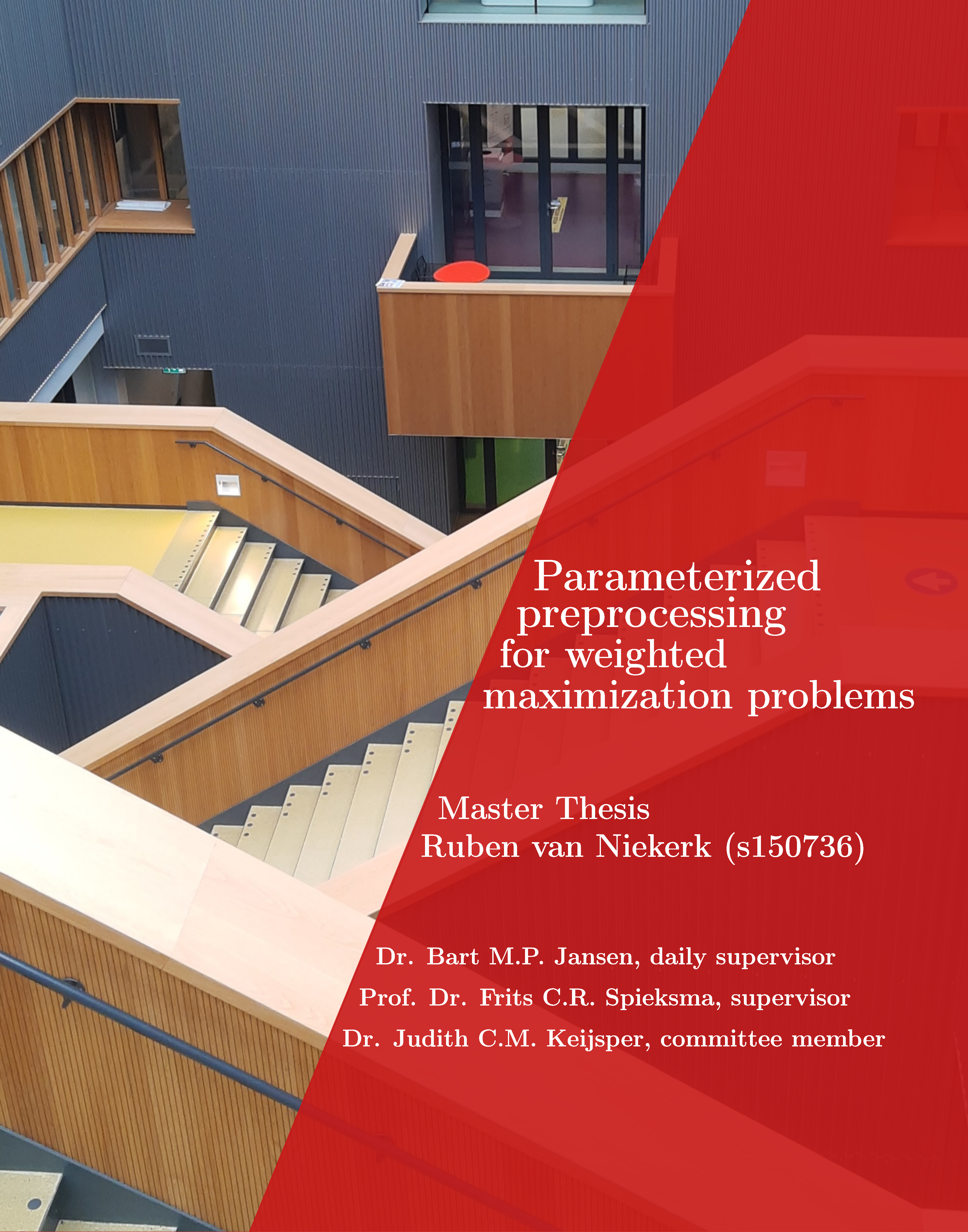
MASTER

Parameterized preprocessing for weighted maximization problems

van Niekerk, Ruben W.A.

Award date:
2020



Parameterized preprocessing for weighted maximization problems

Master Thesis

Ruben van Niekerk (s150736)

Dr. Bart M.P. Jansen, daily supervisor

Prof. Dr. Frits C.R. Spieksma, supervisor

Dr. Judith C.M. Keijsper, committee member

TU/e
EINDHOVEN UNIVERSITY OF TECHNOLOGY

/ Department of Mathematics and Computer Science

Abstract

Given a graph G, equipped with a weight function $\omega : V(G) \to \mathbb{N}$ and integers k and t, the NP-hard parameterized decision problem Maximum Weighted Independent Set corresponds to whether G contains an independent set of size exactly k and weight at least t. In other words, the task is to decide whether there is a vertex subset $X \subseteq V(G)$ of size k such that there are no edges between vertices of X and the total weight of vertices in X is at least t. On general graphs, this problem is known to be $W[1]$-hard. In this paper, we consider this problem for planar graphs. We introduce reduction rules to obtain an equivalent problem, in which the treewidth of the input graph is bounded. Then, we apply a dynamic programming algorithm to solve the problem on planar graphs in a running time only sub-exponential in complexity parameter k.

In the second part we present a kernel with $\mathcal{O}(k^3)$ vertices. Kernelization is a provably effective technique for efficient preprocessing the input for the algorithm and has a lot of applications. A kernel is obtained by deleting vertices that would never be part of a solution. This way an equivalent instance of the problem is obtained, whose graph has at most $\mathcal{O}(k^3)$ vertices.

Finally, we discuss generalizations of the techniques to the similar problems Maximum Weighted Induced Subgraph and Maximum Weighted Induced d-Scattered Subgraph.

Contents

1 Introduction

1.1 Background and motivation

An independent set, also known as a stable set, is a subset of the vertices of a graph, such that no two vertices share an edge. In Maximum Independent Set, the goal is to find an independent set of maximum cardinality. This is one of the most iconic NP-hard problems and it has both theoretical and practical applications [1]. Its equivalent counterpart, the clique problem, is one of the famous 21 NP-complete problems described by Karp in 1972 [2].
In this paper we will focus on a natural generalization, Maximum weighted Independent Set. Here the vertices have an integer weight and instead of the largest independent set, we are now interested in the heaviest independent set of a fixed size. Note that Maximum Independent Set is equivalent to MWIS if the vertices have identical weight. This generalization and variants have plenty of natural applications and have been studied thoroughly in the past decades [3, 4, 5].
The notion of fixed-parameter tractability (FPT) has been introduced to develop algorithms for NP-hard problems, which are efficient when a so-called *complexity parameter* is small [6]. It is for example not too difficult to check whether there exists an independent set of size 5 in a planar graph, regardless of the size of the graph. We will apply this framework to investigate the complexity of MWIS.

One of the main aspects of parameterized complexity is dealing with NP-hard problems, problems we believe cannot be solved efficiently. One way of obtaining FPT algorithms is by efficiently reducing a problem to an equivalent instance whose encoding length is bounded by a function of the complexity parameter k. We can for example delete the irrelevant parts of the graph. A brute-force algorithm can then be used on this so-called *kernel* to solve the problem. The resulting notion of *kernelization* can also be used as a model of provably effective preprocessing, which is of independent interest. Formal definitions are given in Section 2.
An important theorem states that a decidable problem is in the complexity class FPT, when it admits a kernel that is bounded by a function of k. The key insight for this theorem is that any vertex subset of the vertices in such a kernel can be brute forced as a solution in FPT time.
On planar graphs, Maximum Independent set admits a kernel with $4k$ vertices. A sketch of the proof will be provided. According to the 4-color theorem [7], a planar graph can be partitioned in exactly 4 independent sets and his partition can be found efficiently. Therefore, if the number of vertices in the graph is larger than $4k$, there is an independent set of size at least k, in which case the problem is solved. If the number of vertices is smaller, we have a small, equivalent instance, which is sufficient for a kernel.

Meta-kernelization is a framework that can be used as a general recipe to construct kernels for several problems. The main motivation for this paper is an article by Bodlaender et al, published in 2016 [8]. In this article, two meta-theorems regarding kernelization are provided. These theorems are applicable to unweighted problems. In fact many unweighted problems admit (linear) kernels, which can be found using a technique called *protrusion decompositions*. This technique involves decomposing the graph into a small separator and many independent subgraphs. However, this is not directly applicable to kernelization for weighted problems, since weighted problems do not have a property called *finite integer index* [9, Chapter 16]. This was the main inspiration for this paper.

1.2 Problem Statement

The focus point of this paper is the maximum weighted independent set problem on planar graphs. In this section the problem will be discussed and we will show some variants. The problem is defined as follows:

MAXIMUM WEIGHTED INDEPENDENT SET ON PLANAR GRAPHS	**Parameter:** k

Input: A planar graph G equipped with a weight function $\omega : V(G) \to \mathbb{N}$ and integers k, t
Question: Does there exist an independent vertex set $X \subseteq V(G)$ such that $|X| = k$ and $\omega(X) := \sum_{v \in X} w(v) \geq t$?

The used definitions are discussed in the preliminaries in Chapter 2. The reason my work is restricted to planar input graphs, is that the unweighted variant, MAXIMUM INDEPENDENT SET, is known to be $W[1]$-hard on general graphs, so it is not believed to admit an FPT algorithm [10]. On planar graphs, a kernel does exist for the unweighted version, as described in the previous subsection. The question whether a kernel exists for the weighted version will be discussed in this paper.

Variants of MWIS

There is an important variant, in which the goal is to find the (weight of the) maximum weighted independent set disregarding the size of the set. There is also a variant in which the goal is to find the maximum weighted independent set of size at most k. Note that our variant of MWIS is more general, since we can solve the '$\leq k$'-variant by solving the 'exact k'-variant while adding k isolated vertices of weight 0.

Another generalization is MAXIMUM WEIGHTED INDUCED CONNECTED SUBGRAPH (MWICS) on planar graphs. We use $k \cdot H$ to denote the disjoint union of k copies of the graph H. Using this notion, the problem is defined as follows:

MAXIMUM WEIGHTED INDUCED CONNECTED SUBGRAPH	**Parameter:** k

Input: A planar graph G equipped with a weight function $\omega : V(G) \to \mathbb{N}$, a (finite) connected graph H and integers k, t
Question: Does there exist an vertex set $X \subseteq V(G)$ of size $|V(H)| \cdot k$ such that $G[X]$ is isomorphic to $k \cdot H$ and $\omega(X) := \sum_{v \in X} w(v) \geq t$?

Specific variants of this problem are MAXIMUM WEIGHTED INDEPENDENT SET, MAXIMUM WEIGHTED INDUCED MATCHING and MAXIMUM WEIGHTED INDUCED TRIANGLE PACKING, where subgraph H will be a vertex, a path on 2 vertices and a clique of size 3 respectively.
The penultimate generalization we will discuss is MAXIMUM WEIGHTED D-SCATTERED SET (MWdSS), in which the distance of any 2 vertices in the vertex set must be at least d in the input graph.
Finally, there is the combined generalization of MAXIMUM WEIGHTED D-SCATTERED INDUCED SUBGRAPH, in which we are looking for the heaviest k subgraphs that have distance at least d from each other.

