active. However, this implies that $|S \cap V(X_2)| \geq 2$ which again contradicts Claim 2.23. Thus t_2 is never active either.

Putting this together, we observe that y_j never becomes active, thus contradicting the assumption that S was a target set. ∎

This finishes the proof of Claim 2.21. ∎

To conclude the proof of Theorem 2.18, it remains to combine Claims 2.20 and 2.21 and notice that the reduction is indeed polynomial since G has exactly $m + 11n$ vertices. ∎

Grid graphs and Unit Disk Graphs

In the previous section we showed hardness of Target Set Selection when the underlying graph is restricted to be planar with maximum degree at most 4 and the thresholds are at most 2. We utilize this result to show the hardness in the same setting for the class of grid graphs.

Let us begin with few observations about how graph subdivisions affect target sets.

[3] We use superscripts to avoid confusion with the activation process (Definition 1.29)

▶ **Observation 2.25.** *Let $G = (V, E)$ be a graph and $t\colon V \to \mathbb{N}$ a threshold function, S a target set, and let $v \in S$ be a vertex with $t(v) \leq 1$ and $\deg v \geq 1$. $S \setminus \{v\} \cup \{u\}$ is also a target set for any $u \in N(v)$.*

▶ **Observation 2.26.** *Let $G = (V, E)$ be a graph and $t\colon V \to \mathbb{N}$ a threshold function and let $e \in E$. Let G' be a graph that results from G by subdividing an edge e once and creating new vertex $v' \notin V$. Let $t'\colon V(G') \to \mathbb{N}$ be defined by $t'(v') = 1$ and $t'(v) = t(v)$ for $v \neq v'$. Then the following holds:*

i) If S is a target set for G with respect to t, then S is also a target set for G' with respect to t'.

ii) If S' is a target set for G' with respect to t', then there exists target set S for G with respect to t and $|S| = |S'|$.

▶ **Theorem 2.27.** *Target Set Selection is* NP*-hard even if the underlying graph is a grid graph and all thresholds are at most* 2*.*

Proof. We reduce from Target Set Selection on planar graphs with maximum degree 4 and thresholds at most 2. Hardness of this setting is implied by Theorem 2.18. Let (G, t, k) be an instance of TSS where G is planar and $\Delta G \leq 4$. Fix a rectilinear embedding of G which exists by Theorem 2.6. We refer to the rectilinear embedding simply as *drawing*. We now modifiy the graph G as follows. For an edge $e \in E(G)$, consider the polygonal chain P_e representing e in the drawing. Let g denote the number of grid points contained in P_e. We subdivide the edge e exactly $g - 2$ times (see Figure 2.5). Note that the case $g = 2$ vacuously corresponds to no subdivision. After this step, the graph is (not necessarily induced) subgraph of a grid. To make it induced, we further simultaneously subdivide all edges exactly once (see Figure 2.6). After this step, the resulting graph is indeed an induced subgraph of a grid (i.e., a grid graph). We set the thresholds of all newly created vertices to 1. Let G' denote the resulting graph, $t'\colon V(G') \to \mathbb{N}$ the new threshold function and set $k' = k$.

▶ **Claim.** *(G, t, k) is a yes-instance of Target Set Selection if and only if (G', t', k') is a yes-instance of Target Set Selection.*

Proof. Let (G, t, k) be a yes-instance and let $S \subseteq V(G)$ be a target set of size at most k. Inductively for each subdivision, apply i) from Observation 2.26. It follows that S is also a target set with respect to t' and is of size at most $k = k'$, thus (G', t', k') is a yes-instance.

On the other hand, let (G', t', k') be a yes-instance and let $S' \subseteq V(G')$ be a target set of size at most k'. Inductively for each subdivision, apply ii) from Observation 2.26. Observe that in each step we get a target set S with the same size. It follows, that there is a target set S with respect to t and is of size $k' = k$, thus (G, t, k) is a yes-instance. ■

To finish the proof, we notice that the rectilinear embedding can be computed in polynomial time by Theorem 2.6 and its area is at most $O(|V|^2)$. It follows that in both steps of the construction, we only added at most $O(|V|^2)$ many new vertices, thus the size of G' is at most polynomial in the size of G. In other words, the reduction is polynomial. The theorem follows.

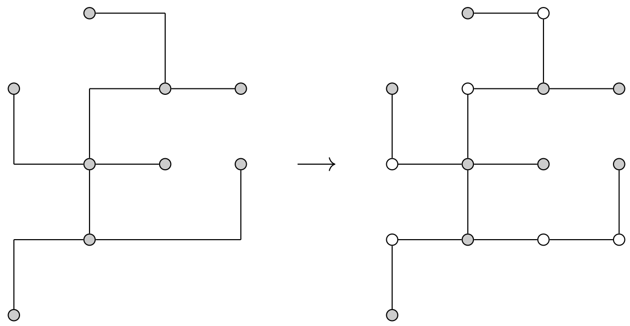
■

As the class of grid graphs is a subclass of the unit disk graphs, we obtain NP-hardness for the unit disk graphs as a corollary of Theorem 2.27.

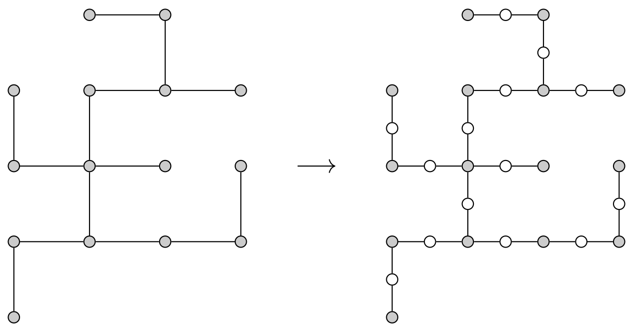
▶ **Corollary 2.28.** *Target Set Selection is* NP*-hard even when the underlying graph is a unit disk graph and all thresholds are at most* 2*.*

2.3 Majority Thresholds

In this section we provide results about Target Set Selection in the majority threshold setting in discussed graph classes. Recall that in this setting we have $t(v) = \left\lceil \frac{\deg v}{2} \right\rceil$ for each vertex v. Before delving into the specific classes of graphs discussed in this work, we first examine how the general case (i.e., when the underlying graph is unrestricted) is proven to be hard. This is content of Theorem 2.29. Initially, the first proof of hardness in this setting is due to Peleg [61]. The proof provided here is slightly inspired by the proof of a related result concerning the inapproximability of the Target Set Selection problem given by Chen [17].



■ **Figure 2.5** Transformation of a planar graph with maximum degree 4 into a subgraph of a grid by subdividing edges that fail to connect two neighboring grid points. Filled vertices correspond to the vertices of the original graph and the empty ones are the newly created vertices.