1.3 Our contribution

The most important theorems in this paper are the following:

Theorem 1.1. MAXIMUM WEIGHTED INDEPENDENT SET on planar graphs can be solved in in $\mathcal{O}^*(2^{33.75\sqrt{k}})$ time.

Theorem 1.2. MAXIMUM WEIGHTED INDEPENDENT SET on planar graphs admits a kernel with $\mathcal{O}(k^3)$ vertices.

Currently it is known that MWIS is $W[1]$-hard on general graphs and NP-hard on planar graphs. This implies that MWIS is either $W[1]$-hard or FPT on planar graphs. Theorem 1.1 implies that the problem is fixed parameter tractable. In Section 3, we will give an algorithm that solves MAXIMUM WEIGHTED INDEPENDENT SET on planar graphs in $\mathcal{O}^*(2^{33.75\sqrt{k}})$ time. The $\mathcal{O}^*$-notation is used to disregard terms that depend only polynomially on all parameters.

The main result of the paper is Theorem 1.2, that guarantees the existence of a polynomial kernel, which used to be an open problem. This result is more powerful than the result in Theorem 1.1, since the existence of a kernel directly implies that a problem is FPT. This theorem will be proven in Section 4.

1.4 Techniques

In this section we will describe the usage of different techniques related to kernelization. In Section 3.1 we will use a reduction rule to reduce the graph to an equivalent instance that has a distance-2-dominating set of size at most k. That is a vertex set, such that any vertex in the graph has distance at most 2 to that set. Then we will use the grid minor theorem for planar graphs to bound the treewidth of that graph. That theorem states that the treewidth of a planar graph is linearly bounded by the largest grid hidden inside it. Treewidth is a measure of the complexity of a graph, because many NP-hard problems can be solved in polynomial time on graphs whose treewidth is bounded by a constant. Finally we will solve the problem using a dynamic programming algorithm for graphs with a bounded treewidth.

In the next section, we will prove that the obtained graph with a distance-2-dominating set, has a so called protrusion decomposition. That technique involves decomposing the graph in a small separator and many independent subgraphs that have small treewidth. We can use this by constructing a treewidth modulator, a small vertex set, such that the remainder of the graph has a small treewidth. Then we will use the obtained protrusion decomposition to get a kernel with $\mathcal{O}(k^3)$ vertices.

This approach differs from the meta-kernelization paper, because they consider problems with a so called finite integer index and that excludes weighted problems. Finally, in Section 5, we will discuss how further applications of the same techniques might be used to obtain similar results for the generalizations MAXIMUM WEIGHTED D-SCATTERED SET and MAXIMUM WEIGHTED INDUCED SUBGRAPH.

1.5 Related work

For this paper, the book KERNELIZATION, THEORY OF PARAMETERIZED PREPROCESSING by Fomin et al. [9] is consulted. This book from 2019 contains the latest developments in the field of kernelization.

Another related book is called PARAMETERIZED ALGORITHMS by Marek Cygan et al. [11] from 2015. It contains a lot of useful information about parameterized algorithms, such as kernelization, as the title suggests. During my project, the article EFFICIENT WEIGHTED INDEPENDENT SET COMPUTATION OVER LARGE GRAPHS by Zheng et al. [12] was published. This article is about data reduction for MWIS on general graphs. To that extent, it is very similar to my paper, but it contains less powerful theoretical results and focuses more on proving effectiveness by experimental implementation on real graphs.

POLYNOMIAL-TIME DATA REDUCTION FOR DOMINATING SET by Jochen Alber et al. is another related article regarding linear kernels for DOMINATING SET on planar graphs. [13] This is an important graph theory problem that is $W[2]$-hard in general graphs, but also proves to be FPT in planar graphs.

The article about polynomial kernels for weighted problems [14] is especially interesting for us, since we are also looking for a polynomial kernel for a weighted problem. Later in Section 4.3 we will be using a weight adjustment algorithm described in this article.

Finally the book BIDIMENSIONALITY AND KERNELS by Fedor V. Fomin at al. [15] also contains a lot of information. It puts the focus on finding kernels for problems with finite integer index, similar to the meta-kernelization paper.

1.6 Organization

First of all, in Section 2, the preliminaries, we will present and explain the adopted definitions. In the following section we will describe an FPT-algorithm for MWIS. After that, in Section 4, we will construct a kernel with $\mathcal{O}(k^3)$ vertices. In Section 5 we will come up with ideas that might be used to generalize this result to other weighted maximization problems.

2 Preliminaries

Definition 2.1 (Parameterized (decision) problem)**.** A **parameterized problem** is a language $L \subseteq \Sigma^* \times \mathbb{N}$, where Σ is a fixed, finite alphabet. For an instance $(x, k) \in \Sigma^* \times \mathbb{N}$, k is called the (complexity) parameter [11, Chapter 1.1].

The instance (x, k) of a parameterized problem is called a yes-instance, if the corresponding question can be answered with 'yes', and otherwise it is called a no-instance.

Definition 2.2 (Fixed-parameter tractable)**.** A parameterized problem $L \subseteq \Sigma^* \times \mathbb{N}$ is called **fixed-parameter tractable** (FPT) if there exists an algorithm $\mathcal{A}$ (called a fixed-parameter algorithm), a computable function $f : \mathbb{N} \to \mathbb{N}$, and a constant c such that, given $(x, k) \in \Sigma^* \times \mathbb{N}$, the algorithm $\mathcal{A}$ correctly decides whether $(x, k) \in L$ in time bounded by $f(k) \cdot |(x, k)|^c$. The complexity class containing all fixed-parameter tractable problems is called FPT [11, Chapter 1.1].

Definition 2.3 (Kernel, Kernelization)**.** A **kernelization** algorithm, or simply a kernel, for a parameterized problem $\mathcal{Q}$ is an algorithm $\mathcal{A}$ that, given an instance (I, k) of $\mathcal{Q}$, works in polynomial time and returns an equivalent instance (I', k') of $\mathcal{Q}$. Moreover, we require that $|I'| + k' \leq g(k)$ for some computable function $g : \mathbb{N} \to \mathbb{N}$ [11, Chapter 2.1].

The function g is called the size of the kernelization, and we have a polynomial kernelization if $g(k)$ is polynomially bounded in k.

Definition 2.4 (Graph)**.** A **graph** is a structure that has a vertex set $V(G)$ and an edge set $E(G) \subseteq \binom{V(G)}{2}$. Those edges are said to connect the two vertices, and the vertices are called the endpoints of the edge.

Definition 2.5 (Simple Graph)**.** A **simple graph** is a graph that does not have loops or parallel edges, i.e. no edges with the same start and end point.

Definition 2.6 (Independent set)**.** An **independent set** $S \subseteq V(G)$ in a graph G is a subset of the vertices such that $G[S]$ has no edges.

Definition 2.7 (Distance)**.** The **distance** between two vertices $s, t \in V(G)$, denoted as $d_G(s, t)$, is defined as the number of edges on a shortest path between s and t in G.

Definition 2.8 (Neighborhood)**.** The **open neighborhood** of a vertex $v \in V(G)$ in a graph G, denoted as $N_G(v)$, is defined as $N_G(v) := \{u \in V(G) \mid \{u, v\} \in E(G)\}$. The **closed neighborhood** of a vertex $v \in V(G)$ in a graph G, denoted as $N_G[v]$, is defined as $N[v] := N(v) \cup \{v\}$.

Definition 2.9. (Planar graph) A graph is called **planar** if it has an embedding in the plane without edge crossings.

Definition 2.10. (k-outerplanar graph) An embedding of a graph is 1-outerplanar, if it is planar, and all vertices lie on the exterior face. For $k \geq 2$, an embedding of a graph G is k-outerplanar, if it is planar, and when all vertices on the outer face are deleted, then a $(k-1)$-outerplanar embedding of the resulting graph is obtained. A graph is k**-outerplanar**, if it has a k-outerplanar embedding. [16]

We will use this concept to bound the treewidth of a graph in Section 4.1. The graph in Figure 1 is 3-outerplanar, since we can remove the vertices incident to the outer face three times until the graph is empty.

Definition 2.11 (Tree decomposition)**.** A **tree decomposition** (T, χ) of G consists of a connected acyclic graph T and a function χ. The vertices of T are called *nodes*. The function $\chi : V(T) \to 2^{V(G)}$ assigns to each node t of $V(T)$ a vertex set $\chi(t) \subseteq V(G)$, a so-called *bag*, such that the following three properties hold.

1. $\bigcup_{t \in V(T)} \chi(t) = V(G)$, so each vertex is represented in at least one bag;

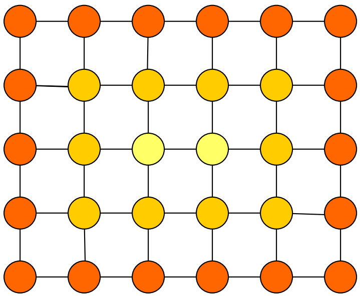


Figure 1: An example of a 3-outerplanar graph.

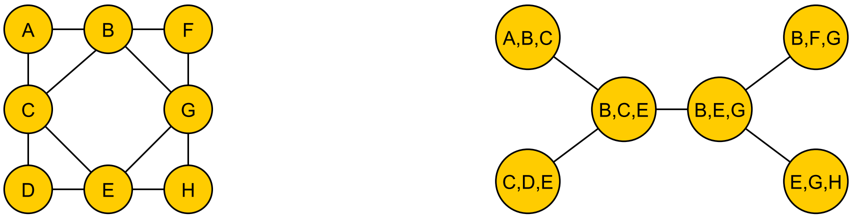


Figure 2: An example of a graph G and a tree decomposition of G.

2. If $v \in \chi(t)$ and $v \in \chi(t')$ for different nodes t and t' then v is in the bag of every node on the unique path in T between t and t'; and
3. For each edge $e \in E(G)$, there exists a node in $V(T)$ that contains both endpoints of e in its bag.

An example of a graph and a possible tree decomposition can be found in Figure 2. Note that all three properties are satisfied. The so-called *width* of a tree decomposition is 1 less than the maximum number of vertices in a bag. We will use the property that the vertices in a bag $\chi(t)$ are a separator in G for the connected components of the tree $T - \{t\}$. In Figure 2, we can conclude from the tree decomposition that the vertex set $\{B, C, E\}$ separates A, C and the other vertices in G. We will introduce tree decompositions in Section 3.2. Tree decompositions are very useful and we will use them to define the parameter *treewidth*.