■ **Figure 2.6** Transformation of graph, that is (not necessarily induced) subgraph of a grid into a graph that is induced subgraph of a grid (i.e., a grid graph) by subdividing all edges exactly once. Filled vertices correspond to the vertices of the original graph and the empty ones are the newly created vertices.

▶ **Theorem 2.29.** *Target Set Selection remains* NP*-hard under the majority threshold setting.*

Proof. We know that Target Set Selection is NP-hard when the threshold function is unrestricted. We reduce from TSS with unrestricted thresholds. Let (G, t, k) be an instance of TSS. We create new instance (G', t', k') as follows. For each vertex v with $t(v) \neq \left\lceil \frac{\deg_G v}{2} \right\rceil$, we perform the following:

Case 1 If $t(v) > \left\lceil \frac{\deg_G(v)}{2} \right\rceil$, we add $2t(v) - \deg_G(v)$ new vertices with threshold 1 incident to v and set $t'(v) = \left\lceil \frac{\deg_{G'}(v)}{2} \right\rceil = t(v)$

Case 2 If $t(v) < \left\lceil \frac{\deg_G(v)}{2} \right\rceil$ we add $\deg_G(v) - 2t(v)$ cherry gadgets (see Figure 2.7) and attach them to v as depicted in Figure 2.7. We set the threshold of the vertices in the gadget to be at majority, that is $t'(g^\ell) = t'(g^r) = 1$ and $t(g^m) = 2$. We increase the threshold of v by $\deg_G(v) - 2t(v)$, i.e., we set $t'(v) = t(v) + \deg_G(v) - 2t(v) = \left\lceil \frac{\deg_{G'}(v)}{2} \right\rceil$.

Let $V_1 \subseteq V(G')$ denote the vertices added in the case 1 of the construction. Let α denote the number of cherries added in the construction, and let g_i^ℓ, g_i^r, g_i^m be the three vertices of the i-th added cherry.

Finally, set $k' = k + \alpha$.

▶ **Claim 2.30.** *(G, t, k) is a yes-instance of Target Set Selection if and only if (G', t', k') is a yes-instance of Target Set Selection.*

Proof. Let (G, t, k) be a yes-instance and let $S \subseteq V(G)$ be a solution with $|S| \leq k$. We claim that $S' = S \cup \{g_i^m \mid i \in [\alpha]\}$ is a solution for (G', t', k'). Certainly, $|S'| \leq k + \alpha = k'$. First, we check that all original vertices indeed get activated. The only vertices that had their threshold values changed were the vertices $v \in V(G)$ for which $t(v) < \left\lceil \frac{\deg_G(v)}{2} \right\rceil$. We attached exactly $\deg_G(v) - 2t(v)$ cherries to these vertices and increased their threshold by exactly $\deg_G(v) - 2t(v)$. However, the cherries will become active because $g_i^m \in S'$ and only $t'(v) - \deg_G(v) - 2t(v) = t(v)$more neighbors of v need to be active in order for v to be active. But that corresponds to the original activation process arising from S in G. It follows that S' is a target set for G' with respect to t', thus (G', t', k') is a yes-instance.

On the other hand, let (G', t', k') be a yes instance and $S' \subseteq V(G')$ a solution with $|S'| \leq k'$.

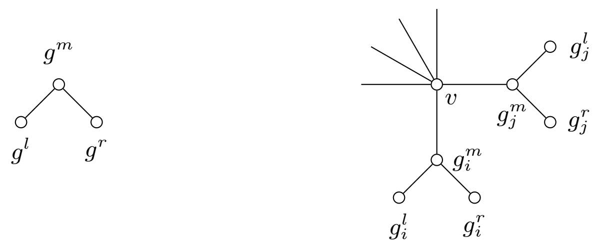
▶ **Claim 2.31.** *For each $i \in [\alpha]$ we have $S' \cap \{g_i^\ell, g_i^m, g_i^r\} \neq \emptyset$.*

Proof. Suppose for the sake of contradiction that there is an index i such that $S' \cap \{g_i^\ell, g_i^m, g_i^r\} = \emptyset$. Observe that the vertex g_i^m has exactly one neighbor outside the cherry gadget. It follows that it will never become active. This contradicts the fact that S' is a target set. ■

We may further assume that S' contains no vertices v with $\deg v \geq 1$ and $t(v) \leq 1$ by Observation 2.25. In particular, we can assume that $S' \cap V_1 = \emptyset$. Denote the set of vertices inside all cherries by V'. That is, $V' = \bigcup_{i=1}^{\alpha} \{g_i^\ell, g_i^m, g_i^r\}$ and let $S = S' \setminus V'$. The task is now to show that S is a valid solution to (G, t, k). Since we assumed that S' does not contain vertices from V_1, and we removed all vertices from the cherries, S indeed contains only vertices of G. Now, we argue that $|S| \leq k$. By Claim 2.31, S' contains at least one vertex from each cherry and since the cherries are pairwise vertex-disjoint, we have $|S' \cap V'| \geq \alpha$. It follows that $|S| = |S' \setminus V'| \leq k' - \alpha = k$. It remains to prove that S is a target set for G with respect to t. Similarly as in the proof of the opposite direction, the only interesting vertices are those, for which $t(v) < \left\lceil \frac{\deg_G(v)}{2} \right\rceil$. In this direction (when going from G' to G), we decreased their threshold by $\deg_G(v) - 2t(v)$ but that is also the number of neighbors we removed from v in G. Thus, activation of these vertices remains unchanged.

This finishes the proof of Claim 2.30. ■

To finish the proof of Theorem 2.29 it remains to notice that the reduction is indeed polynomial since we added at most $3 \cdot \deg_G(v) \leq 3|V(G)|$ vertices for each vertex $v \in V(G)$, i.e., we added at most $3|V(G)|^2$ new vertices. ■



■ **Figure 2.7** The cherry gadget (on the left). Connection of two cherry gadgets to a vertex $v \in V(G)$ with original degree $\deg_G(v) = 4$ and original threshold $t(v) = 1$. The new threshold of v is $t'(v) = t(v) + \deg_G(v) - 2t(v) = 3$ and $\deg_{G'}(v) = 6$, thus it is at majority. The half edges going from v represent connection of v to the rest of G.

▶ **Remark.** We remark that we didn't actually have to mess with the cherry gadgets. Instead, we could have started the reduction with a hard instance, where for all vertices v we have $t(v) > \left\lceil \frac{\deg_G(v)}{2} \right\rceil$. For example, in the proof of Theorem 2.2 the underlying graph in the hard instance of Target Set Selection is 3-regular, and the thresholds are exactly 3. This means that only the first case from proof of Theorem 2.29 would apply and the proof would be much simpler.

Although the cherry gadgets were not necessary, we now use them to show hardness of Target Set Selection under the majority setting for our desired graph classes. We start with the planar graphs.

▶ **Theorem 2.32.** *Target Set Selection is NP-hard under the majority threshold setting even when the underlying graph is planar with maximum degree $\Delta G \leq 4$.*

Proof. The proof combines ideas from proofs of Theorem 2.18 and Theorem 2.29. We reduce from the Restricted Planar 3-Sat problem in a similar way as in proof of Theorem 2.18 and fix the thresholds of the vertices that are not equal to the majority in a similar fashion as in the proof of Theorem 2.29. Let φ be the input formula with variables $x_1, \ldots, x_n$ and clauses $C_1, \ldots, C_m$.

Recall the original variable gadget (see Figure 2.4). Again, we have $\deg t_i = 4$ and $\deg f_i = 2$. Observe that all vertices except d_i and F_i have majority thresholds. We fix the threshold value of the vertices d_i, F_i.

To fix the vertex d_i, because we have $t(d_i) < \left\lceil \frac{\deg d_i}{2} \right\rceil$, we attach one cherry to d_i (and increase the threshold of d_i by 1). This follows case 2 in the proof of Theorem 2.29 because $\deg d_i - 2t(d_i) = 3-2 = 1$. To fix vertex F_i, because we have $t(F_i) > \left\lceil \frac{\deg F_i}{2} \right\rceil$, we add a leaf adjacent to F_i with threshold 1. The modified variable gadget is depicted in Figure 2.8.

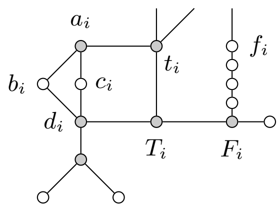
Now comes the clause gadget. Recall that in the Restricted Planar 3-Sat all clauses are of size at most 3. In our construction, this means that for the clause C_j and the original clause gadget Y_j consisting of a single vertex y_j we have $\deg y_j \in \{1, 2, 3\}$. We also had $t(y_j) = 1$. Thus, if $\deg y_j \leq 2$, the threshold is at majority. In this case, the clause gadget remains unchanged. If this is not the case, i.e., $\deg y_j = 3$, we attach exactly one cherry to y_j (and increase the threshold of y_j by 1). The modified clause gadget is depicted in Figure 2.9.

The placement of the variable and clause gadgets and connections between them remains the same as in proof of Theorem 2.18. Let β denote the number of clauses that contain exactly 3 literals (i.e., the number of cherries attached to clause gadgets). Total number of attached cherries is $\alpha = n + \beta$. We set $k = n + \alpha = 2n + \beta$. Let G denote the constructed graph and t the threshold function. It is not hard to see that G is still planar, $t(v) = \left\lceil \frac{\deg_G(v)}{2} \right\rceil$ for all $v \in V(G)$ and $\Delta G \leq 4$, as promised.

▶ **Claim 2.33.** *The formula φ is satisfiable if and only if (G, t, k) is a yes-instance of Target Set Selection.*

Proof. This can be shown by combining Claims 2.20 and 2.21 and Claim 2.30. More precisely, we already know by Claims 2.20 and 2.21 that the formula φ is satisfiable if and only if the originally constructed instance of Target Set Selection in the proof of Theorem 2.18 was a yes-instance. Let $(G_{\text{old}}, t_{\text{old}}, k_{\text{old}})$ denote the constructed instance from the proof of Theorem 2.18. Now, since the reduction from $(G_{\text{old}}, t_{\text{old}}, k_{\text{old}})$ to (G, t, k) is essentially the same as in proof of Theorem 2.29, we get by Claim 2.30 that (G, t, k) is a yes-instance if and only if $(G_{\text{old}}, t_{\text{old}}, k_{\text{old}})$ is a yes-instance. The claim follows by combining these two equivallences. ■

Notice that this reduction is a composition of two polynomial reductions, hence it is also a polynomial reduction. ■

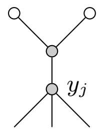


■ **Figure 2.8** Schematic representation of the variable gadget X_i for a variable x_i in the case of majority thresholds. The gray vertices have threshold 2, while the white vertices have threshold 1 (cf. Figure 2.4)

.

It is now straightforward to prove the hardness for the majority setting in the remaining graph classes. That is, grid graphs and unit disk graphs. We employ the same idea as in proof of Theorem 2.27.

▶ **Corollary 2.34.** *Target Set Selection is NP-hard under the majority threshold setting even if the underlying graph is a grid graph.*



■ **Figure 2.9** Representation of the clause gadget Y_j for a clause C_j containing exactly 3 literals in the case of majority thresholds. Filled vertices have threshold 2, empty vertices have threshold 1. The three half edges illustratee the fact that the gadget is connected with the rest of the graph only via y_j.

Proof. Apply the same reduction as in the proof of Theorem 2.27, but start from a planar instance with majority thresholds and $\Delta G \leq 4$ which is NP-hard by Theorem 2.32. Observe that a vertex created by subdividing an edge has degree 2 and all other degrees are unchanged. Notice that since the thresholds of the vertices created by the subdivision is 1, the new threshold function is indeed at majority. ■

As the class of grid graphs is a subclass of unit disk graphs, we also obtain hardness under the majority setting in unit disk graphs.

▶ **Corollary 2.35.** *Target Set Selection is NP-hard under the majority threshold setting even if the underlying graph is a unit disk graph.*

Chapter 3
Bounded degree graphs

In the previous chapter, we established hardness of Target Set Selection in all the commonly studied restrictions of the threshold function – constant, unanimous, majority in the classes of unit disk, and disk contact (planar) graphs. Note that the proofs provided hardness not only for the concrete graph classes but also for general graphs with very small maximum degree. It is natural to explore the problem's complexity on graphs with even smaller degree. Theorem 2.18 shows that Target Set Selection is NP-hard even when the underlying graph G has maximum degree $\Delta G \leq 4$ and thresholds are at most 2. Theorem 2.2 shows that when $\Delta G \leq 3$ and thresholds are at most 3, the problem is also NP-hard. Here, we show that Target Set Selection is still NP-hard when $\Delta G \leq 3$ and thresholds are at most 2. This complements a result of Kynčl et al. [51] who showed that the problem is solvable in polynomial time if all thresholds are exactly 2 and $\Delta G \leq 3$. This result suggests that the scenario where thresholds are allowed to be at most c may be different from the scenario where they are all exactly c. We show that even for unit disk graphs, Target Set Selection is NP-hard even when all thresholds are set to c for infinitely many c. In particular, we prove the result for $c = 2, 3, 4$. We also extend the tractability result by observing how to compute an optimal target set in graphs with $\Delta G \leq 2$.

We start with a simple observation that we can always upper-bound the threshold of a vertex by its degree.