Definition 2.12 (Treewidth)**.** The **treewidth** of a graph G is 1 less than the minimum over all valid tree decompositions of G of the maximum number of vertices contained in one bag.

The treewidth of the graph in Figure 2 is 2, since there exists a valid tree decomposition with at most 3 vertices in each bag, but not with at most 2 vertices in each bag.

Definition 2.13 (Treewidth-η-modulator)**.** A vertex set $X \subseteq V(G)$ is called a **treewidth-η-modulator** of G if $G - X$ has treewidth at most η [9, Chapter 15.2].

Definition 2.14 (Nice tree decomposition)**.** A tree decomposition (T, χ) of G is called **nice** if it satisfies the following 5 properties:

1. T is a rooted tree with root r;
2. $\chi(r) = \emptyset$ and for each node ℓ of T without children, we have $|\chi(\ell)| = 1$[1];
3. every node has at most 2 children;

[1] In [9, Chapter 14.3], Fomin et al. take $\chi(\ell) = \emptyset$, but for this paper, our similar definition is more useful.

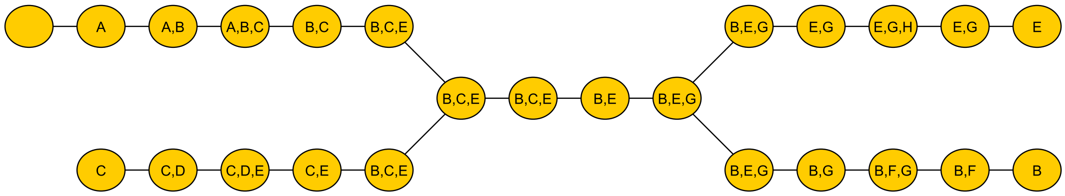


Figure 3: A nice tree decomposition of the graph from Figure 2.

4. if b has two children, say b_1 and b_2, then $\chi(b) = \chi(b_1) = \chi(b_2)$; and
5. if b has one child, say b_1, then $|\chi(b) \triangle \chi(b_1)| \leq 1$.

We call bags of nodes with no children *leaf bags*. We call bags of nodes with 2 children *join bags*. We call bags of nodes with one child *introduce bags* if they have more vertices than their child. We call them *forget bags* if they have less vertices than their child. In nice tree decompositions, we use the notation T_a to denote the subtree of T rooted at node a, and we use $\chi(T_a)$ to refer to the vertices in the bags of T_a. Finally, we use χ_a to be the restriction of χ to T_a. Then for each node a, we have that (T_a, χ_a) is a tree decomposition of $G[\chi(T_a)]$ [9, Chapter 14.3].

In this paper, nice tree decompositions are used for dynamic programming algorithms for graph with a bounded treewidth, since they have very useful properties and can be transformed from arbitrary tree decompositions efficiently. An example of a nice tree decomposition of the graph in Figure 2 can be found in Figure 3. The node with the empty bag can be chosen as the root of T. Note that all properties of a tree decomposition and a nice tree decomposition are satisfied.

Definition 2.15 (Semi-nice tree decomposition)**.** A **semi-nice tree decomposition**, is a nice tree decomposition without the requirement that the rootbag is empty.

Definition 2.16 (Protrusion decomposition)**.** [9, Chapter 15.4] For integers α, β and t, an (α, β, t)**-protrusion decomposition** is a tree-decomposition (T, χ) of G, such that the following conditions are satisfied:

- T is a rooted tree with root r and $|\chi(r)| \leq \alpha$;
- r has degree at most β in T; and
- For every node $v \in V(T)$ except r, we have $|\chi(v)| \leq t$.

Definition 2.17 (Nice Protrusion decomposition)**.** A protrusion decomposition (T, χ) of G is a **nice protrusion decomposition** if for every non root node a of T, (T_a, χ_a) is a semi-nice tree decomposition of $G[\chi(T_a)]$ [9, Chapter 15.4].

Protrusion decompositions are very useful when the graph has a small separator. That is when only a few vertices keep the graph together. An example of a graph and corresponding nice protrusion decomposition can be found in Figure 4. Here, the stars represent the other semi-nice tree decompositions. Note that this is a $(5, 4, 4)$-protrusion decomposition.

Definition 2.18 (Graph minor)**.** An undirected graph H is called a **minor** of the simple graph G if H can be formed from G by deleting edges and vertices and by contracting edges. The contraction operations replaces an edge and its two endpoints by a new vertex, connected to all neighbors of the replaced vertices. Then it deletes loops and parallel edges to keep H simple.

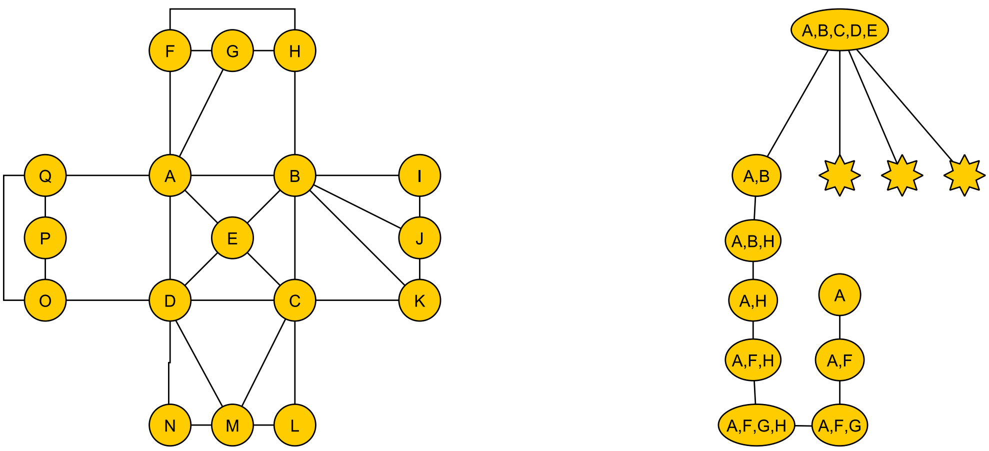


Figure 4: A graph G with a small separator and a $(5, 4, 4)$-protrusion decomposition of G.

3 An FPT algorithm for MWIS

In this section, we first present a reduction rule for MWIS. This reduction rule compresses any instance to an equivalent instance that contains a distance-2-dominating set. Then, we bound the treewidth of the graph of this equivalent instance. Finally, we will use this to solve the problem in $\mathcal{O}^*(2^{33.75\sqrt{k}})$ time, using a dynamic programming algorithm.

3.1 A reduction rule for MWIS

First, recall the definition of MWIS.

> Maximum Weighted Independent Set on planar graphs **Parameter:** k
> **Input:** A planar graph G equipped with a weight function $\omega : V(G) \to \mathbb{N}$ and integers k, t
> **Question:** Does there exist an independent vertex set $X \subseteq V(G)$ such that $|X|= k$ and $\omega(X) := \sum_{v \in X} w(v) \geq t$?

We will need the following lemma:

Lemma 3.1. Let G be a graph and let $(s_1, \ldots, s_k)$ be a sequence of vertices of G such that $N_G[s_i] \cap N_G[s_j] = \emptyset$ for all $1 \leq i < j \leq k$. For each $1 \leq i \leq k$, define $G_i := G - \bigcup_{1 \leq j \leq i} \{v \in V(G) : d_G(v, s_j) \leq 2\}$ and define $G_0 := G$. If for all $i \in [k]$ we have $\omega(s_i) = \max_{v \in V(G_{i-1})}\{\omega(v)\}$, then (G, k, t) is a yes-instance if and only if $(G - G_k, k, t)$ is a yes-instance. Furthermore, $(G - G_k, k, t)$ can be constructed in linear time.

Proof. Define $G' := G - G_k$ and let vertex set $S := \{s_i | i \in [k]\}$ be given. Clearly, (G', k, t) being a yes-instance directly implies that (G, k, t) is a yes-instance, because the same vertex set used as the solution in G' can be used as the solution in G.

For the other direction, let W be a solution in G that minimizes $|W \setminus V(G')|$. If $|W \setminus V(G')|= 0$ then $W \subseteq V(G')$ and W also forms a solution in G'. Therefore (G', k, t) is a yes-instance.
Suppose that $|W \setminus V(G')|> 0$. First we will show that that there exists a vertex $s \in S \setminus W$, such that $N_G[s] \cap W = \emptyset$. This holds, since both W and S have size k and there is an element in $W \setminus V(G') \subseteq W \setminus S$. Therefore, the only way there is no vertex in S for which this property holds, is when all k vertices in S are neighbors of the at most $k-1$ vertices in $W \setminus \{v\}$. In that case, the pigeon hole principle gives us that there is at least one vertex $u \in W$ that is neighboring at least 2 vertices of S in G. However, this is not possible, since $N_G[s_i] \cap N_G[s_j] = \emptyset$ for all $1 \leq i < j \leq k$.
Now, let v be a vertex in $W \setminus V(G')$. We will show that $W \setminus \{v\} \cup \{s_i\}$ is also a valid solution in G, where $s_i \in S \setminus W$ such that $N_G[s_i] \cap W = \emptyset$. We have already shown that that $W \setminus \{v\} \cup \{s_i\}$ is also an independent set of size k and now we will show that it has at least the weight of W. For this, we need to show that s_i has at least the weight of v. This holds, because if v has a larger weight than s_i, we do not satisfy $\omega(s_i) = \max_{v \in V(G_{i-1})}\{\omega(v)\}$. This means that $W \setminus \{v\} \cup \{s_i\}$ is also an independent set of size k of at least the same weight as W in G.
This new solution W' uses strictly more vertices of G'. This contradicts the assumption that W is a solution in G that minimizes $|W \setminus V(G')|$. Therefore we can conclude that if there exists a solution in $V(G)$, there also exists a solution in $V(G')$. This proves the statement is proven. All steps necessary for the construction of G_k take linear time, so the new graph $(G - G_k, k, t)$ can be found in linear time.

■

Before the reduction rule is stated, we introduce the notion of a distance-2-dominating set of a graph. Intuitively, this is a vertex subset for which any vertex in the graph is at distance at most 2 to any vertex in the subset.

Definition 3.2. Distance-2-dominating set.
Let G be a graph and let $D \subseteq V(G)$. Then D is called a **distance-2-dominating set** if for each $v \in V(G)$ we have $d(u, v) \leq 2$ for some $u \in D$.

Now, we have the appropriate tools to present the reduction rule and prove it.

Reduction rule 3.3. First we define $G_0 := G$. Then we iteratively choose s_i to be a vertex of highest weight in G_{i-1} and define $G_i := G_{i-1} - \{v \in V(G_i) : d_{G_{i-1}}(v, s_i) \leq 2)$. In other words, we keep removing a heaviest vertex and its neighbors and their neighbors. If $V(G_i) = \emptyset$ for any $i \leq k$, we define $V(G_k) := \emptyset$ instead. Then, the new instance is $(G - G_k, k, t)$.

Proof. First, note that constructed sequence of vertices $(s_1, \dots, s_k)$ satisfies $N[s_i] \cap N[s_j] = \emptyset$ for all $1 \leq i < j \leq k$ and $\omega(s_i) = \max_{v \in V(G_{i-1})}\{\omega(v)\}$. In other words, this sequence satisfies the precondition of Lemma 3.1. Now this lemma gives us that the instance $(G - G_k, k, t)$ is equivalent and can be obtained in polynomial time. This lemma guarantees the safeness of the reduction rule. ■

We will present the result of this reduction rule in the following lemma.

Lemma 3.4. Let (G, k, t) be an instance of MWIS. Then in polynomial time, we can construct an equivalent instance (G', k, t) that has a distance-2 dominating set of size at most k.

Proof. The application of Reduction rule 3.3, yields an equivalent instance $(G - G_k, k, t)$ in polynomial time. This $G - G_k$ is a graph that contains a 2-dominating-set of size k, since it contains exactly all vertices at distance at most 2 to the vertices in $S = \{s_1, \dots, s_k\}$. Note that if G_i has no vertices for some $i \leq k$, the reduction rule does not alter the graph, but it does guarantee the existence of a 2-dominating-set of size at most k. ■

3.2 A Bound on the treewidth

We will first use the notion of *branch sets*, as defined in [11, Chapter 6.3]. Intuitively, for any minor H of G, a branch set V_h contains the all vertices of G that will form a vertex $h \in V(H)$. All edges between the vertices in the same branch set are contracted or deleted in H.

Definition 3.5 (Minor model)**.** Let H be a minor of G. Then for every $h \in V(H)$ we can assign a nonempty so-called *branch set* $V_h \subseteq V(G)$, such that

(i) $G[V_h]$ is connected;

(ii) for different $g, h \in V(H)$, the branch sets V_g and V_h are disjoint; and

(iii) for every $gh \in E(H)$ there exists an edge $v_g v_h \in E(G)$ such that $v_g \in V_g$ and $v_h \in V_h$.

Such a family $(V_h)_{h \in V(H)}$ of branch sets is called a **minor model** of H in G.

Definition 3.6 (Grid minor)**.** A minor H of a graph G is called a **grid minor** of size $\ell \times \ell$ if it has branch sets $V_{h_{i,j}}$ such that $V(H) = h_{i,j}$ with $i, j \in [\ell]$ and $E(H) =: \{\{h_{i,j}, h_{i',j'}\} : |i-i'|+|j-j'|= 1\}$. ℓ is also called the *side length* of H.

We will use these definitions in the following lemma, that we will use to bound the treewidth.

Lemma 3.7. Let G be a graph with a fixed planar embedding. Let H be a largest grid minor of G with branch sets $V_{h_{i,j}}$. Then for any vertex $v \in V(G)$ the collection $\{V_{h_{i,j}} \mid V_{h_{i,j}} \cap d^2_G(v) \neq \emptyset, i, j \in \{3, \dots, \ell - 2\}\}$ has size at most 25, in which $d^2_G(v)$ is the set of vertices that have distance at most 2 from v in G. ℓ denotes the side length of grid minor H.

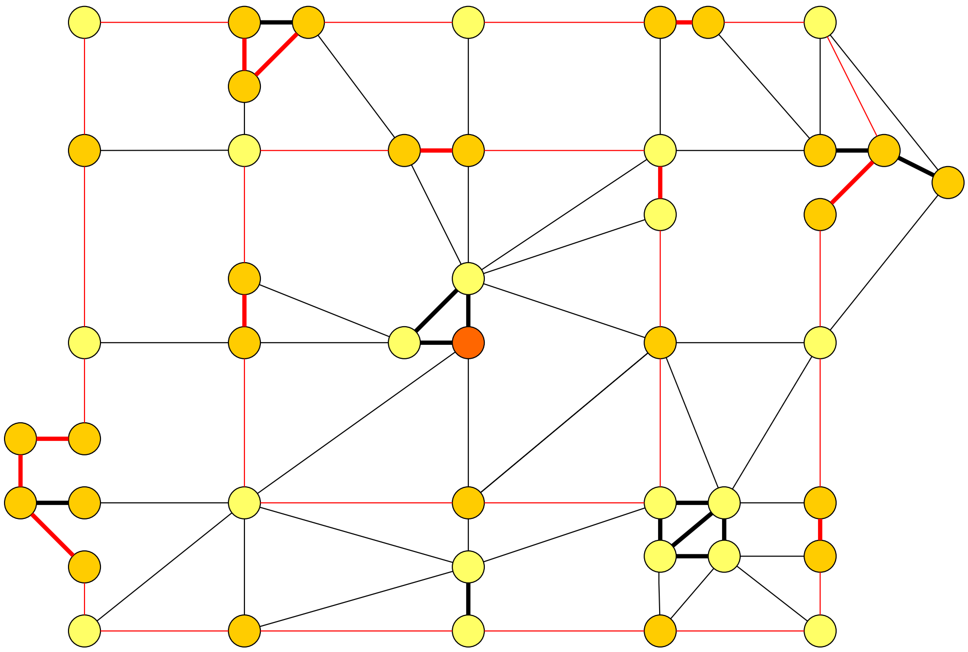


Figure 5: A graph that illustrates the idea of Lemma 3.7. Vertices in the same branch set have similar colors. Edges between vertices in the same branch set are thicker. The red cycles traverse all non-center branch sets. The dark orange colored vertex is part of the center branch set.

Proof. As a proof sketch, consider Figure 5. Its 25 branch sets form a 5×5 grid. The edges within the branch sets are depicted thicker. The center branch contains dark orange colored vertex v and the other branch sets are encountered when traversing the red cycles. They have alternate colors. The idea behind this proof is that no neighbors of a vertex in the center branch set, are outside the innermost red cycle, because the cycle can only be crossed through a vertex. Similarly there cannot be any vertices at distance at most 2 from v outside the outermost red cycle. This implies that for any grid minor H of G, there are at most 25 branch sets within distance 2 of v in G. Now, for the formal proof, there will be a case distinction. Either $\exists i, j \in [\ell] : v \in V_{h_{i,j}}$ or $v \notin V_{h_{i,j}}$ for any $i, j \in [\ell]$.

- **Case: v is in a branch set of H**, i.e. $\exists i, j \in [\ell] : v \in V_{h_{i,j}}$.
  - **Case:** $i, j \in \{3, \ldots, \ell - 2\}$. Then there exists a cycle C in G that loops through the 8 surrounding sets, from $V_{h_{i+1,j}}$ to $V_{h_{i+1,j+1}}$ to $V_{h_{i,j+1}}$ to $V_{h_{i-1,j+1}}$ to $V_{h_{i-1,j}}$ to $V_{h_{i-1,j-1}}$ to $V_{h_{i,j-1}}$ to $V_{h_{i+1,j-1}}$ and back to $H_{i+1,j}$. This cycle must exist, since there are edges between the sets by property (iii), and the sets are connected by property (i). Therefore, by planarity of G, sets $V_{h_{i',j'}}$ cannot contain vertices of $N[v]$ if $|i' - i| > 2$ or $|j' - j| > 2$.

    There will also be a cycle C' in G that will loop from $V_{h_{i+2,j}}$ to $V_{h_{i+2,j+1}}$ to $V_{h_{i+2,j+2}}$ to $V_{h_{i+1,j+2}}$ to $V_{h_{i,j+2}}$ to $V_{h_{i-1,j+2}}$ to $V_{h_{i-2,j+2}}$ to $V_{h_{i-2,j+1}}$ to $V_{h_{i-2,j}}$ to $V_{h_{i-2,j-1}}$ to $V_{h_{i-2,j-2}}$ to $V_{h_{i-1,j-2}}$ to $V_{h_{i,j-2}}$ to $V_{h_{i+1,j-2}}$ to $V_{h_{i+2,j-2}}$ to $V_{h_{i+2,j-1}}$ and back to $V_{h_{i+2,j}}$. All the vertices on this cycle have at least distance 2 to v, and by the same reason as before, all vertices outside of C' must have distance at least 3 to v. This means that only the 25 vertex sets $V_{h_{i',j'}}$ with $|i' - i| \leq 2$ and $|j' - j \leq 2|$ can contain the vertices that have distance at most 2 to v.
  - **Case:** $i \in \{1, 2, \ell - 1, \ell - 2\}$ **or** $j \in \{1, 2, \ell - 1, \ell - 2\}$.

We assume without loss of generality $i = 1, j \in \{3, \ldots, \ell - 2\}$. Then the surrounding collection of sets include the at most 5 surrounding sets $V_{h_{i',j'}}$ with $|i' - i| \leq 1$ and $|j' - j| \leq 1$ and all sets that are incident to the outer face. All sets at distance at most 2 include the at most 15 surrounding sets, all sets that are incident to the outer face and sets incident to sets incident to the outer face. However, we do not count them in our collection, so they do not contribute and the statement still holds. A similar argument can be used for other cases where $i \in \{1, 2, \ell-1, \ell-2\}$ or $j \in \{1, 2, \ell-1, \ell-2\}$.

- **Case: v is not in a branch set**, i.e. $v \notin V_{h_{i,j}}$ for any $i, j \in [\ell]$.
  If this would be the case Then there exist integers $i, j \in [\ell]$, such that a cycle in G that will loop from $V_{h_{i,j}}$ to $V_{h_{i,j+1}}$ to $V_{h_{i+1,j+1}}$ to $V_{h_{i+1,j}}$ and back to $V_{h_{i,j}}$ has vertex v inside this cycle. Therefore, every vertex outside of this cycle will have at least distance 2 to v. Analogously, we can apply the same argument to conclude that there are at most 16 sets $V_{h_{i',j'}}$ that can contain vertices at distance at most 2 from v. If v has any neighbors in G that are part of a branch set $V_{h_{i,j}}$, where $i \in \{1, \ell\}$ or $j \in \{1, \ell\}$, we can use a similar argument as above.

■

Corollary 3.8. Let G be a planar graph with distance-2-dominating set S of size k. Then a largest grid minor of G has side length at most $5\sqrt{k} + 4$.

Proof. We will apply Lemma 3.7 to all k vertices in distance-2-dominating set S. We obtain the result that $|\bigcup_{v \in S}\{V_{h_{i,j}} \mid V_{h_{i,j}} \cap d_G^2(v) \neq \emptyset, i, j \in \{3, \ldots, \ell - 2\}\}| \leq 25k$. Since S is a distance-2-dominating set, this collection contains all non-empty branch sets V_h in G at distance at least 2 to the outer face. Since branch sets are non-empty, we conclude that the largest grid minor contains at most $25k$ branch sets at distance at least 2 to the outer plane. This means that the length of the grid containing those sets is at most $5\sqrt{k}$. Since there can be at most 2 extra layers of sets, the side length of the total grid is at most $5\sqrt{k} + 4$, . ■

The largest grid minor can be used as a bound for the treewidth of a planar graph up to a constant, as stated in the Grid Minor Theorem, for which a proof can be found in [17].