▶ **Lemma 3.1.** *Let (G, t, k) be an instance of Target Set Selection. Then there is an equivalent instance (G', t', k') with $t'(v) \leq \deg_{G'}(v)$ for all $v \in V(G')$.*

Proof. If v is a vertex with threshold $t(v) > \deg v$, then it must be included in any target set. We thus set $G' = G - v$, decrease the threshold value of all neighbors of v by 1 (if not already at zero) and $k' = k - 1$. Certainly the new instance is equivalent to (G, t, k). Repeat this step until there are no vertices with threshold $t(v) > \deg v$. ■

▶ **Theorem 3.2.** *Target Set Selection is NP-hard even when the underlying graph has maximum degree $\Delta G \leq 3$ and thresholds are at most 2.*

Proof. We utilize the reduction of Kynčl et al. [51] used to show the NP-hardness of Irreversible 2-Conversion Set in graphs with maximum degree 4. Their problem exactly corresponds to Target Set Selection with thresholds set to 2. In their reduction, they make use of leafs with threshold 2 to virtually decrease the thresholds of some vertices in the resulting graph. In their problem, they are not explicitly allowed to have other thresholds than 2. By Lemma 3.1, we can erase all these leaf vertices with threshold 2 and decrease the threshold of their neighbors to obtain an equivalent instance (G, t, k). Observe that in their reduction, after erasing all these leaf vertices, we end up with a graph with maximum degree 3, i.e., we have $\Delta G \leq 3$. Also, the thresholds are at most 2, as promised. The theorem follows. ■

This result suggests that it might be of interest to distinguish between constant thresholds (i.e., $t(v) \leq c$ for some fixed constant c) and *exact* thresholds (i.e. $t(v) = c$ for some fixed constant c)(i.e., $t(v) = c$ for some c). Exact thresholds correspond exactly to the Irreversible c-Conversion Set problem. This problem is known to be NP-hard for all $c \geq 3$ [29] and also for $c = 2$ [17, 51]. Note that when $c = 2$ and the graph is 3-regular, the problem is equivalent to the Feedback Vertex Set problem (see [66, Lemma 2]), which is solvable in polynomial time in graphs with maximum degree 3 [67]. Kynčl et al. [51] extended the tractability result for Target Set Selection with $t(v) = 2$ (or, in their paper, Irreversible 2-Conversion Set) from 3-regular graphs to graphs with maximum degree 3. Going back to the class of unit disk graphs discussed in Chapter 2, we also have slightly weaker result for this class when the thresholds are exact. We rely on the result about NP-hardness of Independent Set on regular graphs (see Theorem 2.8 and corollary 2.15).

▶ **Theorem 3.3.** *For infinitely many constants c Target Set Selection is NP-hard when restricted to the class of unit disk graphs and the thresholds are exactly c. In particular, the claim holds for $c = 2, 3, 4$.*

Proof. For $c \geq 3$ we reduce from Independent Set restricted to instances where the underlying graph is c-regular unit disk graph where $c > 0$ and $c \equiv -1 \mod 4$ or $c \equiv -1 \mod 5$. Hardness of this setting is implied by Corollary 2.15. We proceed in a similar way as in proof of Theorem 2.17 but start with a c-regular graph for appropriate c.

For $c = 2$ we reduce from Target Set Selection with majority thresholds on grid graphs. Hardness of this setting is implied by Corollary 2.34. Let (G, t, k) be such instance and let $V(G) = \{v_1, \ldots, v_n\}$. We create a new instance (G', t', k') as follows. We are aiming for $t'(v) = 2$ for all $v \in V(G')$.

First, we obtain a disk representation $\mathcal{D} = \{D_1, \ldots, D_n\}$ for G as in proof of Observation 1.11. The disk D_i corresponds to the vertex v_i. Now, we fix vertices v_i with threshold 1 by attaching a leaf vertex v_i' with threshold 2 to v_i and we increase threshold of v_i by 1. In this way we have $t'(v_i) = t'(v_i') = 2$. Let z denote the number of vertices $v_i \in V(G)$ with $t(v_i) = 1$. We set $k' = k + z$. Let G' be the newly created graph.

▶ **Claim.** *The instances (G, t, k) and (G', t', k') are equivalent.*

Proof. Observe that by applying Lemma 3.1 to the instance (G', t', k') we obtain precisely the instance (G, t, k). The claim follows. ■

It remains to say how to realize the attachment of a leaf vertex in the unit disk representation. Let $s_i \in \mathbb{R}^2$ be the center of D_i and let $\varepsilon = \frac{1}{5}$. Observe that the representation satisfies: Every two disks have at most 1 point in common. Let v_i satisfy $t(v_i) = 1$. Thus, $\deg_G v_i \in \{1, 2\}$ because t is majority. As all the disks are embedded in an integer grid and $\deg v_i \leq 2$, there exists a direction $d_i \in \{(0,1), (1,0), (-1,0), (0,-1)\}$ such that $s_i + d_i$ is not a center of any other disk D_j. We add a new disk D_i' with diameter 1 centered at $s_i + \varepsilon d_i$ (see Figure 3.1).

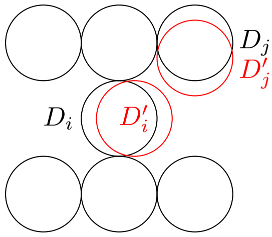
It is not hard to see that $D_i' \cap D_i \neq \emptyset$ and that D_i' does not intersect any other disks. In other words, this exactly corresponds to attaching a leaf vertex v_i' to v_i. We repeat this step for all other vertices $v \in V(G)$ satisfying $t(v) = 1$. Observe that the selection[1] $\varepsilon = \frac{1}{5}$ ensures that no matter which direction d_j we choose for any other disk D_j, the newly created disk D_i' intersects only the disk D_i. This can be checked by applying the triangle inequality several times, or directly, by computing the distances of the centers of corresponding disks.

To conclude the proof, note that we added at most 1 new vertex per each original vertex, thus the reduction is indeed polynomial. ■

▶ **Remark.** We remark that in the case for $c = 2$, the resulting unit disk graph still satisfies $\Delta G \leq 4$. We thus have a hardness result not only for $t(v) \leq 2$ and $\Delta G \leq 4$, but also for $t(v) = 2$ and $\Delta G \leq 4$ in the class of unit disk graphs. Notice that a very same proof also works for planar graphs but without all the geometrical mess.

▶ **Corollary 3.4.** *Target Set Selection is NP-hard even when the underlying graph is planar or unit disk graph with $\Delta G \leq 4$ and the thresholds are exactly 2.*

[1] The only problem might arise in the scenario shown in Figure 3.1. In this case $s_i + p_i = s_j + p_j$ and the two red disks might overlap if ε was choosen too large. In fact, one can compute that any $\varepsilon \in \left(0, 1 - \frac{1}{\sqrt{2}}\right)$ would suffice.



■ **Figure 3.1** Attaching leafs to vertices v_i, v_j corresponding to disks D_i, D_j. The black disks correspond to the original graph while the two red disks are the newly created leafs. The corresponding directions are $d_i = (1, 0)$ and $d_j = (0, -1)$.

▶ **Remark.** We remark that the first constant for which we don't have an NP-hardness result for Target Set Selection in the class of unit disk graphs and thresholds set exactly to c, is $c = 5$. It is implied by the fact that the proof heavily relies on the hardness of Independent Set, for which the situation is pretty much the same. Refer back to Remark 2.16 for more details.

We end the chapter with an observation on how to generalize the computation of an optimal target set for general graphs with $\Delta G \leq 2$. Such graphs are disjoint unions of cycles and paths. We further assume that the graph is connected since an optimal target set can be computed for each connected component. In other words, it is enough to show how to compute an optimal target set when the underlying graph is a path or a cycle.

Observe that if the graph contains a vertex with threshold 1 that induces a path on three vertices together with its neighbors, we can always bypass it – erase it and connect his two neighbors by an edge and the size of an optimal target set does not change. This can be seen as a reverse operation to subdivision. We formalize this in Observation 3.5.

▶ **Observation 3.5.** *Let $G = (V, E)$ be a graph and $t\colon V \to \mathbb{N}$ a threshold function and let $v \in V$ satisfy $\deg v = 2$ and $t(v) = 1$. Denote $N(v) = \{u, w\}$ and also assume that $\{u, w\} \notin E$. Let G' be a graph created from G by deleting v and adding edge $\{u, w\}$. Let t' be the function t restricted to $V(G')$. Then the following holds:*

i) If S is a target set for G with respect to t, then there is a target set S' for G' with respect to t' and $|S| = |S'|$.

ii) If S' is a target set for G' with respect to t', then S' is also a target set for G with respect to t.

Since we have $\Delta G \leq 2$, we can also assume that $t(v) \leq 2$ by Lemma 3.1.

▶ **Theorem 3.6.** *An optimal target set for the cycle C_n is of size $\max\left\{1, \left\lceil \frac{q}{2} \right\rceil\right\}$, where q is the number of vertices with threshold 2.*

Proof. Apply Observation 3.5 to all vertices of threshold 1 until we either end up with a triangle or with a cycle where all vertices have thresholds 2. If we end up with a tringle, then there are two cases to consider. If all vertices have threshold 2, then the optimal solution is of size $2 = \left\lceil \frac{3}{2} \right\rceil$, as claimed. If this is not the case picking either one of the vertex with threshold 2 or any vertex (if all vertices have threshold 1) suffices (and is also necessary).

If we end up with a cycle of length $\ell \geq 4$ with all thresholds 2, then this corresponds to the Irreversible 2-Conversion Set and the optimal solution is of size $\left\lceil \frac{\ell}{2} \right\rceil = \left\lceil \frac{q}{2} \right\rceil$ as noted in [51].

We now use part ii) of Observation 3.5 and conclude that the computed target set is also optimal for the original cycle we started with. ■

▶ **Theorem 3.7.** *An optimal target set for path P_n is of size $\max\left\{1, \left\lceil \frac{q+1}{2} \right\rceil\right\}$, where q is the number of vertices with threshold 2.*

Proof. Apply Observation 3.5. Then erase leafs with threshold 1 as there is always an optimal solution not containing them and they will become active if their neighbor becomes active. After this process, we either end up with a single vertex with threshold 1 – this happens in the case where all thresholds in the original graph were equal to 1, so the solution is of size 1. Otherwise we end up with a path P_ℓ where all vertices have thresholds 2. This again corresponds to the Irreversible 2-Conversion Set and the optimal solution is of size $\lceil \frac{\ell+1}{2} \rceil = \lceil \frac{q+1}{2} \rceil$ as noted in [51].

In the same way as in proof of Theorem 3.6, we use part ii) of Observation 3.5 and conclude that the computed target set is also optimal for the original path we started with. ■

Chapter 4

Conclusion

We have shown NP-hardness of TARGET SET SELECTION in the class of unit disk graphs in all the three commonly studied settings – constant, unanimous and majority. En route, we also showed NP-hardness results for the classes of grid graphs and planar graphs. There was one exception – in the unanimous threshold setting on grid graphs, TARGET SET SELECTION is solvable in polynomial time.

In the last chapter, we demonstrated that there is a difference between exact and constant thresholds. That is, $t(v) = c$ for fixed c and $t(v) \leq c$ for fixed c, respectively. More precisely, we showed that if $\Delta G \leq 3$ and $t(v) \leq 2$, TARGET SET SELECTION is hard in general graphs, whereas it is solvable in polynomial time if $t(v) = 2$ and $\Delta G \leq 3$ by a previous result of Kynčl et al. [51].

We also returned back to the class of unit disk graphs and addressed the exact threshold setting and showed that for infinitely many constants c TARGET SET SELECTION remains NP-hard even in the classes of unit disk and planar graphs with $t(v) = c$. In particular, we showed this for $c = 2, 3, 4$. For the case $c = 2$, we still preserved the tight upper bound on maximum degree of the graph, i.e., $\Delta G \leq 4$.

We completed the complexity picture regarding maximum degree and observed how to compute optimal target sets in graphs with $\Delta G \leq 2$, i.e., cycles and paths.

Future directions and open questions

We give a few open questions that might be interesting to explore.

► **Question 4.1.** Is TARGET SET SELECTION NP-hard when restricted to the classes of unit disk or planar graphs even if the maximum degree is at most 3 and the thresholds are at most 2?

► **Question 4.2.** Is TARGET SET SELECTION NP-hard when restricted to the class of grid graphs and the threshols are exactly 2?

As TARGET SET SELECTION turned out to be still NP-hard even in the class of unit disk graphs and grid graphs, it might be a reasonable direction to step back and take a different path in the class hierarchy (see Figure 4.1). Instead of restricting disk graphs to unit disk graphs and grids, one could restrict the disk graphs to different subclasses. A subclass of the class of disk graphs that already provided a tractability result is the class of *interval graphs.* In this class, the constant threshold setting is solvable in polynomial time [10], and unanimous threshold setting even in linear time, as it is equivalent to the VERTEX COVER problem [56].

A natural question is: What is the complexity of TARGET SET SELECTION in the class of interval graphs under the majority threshold setting?

Note that while the algorithm given by Bessy et al. [10] for thresholds at most c runs in polynomial time, the degree of the polynomial depends on c. This implies that their algorithm cannot be extended to a polynomial-time algorithm for the majority threshold setting.