Theorem 3.9 (Grid Minor Theorem)**.** Let G be a planar graph, whose largest grid minor has size $\ell \times \ell$. Then the treewidth of G is at most $4.5 \cdot \ell$ [17].

Corollary 3.10. Let G be a planar graph with distance-2-dominating set S of size k. Then G has treewidth at most $22.5\sqrt{k} + 18$.

Proof. The largest grid minor of G has side length at most $5\sqrt{k} + 4$, as stated in Corollary 3.8. Furthermore, the Grid minor theorem, Theorem 3.9, states that the treewidth of a planar graph G is bounded by 4.5 times the side length of the largest grid minor of G. Combining these results, we obtain a treewidth bound of $4.5 \cdot (5\sqrt{k} + 4) = 22.5\sqrt{k} + 18$. ■

We can combine these results to obtain the following lemma.

Lemma 3.11. There is a polynomial-time algorithm that, given an instance (G, k, t) of Maximum Weighted Independent Set on planar graphs, constructs an equivalent instance (G', k, t) and a nice tree decomposition of (T, χ) of G' of width at most $33.75\sqrt{k} + 27$.

Proof. First we apply Reduction rule 3.3 to the instance (G, k, t) to obtain an equivalent instance (G', k, t) in polynomial time. In Lemma 3.4 it is shown that G' contains a distance-2-dominating set of size at most k. Then Corollary 3.10 gives us that the treewidth of this G' is bounded by $22.5\sqrt{k} + 18$. Furthermore, it is known that for planar graphs, within polynomial time a tree-decomposition with a 1.5-approximation of the treewidth can be constructed [18] and transformed into a nice tree decomposition [9, Chapter 14.2]. This means that in polynomial time, we can obtain a nice tree decomposition of this graph G' of width at most $(22.5\sqrt{k} + 18) \cdot 1.5 = 33.75\sqrt{k} + 27$.

■

3.3 A dynamic programming algorithm for Maximum Weighted Independent Set on graphs with bounded treewidth

The algorithm

The algorithm has the following input and output:

MAXIMUM WEIGHTED INDEPENDENT SET	**Parameter:** ℓ
Input: A graph G equipped with a weight function $\omega : V(G) \to \mathbb{N}^+$, an integer k and a (semi-)nice tree-decomposition (T, χ) of G of width at most ℓ.	
Output: The maximum weighted independent set of G of size exactly k and its weight.	

Let $r \in V(T)$ denote the root node of T. Let $t \in V(T)$ denote a node of the tree decomposition. Let G_t denote the subgraph of G induced by the vertices of the bags of the nodes in the subtree rooted at node t. For a node $t \in V(T)$, a subset $Y \subseteq \chi(t)$ of its bag, and an integer $m \in \{0, \ldots, k\}$, let the table entry $T[t, Y, m]$ be defined as:

$$T[t, Y, m] := \max\{\omega(W) \mid W \text{ is an independent set of } G_t \wedge |W| = m \wedge W \cap \chi(t) = Y\}$$

Informally, $T[t, Y, m]$ denotes the maximum weight of an independent set of size exactly m of the graph G_t where we include from $\chi(t)$ exactly the vertices of Y.
We fill in this table for all values of Y and m from the bottom up. We now give the formula for the recursion for all 4 types of nodes in the tree decomposition.

- **Leaf bag**
  Let $\chi(t)$ be a leaf bag, so it contains only vertex v. Then we have

$$\begin{aligned} T[t, \emptyset, 0] &= 0 && \\ T[t, \emptyset, m] &= -\infty && \text{if } m \neq 0 \\ T[t, \{v\}, 1] &= \omega(v) && \\ T[t, \{v\}, m] &= -\infty && \text{if } m \neq 1 \end{aligned}$$

  In the second and the last case, we set the value to $-\infty$, because the problem is infeasible, since we take the maximum over an empty set.

- **Introduce bag**
  Let $\chi(t)$ be an introduce bag that contains the vertices of the bag $\chi(j)$ as well as vertex v, where j is the child of node t. Then we have

$$\begin{aligned} T[t, Y, m] &= T[j, Y, m] && \text{if } v \notin Y \\ T[t, Y \cup \{v\}, m] &= T[j, Y, m-1] + \omega(v) && \text{if } N(v) \cap Y = \emptyset \\ T[t, Y \cup \{v\}, m] &= -\infty && \text{if } N(v) \cap Y \neq \emptyset \end{aligned}$$

  In the last case, we set the value to $-\infty$, because the problem is infeasible, since we take the maximum over an empty set.

- **Forget bag**
  Let $\chi(t)$ be a forget bag such that node t has a child j and $\chi(j)$ contains the same vertices as $\chi(t)$ as well as vertex v. Then we have

$$T[t, Y, m] = \max\{T[j, Y, m], T[j, Y \cup \{v\}, m]\}$$

  This holds since we can take the same independent set W for i as for child j.

- **Join bag**
  Let $\chi(t)$ be a join bag that contains the same vertices as $\chi(j)$ and $\chi(j')$, the bags of the children of node t. Then we have

$$T[t,Y,m] = \max_{n\in\{|Y|,\ldots,m\}} \{T[j,Y,n] + T[j',Y,m+|Y|-n]\} - \omega(Y)$$

  This recursion takes the maximum weight independent set where it chooses exactly the vertices Y from both sets, so the children can share a total of $m-|Y|$ vertices. At the end the vertices of Y are added twice, so we subtract them to account for that. Here we use the fact that the vertices in $\chi(t)$ are a separator in G for the vertices in the trees rooted at the children of node t

Output
First we calculate $T[t,Y,m]$ for all $Y \subseteq \chi(t)$ and all $m \leq k$ for each leaf node t, and then we fill the table traversing the tree from the bottom up. The output for this dynamic programming algorithm will be

$$\max_{Y\subseteq\chi(r)} \{T[r,Y,k]\}$$

Although the output is a number, namely the maximum total weight of an independent set of size k, it is trivial to reconstruct a corresponding independent set traversing the table top down, tracking the chosen maxima. This method is explained in more detail in 'Introduction to Algorithms' by Cormen et al. [19, Chapter 15]. We will add an encoding of this set to the output of the algorithm.

Running time and application

First, we will show that Maximum Weighted Independent Set can be solved by this dynamic programming algorithm in $\mathcal{O}^*(2^\ell)$ time if a valid tree decomposition of width at most ℓ is provided. Then, we will prove the first main result, Theorem 1.1 from the introduction.

Lemma 3.12. Maximum Weighted Independent Set can be solved by the dynamic programming algorithm described in this section in $\mathcal{O}^*(2^\ell)$ time if a nice tree decomposition of width at most ℓ is provided.

Proof. It is argued above that the value of each cell is indeed the maximum weighted independent set of size exactly k of the graph induced by the vertices in the subtree rooted at t, in which exactly Y is included from the vertices in t. We are left to show that the running time is $\mathcal{O}^*(2^\ell)$. Let n denote the size of $V(G)$. First note that the number of cells is bounded by $(24\cdot(\ell+2)\cdot n)\cdot 2^{\ell+1}\cdot n$, because the number of bags is bounded by $24\cdot(\ell+2)\cdot n$ [9, Chapter 14.3]. The calculation of the value of a cell takes linear time in k if t is a join bag and constant time otherwise. When all cells are calculated, we can efficiently look up the maximum value v as stated above. We will then compare this value to t and output Yes if $v \geq t$ and No otherwise. This means that we can solve Maximum Weighted Independent Set in running time $(24\cdot(\ell+2)\cdot n)\cdot 2^{\ell+1}\cdot n\cdot k = \mathcal{O}(\ell\cdot n^2\cdot k\cdot 2^\ell) = \mathcal{O}^*(2^\ell)$ ■

Theorem 1.1 from the introduction is restated below, which we will prove using Lemma 3.12.

Theorem 1.1. Maximum Weighted Independent Set on planar graphs can be solved in in $\mathcal{O}^*(2^{33.75\sqrt{k}})$ time.

Proof. We will apply the algorithm of this section to the weighted graph G and parameter k of our instance (G,k,t) of MWIS. The input further consists of the corresponding nice tree-decomposition of width at most $33.75\sqrt{k}+27$ that we have obtained with the algorithm provided by Lemma 3.11. Now we can apply Lemma 3.12 and conclude that MWIS is solvable in $\mathcal{O}^*(2^\ell) = \mathcal{O}^*(2^{33.75\sqrt{k}})$ time. ■

4 Kernelization of MWIS

In this section we will show how to reduce the instance of MWIS to an equivalent instance with $\mathcal{O}(k^3)$ vertices. To achieve this, we first present a bounded treewidth-5-modulator in Lemma 4.4. That is a vertex set $X \subseteq V(G)$, such that the treewidth of $G - X$ is only 5. Then, a protrusion decomposition will be obtained. Finally, we will use this decomposition to obtain a kernel with $\mathcal{O}(k^3)$ vertices.

4.1 Treewidth modulator

In this section we will prove that after application of Reduction rule 3.3, the graph has a treewidth-5-modulator S of size at most $5k - 2$.

Lemma 4.1. Let G be a planar graph with distance-2 dominating set S of size k. Then each connected component C of G has a connected distance-2-dominating set of size at most $5|S_C|-4$, where $S_C := S \cap V(C)$. Furthermore, this set can be found in polynomial time.

Proof. Let C be a connected component of G. Then S_C is a distance-2-dominating set of C. We will construct a connected distance-2-dominating set $\overline{S_C}$ by adding vertices to S_C. Then $\overline{S_C}$ is always a distance-2-dominating set, because it contains one. This means we add vertices to S_C to construct a connected set.
Assume $G[\overline{S_C}]$ contains ℓ connected components. Then, if $\ell = 1$, $\overline{S_C}$ is connected. If $\ell \neq 1$, we can take 2 vertices u and v in different connected components of $G[\overline{S_C}]$ with $d_G(u,v) \leq 5$. These vertices always exist, since otherwise there must be a vertex at distance at least 3 to all vertices of $\overline{S_C}$, while $S_C \subseteq \overline{S_C}$ is a distance 2 dominating set of C and $G[C]$ is a connected graph. These vertices can be found in polynomial time using for example breadth-first search on vertices in $\overline{S_C}$. Then we can add the 4 vertices on the shortest path between u and v to $\overline{S_C}$ to obtain a distance-2-dominating set that consists of $\ell - 1$ connected components.
We begin with $\overline{S_C}$ consisting of $|S_C|$ vertices and at most S_C connected components. Then we reduce the number of connected components to 1, by adding at most $4 \cdot (|S_C|-1)$ vertices to $\overline{S_C}$. We end up with a connected distance-2-dominating set of size $|S_C|+4 \cdot (|S_C|-1) = 5S_C - 4$. ■

Corollary 4.2. Let G be a planar graph with distance-2 dominating set S of size k. Then each connected component C of G has a connected distance-2-dominating set of size at most $5|S_C|-2$ that is incident to the outer face of $G[C]$, where $S_C := S \cap V(C)$. Furthermore, this set can be found in polynomial time.