► **Question 4.3.** Is TARGET SET SELECTION solvable in polynomial time in the majority threshold setting when the underlying graph is an interval graph?

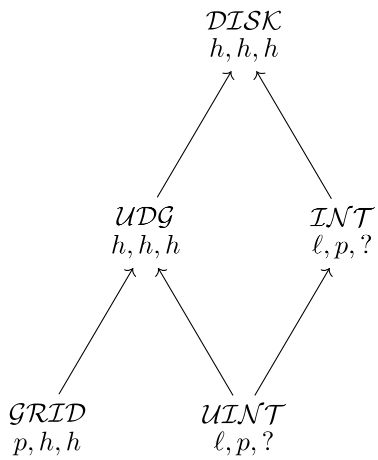
If the answer to the latter question is *yes*, we can even ask for an efficient algorithm for the degree-dependent threshold setting, which generalizes the exact, the unanimous, and the majority settings.

▶ **Question 4.4.** Is Target Set Selection solvable in polynomial time when the underlying graph is an interval graph and the threshold function satisfies $t(v) = f(\deg v)$ for some function f?

To the best of our knowledge, no NP-hardness result for Target Set Selection is known for the class of interval graphs, thus we may even ask:

▶ **Question 4.5.** Is Target Set Selection solvable in polynomial time when the underlying graph is an interval graph and the threshold function is unrestricted?

If an NP-hardness result occurs for the class of interval graphs, one could restrict the structure even further, e.g., to the class of *unit interval* graphs, where all the intervals in the representation have equal length. The class of unit interval graphs is also a subclass of unit disk graphs. Refer to Figure 4.1 for an overview of the mentioned graph classes and corresponding complexity results regarding Target Set Selection.



■ **Figure 4.1** Relationship between the considered graph classes in the further research of complexity of Target Set Selection. An arrow from a class $\mathcal{G}$ to a class $\mathcal{H}$ indicates that $\mathcal{G}$ is a subclass of $\mathcal{H}$. The three letters below each class correspond to the complexity of Target Set Selection in the unanimous, constant, and majority threshold setting, respectively. The letter h indicates NP-hardness of the given setting, p and ℓ indicate polynomial- and linear-time solvability, respectively. A question mark indicates an open question. Going left to right top to bottom, the graph classes are: the disk graphs, the unit disk graphs, the interval graphs, the grid graphs, the unit interval graphs.

Another question, unrelated to target sets, is about the hardness of the Independent Set problem on r-regular unit disk graphs for all constants r.

We explicitly showed the NP-hardness for the cases $r = 3$ and $r = 4$ in Theorem 2.8. We then noticed that this extends the hardness to all constants r satisfying $r = -1 \mod 4$ or $r = -1 \mod 5$. A natural question is, whether the problem NP-hard for *all* constants $r \geq 3$? We remark that the first r for which we don't have a proof of NP-hardness is $r = 5$. As noted in Remark 2.16, our approach from Section 2.2 is not applicable to give a proof for every constant r.

▶ **Question 4.6.** Is Independent Set NP-hard when restricted to the class of r-regular unit disk graphs for all constants $r \geq 3$?

Similar question follows with the hardness of exact thresholds in unit disk graphs given in Theorem 3.3. The proof relied heavily on the result about Independent Set. For Target Set Selection, we managed to include a different approach for $c = 2$, however, as before, the first c for which we don't have a proof of NP-hardness if $c = 5$.

▶ **Question 4.7.** Given any $c \geq 2$, is Target Set Selection NP-hard even when restricted to the class of unit disk graphs and all thresholds are exactly c?

References

[1] Sarah Spence Adams et al. “Modeling the spread of fault in majority-based network systems: Dynamic monopolies in triangular grids”. In: *Discrete Applied Mathematics* 160.10 (2012), pp. 1624–1633. DOI: `https://doi.org/10.1016/j.dam.2012.02.011`.

[2] Akanksha Agrawal and Maadapuzhi S. Ramanujan. “Distance from Triviality 2.0: Hybrid Parameterizations”. In: *Combinatorial Algorithms*. Ed. by Cristina Bazgan and Henning Fernau. Cham: Springer International Publishing, 2022, pp. 3–20.

[3] Saeed Akhoondian Amiri. “A note on the fine-grained complexity of MIS on regular graphs”. In: *Information Processing Letters* 170 (2021), p. 106123. DOI: `https://doi.org/10.1016/j.ipl.2021.106123`.

[4] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. DOI: `10.1017/CBO9780511804090`.

[5] Andrei Asinowski et al. “String graphs of k-bend paths on a grid”. In: *Electronic Notes in Discrete Mathematics* 37 (2011). LAGOS'11 – VI Latin-American Algorithms, Graphs and Optimization Symposium, pp. 141–146. DOI: `https://doi.org/10.1016/j.endm.2011.05.025`.

[6] Eytan Bakshy et al. “The Role of Social Networks in Information Diffusion”. In: *Proceedings of the 21st International Conference on World Wide Web*. WWW '12. Lyon, France: Association for Computing Machinery, 2012, pp. 519–528. DOI: `10.1145/2187836.2187907`.

[7] József Balogh, Béla Bollobás, and Robert Morris. “Bootstrap Percolation in High Dimensions”. In: *Combinatorics, Probability and Computing* 19.5-6 (2010), pp. 643–692.

[8] Suman Banerjee, Rogers Mathew, and Fahad Panolan. “Target Set Selection Parameterized by Vertex Cover and More”. In: *Theory of Computing Systems* 66 (Sept. 2022). DOI: `10.1007/s00224-022-10100-0`.

[9] Oren Ben-Zwi et al. “Treewidth governs the complexity of target set selection”. In: *Discrete Optimization* 8.1 (2011), pp. 87–96. DOI: `10.1016/j.disopt.2010.09.007`.

[10] Stéphane Bessy et al. “Dynamic monopolies for interval graphs with bounded thresholds”. In: *Discrete Applied Mathematics* 260 (2019), pp. 256–261. DOI: `10.1016/j.dam.2019.01.022`.

[11] John A. Bondy and Uppaluri S. R. Murty. *Graph Theory with Applications*. New York: Elsevier, 1976.

[12] Kellogg S. Booth and George S. Lueker. “Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms”. In: *Journal of Computer and System Sciences* 13.3 (1976), pp. 335–379. DOI: `https://doi.org/10.1016/S0022-0000(76)80045-1`.

[13] Heinz ”Breu and David G.” Kirkpatrick. “”On the complexity of recognizing intersection and touching graphs of disks””. In: *”Graph Drawing”*. Ed. by Franz J.” ”Brandenburg. ”Berlin, Heidelberg”: ”Springer Berlin Heidelberg”, ”1996”, ”88–98”.