Proof. Lemma 4.1, provides us with a connected distance-2-dominating set of size at most $5|S_C|-4$. we can easily transform this into a connected distance-2-dominating set that is incident to the outer face of the connected component. We add an arbitrary vertex v incident to the outer face, to the set. If necessary, we also add a vertex w that is on a path of length at most 2 from v to a vertex in the connected distance-2-dominating set. Now we have constructed a connected distance-2-dominating set that it is incident to the outer face. This set contains at most $5|S_C|-2$ vertices. ■

In order to bound the treewidth of the graph, we first need the following well known theorem. A proof can be found in [16].

Theorem 4.3. The treewidth of a k-outerplanar graph is at most $3k - 1$ [16].

Lemma 4.4. Let G be a planar graph with a distance 2-dominating set S of size k. Then G has treewidth-5-modulator of size at most $5k - 2$. Furthermore, it can be found in polynomial time.

Proof. We will bound the treewidth of an arbitrary connected component C of $G \setminus \tilde{S}$, which is a bound to the treewidth of the graph $G \setminus \tilde{S}$, where $\tilde{S}$ is a connected distance-2-dominating set of size at most $5|S_C|-2$ in C, incident to the outer face. We can find such a set in polynomial time

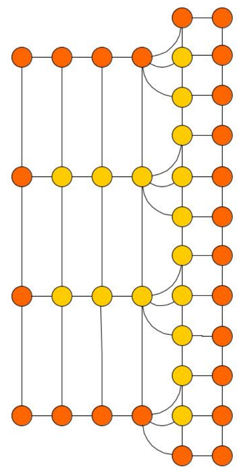


Figure 6: An example of a 2-outerplanar graph with treewidth exactly 5. This is discussed after Lemma 4.4.

by Corollary 4.2. We will now show that $\tilde{S}$ is a treewidth-5-modulator of G.
Note that all vertices at distance 1 from $\tilde{S}$ are incident to the outer face of C in $V(C) \setminus \tilde{S}$, since $\tilde{S}$ is a connected set incident to the outer face. Furthermore, note that all vertices at distance 2 from $\tilde{S}$ are neighbors of at least one vertex at distance 1 from $\tilde{S}$. Combining these notions, we can conclude that $G \setminus \tilde{S}$ is 2-outerplanar.
Now we can apply Theorem 4.3 and conclude that the treewidth of C is at most $3 \cdot 2 - 1 = 5$. Therefore $\tilde{S}$ is a treewidth-5-modulator of G of size at most $5k-2$ that can be found in polynomial time. ■

The treewidth bound of 5 is tight since we can be left with a graph consisting of a 4×4-grid connected to a 2×12-grid, as shown in Figure 6. This graph can be a connected component of $G \setminus \tilde{S}$, because it is 2-outerplanar. This graph has treewidth exactly 5 as argued by Kammer and Tholey in [20].

We apply Lemma 4.4 to to obtain the main result of this subsection.

Lemma 4.5. There is a polynomial time algorithm that, given an instance (G, k, t) of Maximum Weighted Independent Set on planar graphs, constructs an equivalent instance (G', k, t), such that G' has a treewidth-5-modulator S of size at most $5k-2$. Furthermore, the algorithm provides a tree decomposition of $G - S$ of width at most 5.

Proof. First, in polynomial time, we can transform the instance (G, k, t) of MWIS into an equivalent instance (G', k, t), in which G' has a distance-2-dominating set of size at most k, using the result of Lemma 3.4. Then we can apply Lemma 4.4 to G', to construct a treewidth-5-modulator S of size at most $5k - 2$ in polynomial time. Finally, we can use Bodlaenders Theorem, which states that for a graph with a constant treewidth, we can construct a tree decomposition with that width in linear time [21]. Therefore, a tree decomposition of $G - S$ of width at most 5 can be obtained in polynomial time. ■

4.2 Protrusion Decomposition

In this subsection, we will construct a protrusion decomposition of the graph G obtained after the application of Lemma 4.5. This graph has a treewidth-5-modulator S of size $5k-2$. With the help of Lemma 4.6, we will construct a superset Z of S that contains at most 25 times as many vertices as S. The number of vertices in this set is still linear in k, but it contains the vertices that are the most important for the structure of G. The remaining vertices form a collection of connected components whose treewidth is bounded by a constant. This protrusion decomposition will later be used to obtain a kernel for Maximum Weighted Independent Set on planar graphs.

Now, we will state and prove Lemma 15.14 from 'Kernelization: theory of parameterized preprocessing', by Fedor V Fomin et al. [9, Chapter 15.3].

Lemma 4.6. If a planar graph G has a treewidth-η-modulator S, then G has a $((4(\eta+1)+1)|S|, (24(\eta+1)+6)|S|, 3\eta+2)$-protrusion decomposition, such that S is contained in the bag of the root node of the protrusion decomposition. Furthermore, there is a polynomial time algorithm that, given G, S, and a tree decomposition (T, χ) of $G-S$ of width at most η, computes such a protrusion decomposition of G.

Proof. We will repeat the proof of Lemma 15.14 from [9, Chapter 15.3], but since there is a small error in the estimation of the degree of the root node, we will present a correct proof.

First, by applying Lemma 15.13 from [9, Chapter 15.3], we construct a set Z with $S \subseteq Z$, such that $|Z| \leq 4(\eta+1)|S|+|S|$, and each connected component of $G-Z$ has at most 2 neighbors in S, and at most 2η neighbors in $Z \setminus S$. Group the components into groups with the same neighborhood in S. More precisely, let $C_1, \dots, C_t$ be the connected components of $G-Z$. Define sets $X_1, \dots, X_\ell$ with the following properties. For each $i \leq t$, there is exactly one $j \leq \ell$ such that $C_i \subseteq X_j$, and for all $j' \neq j$ we have $C_i \cap X_{j'} = \emptyset$. Furthermore, for all i, i' and j, it holds that $C_i \subseteq X_j$ and $C_{i'} \subseteq X_j$ if and only if $N(C_i) = N(C_j)$. The definition of the sets $X_1, \dots, X_\ell$ immediately gives a way to compute them from $C_1, \dots, C_t$.
For each $i \leq \ell$ we make a tree-decomposition (T_i, χ_i) of $G[X_i \cup N(X_i)]$ by starting with the tree-decomposition (T, χ) of $G-S$, removing all vertices not in X_i from all bags of the decomposition, turning this into a nice tree decomposition of $G[X_i]$ using Lemma 14.23 [9, Chapter 14.3] and, finally, inserting $N(X_i)$ into all bags of the decomposition. The width of (T_i, χ_i) is at most $\eta + |N(X_i)| \leq 3\eta + 2$.
We now make a tree-decomposition $(\hat{T}, \hat{\chi})$ that is to be our protrusion decomposition. The tree $\hat{T}$ is constructed from $T_1, \dots, T_\ell$ by adding a new root node r and connecting r to an arbitrary node in each tree T_i. We set $\hat{\chi}(r) = Z$ and, for each node $a \in V(\hat{T})$ that is in the copy of T_i in $\hat{T}$, we set $\hat{\chi}(a) = \chi_i(a)$. It is easy to verify that $(\hat{T}, \hat{\chi})$ is indeed a tree-decomposition of G and that every node $a \in V(T)$ except for r satisfies $|\chi(a)| \leq 3\eta+2$. Thus, $(\hat{T}, \hat{\chi})$ is a $((4(\eta+1)+1)|S|, \ell, 3\eta+2)$-protrusion decomposition of G. To prove the statement of the lemma it is sufficient to show that $\ell \leq (24(\eta+1)+6)|S|$. Because the neighborhoods of the sets $X_1, \dots, X_\ell$ are distinct, there are at most $|Z| \leq (4(\eta+1)+1)|S|$ sets X_i such that $|N(X_i)| = 1$. We can interpret the number of sets X_i such that $|N(X_i)| = 2$ as the number of edges in the planar graph $G[Z]$. Using Eulers formula [9, Chapter 13.1], we conclude that there are at most $3|Z|-6 \leq (12(\eta+1)+3)|S|$ sets X_i such that $|N(Xi)| = 2$. Finally, by Lemma 13.3 of [9, Chapter 13.1], there are at most $2|Z|-4 \leq (8(\eta+1)+2)|S|$ sets X_i such that $|N(X_i)| \geq 3$. It follows that $\ell \leq 6|Z| \leq (24(\eta+1)+6)|S|$, as claimed. ■

Now we will combine the results of Lemma 4.5 and Lemma 4.6 to summarize the results of this subsection.

Lemma 4.7. There is a polynomial time algorithm that, given an instance (G, k, t) of Maximum Weighted Independent Set on planar graphs, constructs an equivalent instance (G', k, t) and gives a nice $(\mathcal{O}(k), \mathcal{O}(k), \mathcal{O}(1))$-protrusion decomposition of G' in polynomial time.

Proof. Let (G, k, t) be an instance of MWIS. First we can apply Lemma 4.5 to that instance to obtain an equivalent instance (G', k, t), a treewidth-5-modulator S of size at most $5k-2$ and a tree decomposition of $G-S$ of width at most 5. Then we can apply Lemma 4.6 that computes an (α, β, t)-protrusion decomposition of G in polynomial time, in which $\alpha = 25|S| = 125k - 50$, $\beta = 150|S| = 750k - 300$ and $t = 17$. Finally we can transform this protrusion decomposition into a nice protrusion decomposition with the same parameters in polynomial time [9, Chapter 15.4]. ■

4.3 A kernel with $\mathcal{O}(k^3)$ vertices

In this section, we use the obtained protrusion decomposition to find a kernel of MWIS with $\mathcal{O}(k^3)$ vertices. First, we describe how we can construct an equivalent instance of MWIS with $\mathcal{O}(k^3)$ vertices. Then, we will transform this into a kernel. Recall from Definition 2.3 that for a polynomial kernel not only the graph, but the complete new input has to be bounded by a polynomial function of the complexity parameter. Later, we will ensure that all parameters are bounded by a polynomial function of k.

A bound on the number of vertices

Lemma 4.8. Given an instance (G, k, t) of Maximum Weighted Independent Set and a nice protrusion decomposition (T, χ) of G with parameters (α, β, t), where $\alpha = \mathcal{O}(k)$, $\beta = \mathcal{O}(k)$ and $t = \mathcal{O}(1)$, we can transform this instance into an equivalent instance (G', k, t) in polynomial time with $|V(G')|= \mathcal{O}(k^3)$.