[14] Heinz Breu and David G. Kirkpatrick. “Unit disk graph recognition is NP-hard”. In: *Computational Geometry* 9.1 (1998), pp. 3–24.

[15] Maw-Shang Chang. "Efficient Algorithms for the Domination Problems on Interval and Circular-Arc Graphs". In: *SIAM Journal on Computing* 27.6 (1998), pp. 1671–1694.

[16] Steven Chaplick et al. "Bend-Bounded Path Intersection Graphs: Sausages, Noodles, and Waffles on a Grill". In: *Graph-Theoretic Concepts in Computer Science*. Ed. by Martin Charles Golumbic et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 274–285.

[17] Ning Chen. "On the Approximability of Influence in Social Networks". In: *SIAM Journal on Discrete Mathematics* 23.3 (2009), pp. 1400–1415. DOI: 10.1137/08073617X.

[18] Chun-Ying Chiang et al. "Some results on the target set selection problem". In: *Journal of Combinatorial Optimization* 25.4 (May 2013), pp. 702–715. DOI: 10.1007/s10878-012-9518-3.

[19] Morgan Chopin et al. "Constant Thresholds Can Make Target Set Selection Tractable". In: *Theory of Computing Systems* 55.1 (2014), pp. 61–83. DOI: 10.1007/s00224-013-9499-3.

[20] Brent N. Clark, Charles J. Colbourn, and David S. Johnson. "Unit disk graphs". In: *Discrete Mathematics* 86.1 (1990), pp. 165–177. DOI: 10.1016/0012-365X(90)90358-O.

[21] Derek G. Corneil, H. Lerchs, and L. Stewart Burlingham. "Complement reducible graphs". In: *Discrete Applied Mathematics* 3.3 (1981), pp. 163–174. DOI: https://doi.org/10.1016/0166-218X(81)90013-5.

[22] Bruno Courcelle, Joost Engelfriet, and Grzegorz Rozenberg. "Handle-rewriting hypergraph grammars". In: *Journal of Computer and System Sciences* 46.2 (1993), pp. 218–270. DOI: 10.1016/0022-0000(93)90004-G.

[23] Bruno Courcelle, Johann Makowsky, and Udi Rotics. "Linear Time Solvable Optimization Problems on Graphs of Bounded Clique Width". In: vol. 33. Jan. 1970, pp. 287–291. DOI: 10.1007/10692760_1.

[24] Bruno Courcelle and Stephan Olariu. "Upper bounds to the clique width of graphs". In: *Discrete Applied Mathematics* 101.1 (2000), pp. 77–114. DOI: https://doi.org/10.1016/S0166-218X(99)00184-5.

[25] Marek Cygan et al. *Parameterized Algorithms*. Cham, 2015. DOI: 10.1007/978-3-319-21275-3.

[26] Elias Dahlhaus et al. "The Complexity of Multiterminal Cuts". In: *SIAM Journal on Computing* 23.4 (1994), pp. 864–894. DOI: 10.1137/S0097539792225297.

[27] Andreas Darmann and Janosch Döcker. "On simplified NP-complete variants of Not-All-Equal 3-Sat and 3-Sat". In: *CoRR* abs/1908.04198 (2019). arXiv: 1908.04198.

[28] Reinhard Diestel. *Graph Theory (Graduate Texts in Mathematics)*. Springer, 2005.

[29] Paul A. Dreyer and Fred S. Roberts. "Irreversible k-threshold processes: Graph-theoretical threshold models of the spread of disease and of opinion". In: *Discrete Applied Mathematics* 157.7 (2009), pp. 1615–1627. DOI: 10.1016/j.dam.2008.09.012.

[30] Pavel Dvořák, Dušan Knop, and Tomáš Toufar. "Target Set Selection in Dense Graph Classes". In: *SIAM Journal on Discrete Mathematics* 36.1 (2022), pp. 536–572. DOI: 10.1137/20M1337624.

[31] Martin E. Dyer and Alan M. Frieze. "Planar 3DM is NP-complete". In: *Journal of Algorithms* 7.2 (1986), pp. 174–184. DOI: https://doi.org/10.1016/0196-6774(86)90002-7.

[32] Herbert Fleischner, Gert Sabidussi, and Vladimir I. Sarvanov. "Maximum independent sets in 3- and 4-regular Hamiltonian graphs". In: *Discrete Mathematics* 310.20 (2010), pp. 2742–2749. DOI: 10.1016/j.disc.2010.05.028.

[33] Paola Flocchini et al. "Dynamic monopolies in tori". In: *Discrete Applied Mathematics* 137.2 (2004), pp. 197–212. DOI: https://doi.org/10.1016/S0166-218X(03)00261-0.

[34] Martin Golumbic and Bernard Ries. "On the Intersection Graphs of Orthogonal Line Segments in the Plane: Characterizations of Some Subclasses of Chordal Graphs". In: *Graphs and Combinatorics* 29 (Apr. 2013), pp. 499–517. DOI: 10.1007/s00373-012-1133-7.

[35] Jiong Guo, Falk Hüffner, and Rolf Niedermeier. "A Structural View on Parameterizing Problems: Distance from Triviality". In: *Parameterized and Exact Computation*. Ed. by Rod Downey, Michael Fellows, and Frank Dehne. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 162–173.

[36] Tim A. Hartmann. "Target Set Selection Parameterized by Clique-Width and Maximum Threshold". In: *Proceedings of the 44th International Conference on Current Trends in Theory and Practice of Computer Science, SOFSEM '18*. Vol. 10706. Lecture Notes in Computer Science. Cham: Springer, 2018, pp. 137–149.

[37] Petr Hliněný and Jan Kratochvíl. "Representing Graphs by Disks and Balls (a Survey of Recognition-Complexity Results)". In: *Discrete Mathematics* 229.1 (Feb. 2001), pp. 101–124. DOI: 10.1016/S0012-365X(00)00204-1.

[38] John Hopcroft and Robert Tarjan. "Efficient planarity testing". English. In: *J. Assoc. Comput. Mach.* 21 (1974), pp. 549–568. DOI: 10.1145/321850.321852.

[39] John E. Hopcroft and Richard M. Karp. "An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs". In: *SIAM Journal on Computing* 2.4 (1973), pp. 225–231. DOI: 10.1137/0202019. eprint: https://doi.org/10.1137/0202019.

[40] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. "Wormhole attacks in wireless networks". In: *IEEE Journal on Selected Areas in Communications* 24.2 (2006), pp. 370–380. DOI: 10.1109/JSAC.2005.861394.

[41] Mark L. Huson and Arunabha Sen. "Broadcast scheduling algorithms for radio networks". In: *Proceedings of the 1995 IEEE Military Communications Conference, MILCOM '95*. Vol. 2. 1995, pp. 647–651. DOI: 10.1109/MILCOM.1995.483546.