Proof. First, we will describe the reduction rule. Then, we will present the results. We end this proof by proving the safeness of the reduction rule.
For each of the β children i of the root node r of protrusion decomposition (T, χ), consider the subgraph $G[\chi(T_i)]$ formed by the vertices in the subtree of T rooted at node i. By definition, (T_i, χ_i) is a semi nice tree decomposition of $G(\chi(T_i))$. For each subset Y of $\chi(i)$, for each size $m \in [k - |Y|]$, compute a maximum weighted independent set $W_{i,Y,m}$ of $G[\chi(T_i) \setminus \bigcup_{y \in Y} N[y]]$. This can be done in polynomial time by applying the dynamic programming algorithm of Section 3.3 to the graph $G[\chi(T_i) \setminus \bigcup_{y \in Y} N[y]]$. The necessary corresponding tree decomposition can be obtained in polynomial time by deleting all neighbors of a vertices in Y from the nice tree decomposition (T_i, χ_i). This way we have computed a maximum weighted independent set $W_{i,Y,m}$ for all β children of the root i and all at most 2^{t+1} possible $Y \subseteq \chi(i)$ in polynomial time.
Next, we construct G'. Let G' be the subgraph of G induced by the root bag $\chi(r)$ together with the union over all children i of the root node and all possible subsets $Y \subseteq \chi(i)$ of the set $W_{i,Y,m}$. Then $|V(G')|$ is at most $\alpha + \beta \cdot 2^t \cdot \sum_{m=1}^{k} m = \alpha + \beta \cdot 2^{t-1} \cdot k(k-1) = \mathcal{O}(k^3)$. It remains to prove the rule is safe.
Let W be a vertex set corresponding to a solution of MWIS that minimizes the number of vertices in $W \cap (G \setminus G')$. If this quantity is empty, the reduction rule is safe. Now assume W contains a vertex v_i in subtree T_i that is in $G \setminus G'$. Let $Y = W \cap \chi(i)$ and let $m = |W \cap \chi(T_i)| - |Y|$. Then we can construct an alternative solution set $W' := W \setminus \chi(T_i) \cup W_{i,Y,m} \cup Y$, where $W_{i,Y,m}$ is the independent vertex set of $\chi(T_i) \setminus \bigcup_{y \in Y} N[y]]$ of size m of maximal weight that was not removed by the reduction rule. This implies that $W_{i,Y,m} \cup Y$ is a maximum weighted independent set of $G[\chi(T_i)]$ of size $m + |Y|$ that is a subset of G'.
We will now argue why W' forms an independent set in G. Firstly, we know that $W \setminus \chi(T_i) \cup W_{i,Y,m}$ forms an independent set in G, because $\chi(i)$ is a separator in G. Secondly, $W_{i,Y,m} \cup Y$ forms an independent set in G because $W_{i,Y,m}$ has no vertices from the neighborhood of vertices in Y. Lastly, $W \setminus \chi(T_i) \cup Y$ forms an independent set in G, because it is a subset of W which is an independent set in G. Combining these, we get that $W' = (W \setminus \chi(T_i) \cup W_{i,Y,m}) \cup Y$ forms a maximum weighted independent set in G. Furthermore we can check that it has size exactly k and that it is at least as heavy as W, while having fewer vertices in $G \setminus G'$. This implies that the reduction rule is safe. ■

Combining Lemma 4.7 and Lemma 4.8 gives the following result:

Lemma 4.9. There is a polynomial time algorithm that, given an instance (G, k, t) of Maximum Weighted Independent Set on planar graphs, constructs an equivalent instance (G', k, t), such that $|V(G')|= \mathcal{O}(k^3)$ vertices.

Proof. First we apply Lemma 4.7 to obtain an equivalent instance (G^*, k, t) of MWIS and a nice $(\mathcal{O}(k), \mathcal{O}(k), \mathcal{O}(1))$-protrusion decomposition of G^* in polynomial time. Then we can apply

Lemma 4.8 to the instance (G^*, k, t) with that corresponding protrusion decomposition to obtain a new instance (G', k, t) of MWIS with $|V(G')|= \mathcal{O}(k^3)$. ■

A bound on the other parameters of the input
We will now consider the graph obtained after application of the reduction rule described in Lemma 4.8, that has at most $\mathcal{O}(k^3)$ vertices. We will bound the total length of the input by a polynomial function of the parameter k, which makes the new instance a valid kernel for the original problem. We will use this to prove the second main result of this paper, Theorem 1.2.

We will use Corollary 2 in [14] to obtain new weights to bound the total encoding size of the input. We can use this algorithm to transform an instance (G, k, t) of Maximum weighted Independent Set where $|V(G')|= \mathcal{O}(k^3)$ to an equivalent instance (G', k, t') whose total encoding size is bounded by polynomial function of k.

Theorem 4.10 ([14, Corollary 2])**.** There is an algorithm that, given a vector $w \in \mathbb{Q}^r$ and a rational $W \in \mathbb{Q}$, in polynomial time finds a vector $\overline{w} \in \mathbb{Z}^r$ with $\|\overline{w}\|_\infty= 2^{\mathcal{O}(r^3)}$ and an integer $\overline{W} \in \mathbb{Z}$ with total encoding length $\mathcal{O}(r^4)$, such that $w \cdot x \leq W$ if and only if $\overline{w} \cdot x \leq \overline{W}$ for every vector $x \in \{0,1\}^r$.

Now we will slightly adjust this theorem in the next corollary, which we will then use to obtain a polynomial size kernel.

Corollary 4.11. There is an algorithm that, given a vector $w \in \mathbb{Q}^r$ and a rational $W \in \mathbb{Q}$, in polynomial time finds a vector $\overline{w} \in \mathbb{Z}^r$ with $\|\overline{w}\|_\infty= 2^{\mathcal{O}(r^3)}$ and an integer $\overline{W} \in \mathbb{Z}$ with total encoding length $\mathcal{O}(r^4)$, such that $w \cdot x \geq W$ if and only if $\overline{w} \cdot x \geq \overline{W}$ for every vector $x \in \{0,1\}^r$.

Proof. We have to show that the statement of Theorem 4.10 also holds if we use '$\geq$'-signs in the last line, so we will show that the expression '$w \cdot x \leq W$ if and only if $\overline{w} \cdot x \leq \overline{W}$ for every vector $x \in \{0,1\}^r$' is equivalent to the expression '$w \cdot x \geq W$ if and only if $\overline{w} \cdot x \geq \overline{W}$ for every vector $x \in \{0,1\}^r$'. This follows from multiplying both sides of the inequalities by -1. This gives $-w \cdot x \geq -W \iff w \cdot x \leq W \iff \overline{w} \cdot x \leq \overline{W} \iff -\overline{w} \cdot x \geq \overline{-W}$ for every vector $x \in \{0,1\}^r$. This implies that for each pair (w, W), we can compute a pair $(\overline{w}, \overline{W})$ that meets the requirements by first multiplying w and W by -1. Then we can compute a pair $(\overline{-w}, \overline{-W})$ using Theorem 4.10 and finally we multiply them by -1 again to obtain the pair $(-(\overline{-w}), -(\overline{-W}))$ and define $(w', W') = (-(\overline{-w}), -(\overline{-W}))$, a pair with total encoding length $\mathcal{O}(r^4)$. This yields that there exists an algorithm that, given a vector given $w \in \mathbb{Q}^r$ and a rational $W \in \mathbb{Q}$, in polynomial time finds a vector $w' \in \mathbb{Z}^r$ with $\|w'|_\infty= 2^{\mathcal{O}(r^3)}$ and an integer $W' \in \mathbb{Z}$ with total encoding length $\mathcal{O}(r^4)$, such that $w \cdot x \geq W$ if and only if $w \cdot x \geq W$ for every vector $x \in \{0,1\}^r$.
■

We will now use this corollary to obtain this powerful lemma that describes a reduction rule that transforms an instance of MWIS on a small graph to an instance whose encoding length is bounded by a polynomial function of k.

Lemma 4.12. There is a polynomial time algorithm that transform an instance (G, k, t) of Maximum Weighted Independent Set on planar graphs with $|V(G)|= \mathcal{O}(k^3)$ to an equivalent instance (G', k, t') whose encoding length is bounded by a polynomial function of k. This new weighted graph G' also has $\mathcal{O}(k^3)$ vertices.

Proof. We will use Theorem 4.11 to obtain a new weighted graph G' and a new target t'. To achieve this, we first arbitrarily number the vertices v_i in G and then set w to be a vector in $\mathbb{Z}^{|V(G)|}$ in which the i'th entry corresponds to the weight of vertex v_i. Furthermore, we take W to be the target value t. Then the algorithm yields vectors $\overline{w}$ and $\overline{W}$. We define G' to be the same graph as G, but we set the weight of each vertex v_i to equal the i'th component of $\overline{w}$. Furthermore, we set the new target t' to equal $\overline{W}$.

Now, we will first show that this reduction is safe and then show that the encoding length is bounded by a polynomial function of k.
Now we will proof the safeness by first assuming that (G, k, t) is a yes-instance. In that case there exists an independent set X of size exactly k and total weight at least t. This means that we can construct a vector $x \in \{0, 1\}^{|V(G)|}$ whose entry x_i equals 1 if and only if $v_i \in X$. This vector satisfies the inequality $w \cdot x \geq W$. Now Corollary 4.11 guarantees that $\overline{w} \cdot x \geq \overline{W}$. This is a witness for the yes-instance of (G', k, t'). The other direction where we assume that (G, k, t) is a no-instance can be proven analogously.
Now we show that the encoding length of the new instance (G', k, t) is indeed bounded by a polynomial function of k. From Corollary 4.11, it follows directly that the new weights are bounded by a cubic function of the number of weights. The number of weights equals the number of vertices, which is $\mathcal{O}(k^3)$. This means that we can bound the encoding length of the new weight of the vertices by $\mathcal{O}(k^9)$ and the encoding length of the new target t' by $\mathcal{O}(k^{12})$.
This implies that the new weighted graph G', the complexity parameter k and the target t' have an encoding length bounded by a polynomial function of k. ■

We will now restate Theorem 1.2 from the introduction, which we will prove using Theorem combining Lemma 4.12 and Lemma 4.9. This is the second main result of this paper.

Theorem 1.2. Maximum Weighted Independent Set on planar graphs admits a kernel with $\mathcal{O}(k^3)$ vertices.