[42] Russel Impagliazzo and Ramamohan Paturi. "Complexity of k-SAT". In: *Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat.No.99CB36317)*. 1999, pp. 237–240. DOI: 10.1109/CCC.1999.766282.

[43] Ross J. Kang and Tobias Müller. "Sphere and Dot Product Representations of Graphs". In: *Discrete & Computational Geometry* 47.3 (Apr. 2012), pp. 548–568. DOI: 10.1007/s00454-012-9394-8.

[44] J. Mark Keil. "Finding Hamiltonian Circuits in Interval Graphs". In: *Information Processing Letters* 20 (1985), pp. 201–206.

[45] David Kempe, Jon Kleinberg, and Éva Tardos. "Influential Nodes in a Diffusion Model for Social Networks". In: *Proceedings of the 32nd International Colloquium on Automata, Languages and Programming, ICALP '05*. Vol. 3580. LNCS. Berlin, Heidelberg: Springer, 2005, pp. 1127–1138. DOI: 10.1007/11523468_91.

[46] David Kempe, Jon Kleinberg, and Éva Tardos. "Maximizing the Spread of Influence through a Social Network". In: *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '03*. Washington, D.C.: Association for Computing Machinery, 2003, pp. 137–146. DOI: 10.1145/956750.956769.

[47] David Kempe, Jon Kleinberg, and Éva Tardos. "Maximizing the Spread of Influence through a Social Network". In: *Theory of Computing* 11.4 (2015), pp. 105–147. DOI: 10.4086/toc.2015.v011a004.

[48] Paul Koebe. *Kontaktprobleme der konformen Abbildung*. Hirzel, 1936.

[49] Jan Kratochvíl. "Intersection graphs of noncrossing arc-connected sets in the plane". In: *Graph Drawing*. Ed. by Stephen North. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 257–270.

[50] Jan Kratochvíl and Jiří Matoušek. "Intersection Graphs of Segments". In: *Journal of Combinatorial Theory, Series B* 62 (1994), pp. 289–315.

[51] Jan Kynčl, Bernard Lidický, and Tomáš Vyskočil. "Irreversible 2-conversion set in graphs of bounded degree". In: *Discrete Mathematics & Theoretical Computer Science* 19.3 (Sept. 2017). DOI: 10.23638/DMTCS-19-3-5.

[52] Dénes König. "Über Graphen und ihre Anwendung auf Determinantentheorie und Mengenlehre". ger. In: *Mathematische Annalen* 77 (1916), pp. 453–465.

[53] Michael Lampis. "Algorithmic Meta-theorems for Restrictions of Treewidth". In: *Algorithmica* 64.1 (2012), pp. 19–37. DOI: 10.1007/s00453-011-9554-x.

[54] David Lichtenstein. “Planar Formulae and Their Uses”. In: *SIAM Journal on Computing* 11.2 (1982), pp. 329–343. DOI: 10.1137/0211025. eprint: https://doi.org/10.1137/0211025.

[55] Carsten Lund and Mihalis Yannakakis. “On the Hardness of Approximating Minimization Problems”. In: *Journal of the ACM* 41.5 (1994), pp. 960–981. DOI: 10.1145/185675.306789.

[56] Madhav V. Marathe, R. Ravi, and C. Pandu Rangan. “Generalized vertex covering in interval graphs”. In: *Discrete Applied Mathematics* 39.1 (1992), pp. 87–93. DOI: https://doi.org/10.1016/0166-218X(92)90116-R.

[57] Thiago Marcilon and Rudini Sampaio. “The maximum time of 2-neighbor bootstrap percolation: Complexity results”. In: *Theoretical Computer Science* 708 (2018), pp. 1–17. DOI: https://doi.org/10.1016/j.tcs.2017.10.014.

[58] Tomomi Matsui. “Approximation Algorithms for Maximum Independent Set Problems and Fractional Coloring Problems on Unit Disk Graphs”. In: *Revised Papers of the 2nd Japanese Conference on Discrete and Computational Geometry, JCDCG ’98.* Vol. 1763. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2000, pp. 194–200.

[59] George J. Minty. “On maximal independent sets of vertices in claw-free graphs”. In: *Journal of Combinatorial Theory, Series B* 28.3 (1980), pp. 284–304. DOI: https://doi.org/10.1016/0095-8956(80)90074-X.

[60] André Nichterlein et al. “On tractable cases of Target Set Selection”. In: *Social Network Analysis and Mining* 3.2 (2013), pp. 233–256. DOI: 10.1007/s13278-012-0067-7.

[61] David Peleg. “Majority Voting, Coalitions and Monopolies in Graphs”. In: *Proceedings of the Third International Colloquium on Structural Information & Communication Complexity, SIROCCO ’96.* 1996, pp. 152–169.

[62] Matthew Richardson and Pedro Domingos. “Mining Knowledge-Sharing Sites for Viral Marketing”. In: *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’02.* Edmonton, Alberta, Canada: Association for Computing Machinery, 2002, pp. 61–70.

[63] Marcel Salathé et al. “A high-resolution human contact network for infectious disease transmission”. In: *Proceedings of the National Academy of Sciences* 107.51 (2010), pp. 22020–22025. DOI: 10.1073/pnas.1009094108.

[64] Ramy Shaheen, Suhail Mahfud, and Ali Kassem. “Irreversible k-Threshold Conversion Number of Circulant Graphs”. In: *Journal of Applied Mathematics* 2022 (Aug. 2022), pp. 1–14. DOI: 10.1155/2022/1250951.

[65] Sarah Spence et al. “Irreversible k-threshold and majority conversion processes on complete multipartite graphs and graph products”. In: *Computing Research Repository - CORR* 56 (Feb. 2011).

[66] Asahi Takaoka and Shuichi Ueno. “A Note on Irreversible 2-Conversion Sets in Subcubic Graphs”. In: *IEICE Transactions on Information and Systems* E98.D (Aug. 2015), pp. 1589–1591. DOI: 10.1587/transinf.2015EDL8021.

[67] Shuichi Ueno, Yoji Kajitani, and Shin’ya Gotoh. “On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three”. In: *Discrete Mathematics* 72.1 (1988), pp. 355–360.

[68] Leslie G. Valiant. “Universality considerations in VLSI circuits”. In: *IEEE Transactions on Computers* 30.02 (Feb. 1981), pp. 135–140. DOI: 10.1109/TC.1981.6312176.

[69] David Zuckerman. “Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number”. In: *Proceedings of the Thirty-Eighth Annual ACM Symposium on Theory of Computing.* STOC ’06. Seattle, WA, USA: Association for Computing Machinery, 2006, pp. 681–690. DOI: 10.1145/1132516.1132612.