Proof. We need to show that there exists a polynomial time algorithm that transforms an instance (G, k, t) of MWIS into an equivalent instance (G', k', t') that has total encoding size polynomial in k. Furthermore we require $|V(G')| = \mathcal{O}(k^3)$.
First we will use Lemma 4.9 to obtain an equivalent instance (G^*, k, t) in which G^* has $\mathcal{O}(k^3)$ vertices. Now we can apply the reduction rule described in Lemma 4.12 to transform this instance to an instance (G', k', t') that has total encoding size polynomial in k. Here $|V(G')| = \mathcal{O}(k^3)$. This implies that we have obtained a polynomial size kernel with $\mathcal{O}(k^3)$ vertices for MWIS. ■

Reflection on kernel size

It is tempting to conjecture that for the graph induced by the vertices of a main subtree, the maximum weighted independent sets W_i of all sizes $i \in \{1, \ldots, k\}$ have sufficient overlap such that $|\cup_{j=1}^{k} W_j| = \mathcal{O}(k)$. This would imply that the obtained kernel has only $O(k^2)$ vertices. However, this is in general not the case. In the following lemma, we will prove that for all k we can construct a forest in which the total number of vertices left by the kernelization step of the previous subsection, has $\Omega(k^2)$ vertices.

Lemma 4.13. For each $k \geq 1$ there exists a weighted forest F such that $|\bigcup_{j=1}^{k} W_j| = \Omega(k^2)$ in which W_i is a maximum weighted independent set of size i in F for $1 \leq i \leq k$.

Proof. We will prove this lemma for even integers k. The generalization to all integer k can be proven by adding a heavy isolated vertex that would be in all independent sets.
Let k be any given integer and assume k to be even. Let G be a forest of $k/2$ connected components, $C_1, \ldots, C_{\frac{k}{2}}$. Let component C_i be the complete bipartite graph $K_{1,2i}$. Let this be a tree, rooted at the vertex with the highest degree. For each connected component C_i with $i > 1$ we assign weights $(2i - 1) \cdot k^{i-1}$ to the root of C_i and k^{i-1} to the other vertices of C_i. In connected component C_1, the root vertex gets assigned weight 3 and the other 2 vertices weight 2. The example for $k = 10$ is illustrated in Figure 7. The main idea is that the only way to improve the selected weight in a connected component, is to take all vertices at the bottom, but that is only possible if the size constraint of the independent set allows it.

First we will show that for $i \leq \frac{1}{2}k$, W_i, the maximum weighted independent set of size i in G is a subset of the root vertices of all trees. Suppose towards a contradiction that W_i contains a non-root vertex in C_j. Then either W_i contains all non-root vertices of C_j, just one non-root vertices or more than one but not all. We will derive a contradiction using case distinction.

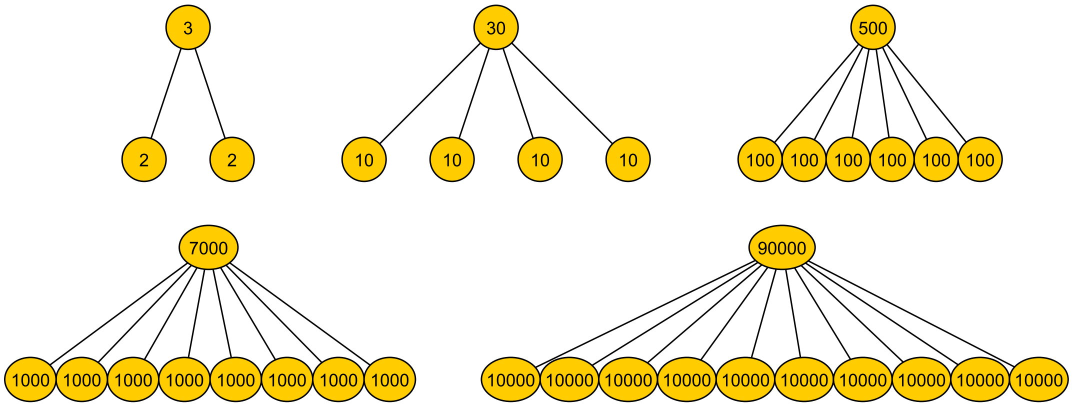


Figure 7: The graph G for $k = 10$ in Lemma 4.13.

- **If W_i contains one non-root vertex of C_j**, the weight of W_i can be improved by replacing that vertex by the root of C_j, which leads to a contradiction.
- **If W_i contains ℓ non-root vertices of C_j with** $2 \leq \ell \leq i$, the the weight of W_i can be improved by replacing these vertices by the root vertex of C_j and root vertices of connected components that have no vertices in W_i. Note this is possible since the number of connected components is at least i, the size of W_i. This new set is heavier, so that leads to a contradiction.
- **If W_i contains all non-root vertices of C_j**, then there is a connected component $C_{j'}$ with $j' > j$, such that $W_i \cap C_{j'} = \emptyset$. However, then we can improve the weight by replacing a vertex in C_j by any vertex in $C_{j'}$ that is at least k times as heavy. This also leads to a contradiction.

Therefore for $i \leq \frac{1}{2}k$, W_i, the maximum weighted independent set of size i in G is a subset of the root vertices of all trees. Now we will show that for $\frac{1}{2}k < i \leq k$, W_i consists exactly of all the children of $C_{i-\frac{1}{2}k}$ and the roots of C_j for $j > i - \frac{1}{2}k$. Suppose towards a contradiction that W_i contains another vertex of F. Then it either contains a non-root vertex of C_j, the root of $C_{i-\frac{1}{2}k}$ or a vertex in $C_{j'}$ for $j' < i - \frac{1}{2}k$. We will derive a contradiction using a case distinction.

- **If W_i contains a non-root vertex of C_j**, we can improve the solution by replacing the vertices in $W_i \cap C_j$ by the root vertex of C_j and maybe some vertices of other connected components. This might involve replacing a root vertex in a lighter connected component by its children to make room for the independent set. This alternative solution is definitely heavier, since for any j and ℓ, the maximum weighted independent set of size ℓ in $F \setminus C_j$ is always heavier than the maximum weighted independent set of size $\ell - 1$ in $F \setminus C_j$. The weight of $W_i \cap C_j$ has not decreased in the mean time, since i is not big enough to have all non-root vertices of c_j in W_i. Therefore we could improve upon the weight of a maximum weighted independent set and this leads to a contradiction.
- **If W_i contains the root of $C_{i-\frac{1}{2}k}$**, we can improve the solution by replacing all remaining vertices by the non root vertices of $C_{i-\frac{1}{2}k}$. By the previous case, we know that in C_j, the root vertex is picked for W_i if $j > i - \frac{1}{2}k$. This replacement increases the weight of $W_i \cap C_{i-\frac{1}{2}k}$ by $k^{i-\frac{1}{2}k-1}$. This increase of weight is higher than the decrease in weight in the other connected components by the exponential behavior of the weights. This means that the total weight can be improved, which leads to a contradiction.

- **If W_i contains a vertex in $C_{j'}$ for** $j' < i - \frac{1}{2}k$, it does not contain all children of $C_{i-\frac{1}{2}k}$ or not all roots of C_j with $j > i - \frac{1}{2}k$. The solution can be improved by replacing the vertex in $W_i \cap C_{j'}$ by such a missing vertex. This improves the weight as argued above and leads to a contradiction.

This means that W_i does not contain any other vertices than described above, so for $\frac{1}{2}k < i \leq k$, W_i consists exactly of all the children of $C_{i-\frac{1}{2}k}$ and the roots of C_j for $j > i - \frac{1}{2}k$.
Combining these statements, we conclude that every vertex of the graph will be in one of the maximum weighted independent sets, while the number of root vertices equals $\frac{1}{2}k$ and the number of non-root vertices equals $2 + 4 +, \ldots, +k = \sum_{j=1}^{\frac{1}{2}k}(2j) = 2\sum_{j=1}^{\frac{1}{2}k} j = 2 \cdot (\frac{1}{2} \cdot (\frac{1}{2}k)((\frac{1}{2}k) + 1)) \geq \frac{1}{4}k^2 = \Omega(k^2)$. For this last derivation, we use the formula $\sum_{i=1}^{n} i = \frac{1}{2}n(n+1)$. This means that the for the weighted forest F, we indeed have $|\bigcup_{j=1}^{k} W_j| = \Omega(k^2)$ ■

This lemma shows that using the reduction rules in this paper, we cannot provide an upperbound of $\mathcal{O}(k^2)$ for the number of vertices in the kernel.

5 Further applications of the techniques

In this section we will investigate how we can apply the techniques from the previous sections to the problems Maximum Weighted d-Scattered Set (MWdSS) and Maximum Weighted Induced Connected Subgraph (MWICS). First, we use reduction rules to obtain equivalent instances in which the graph has a bounded treewidth. In this section, we leave out the dynamic programs, because they do not contribute the explanation of the concepts. An example of a dynamic programming algorithm for Maximum Induced Matching is presented in [22]. We expect that this algorithm can be generalized to work for MWdSS with a few adjustments. Finally we expect that we can use this dynamic programming algorithm on graphs with a bounded treewidth to obtain a kernel with $\mathcal{O}(k^3)$. We assume parameter d of MWdSS and the number of vertices $|H|$ in the subgraph of MWICS to be of constant size.

5.1 Maximum weighted d-scattered Set

Recall the definition of MWdSS:

Maximum Weighted d-Scattered Set on planar graphs **Parameter:** k
Input: A planar graph G equipped with a weight function $\omega : V(G) \to \mathbb{N}$, integers k, t and a constant d
Question: Does there exist a vertex vertex set $X \subseteq V(G)$ of size exactly k such that each pair of vertices in X has distance at least d in G and $\omega(X) := \sum_{v \in X} w(v) \geq t$?

Similar to the previous sections, we will present a reduction rule. Then we will bound the treewidth of the graph of the equivalent instance by a function of k. Finally, we will describe how we expect to obtain a kernel with $\mathcal{O}(k^3)$ vertices. **Reduction rule**
We first state the lemma that describes the result of the reduction rule and then we will provide a proof sketch.

Lemma 5.1. There is a polynomial time algorithm that reduces an instance (G, k, t) of MWdSS to an equivalent instance (G', k', t') in which G' contains a distance-$(2d-2)$-dominating set of size at most k.

Similar to the procedure in Section 3.1, for $0 \leq i \leq k$ we will select a heaviest vertex v_i in G_i and then construct G_{i+1} by deleting all vertices within a distance $2d-2$ from v_i in G. This ensures that any pair of chosen vertices $(v_i, v_{i'})$ have distance at least $2d-1$ to each other. This way, selecting any vertex that will be in the solution set X, can only have a distance less than d to at most one chosen vertex v_i. This ensures safeness of the reduction rule, which can be proven similarly to Reduction rule 3.3.

A bounded on the Treewidth
Similar to the procedure in Section 3.2, we use the grid minor theorem to bound the treewidth of the graph obtained after the application of the reduction rule described in Lemma 5.1.

Lemma 5.2. Let G be a planar graph with distance-$(2d-2)$-dominating set S of size at most k. Let H be a largest grid minor of G with branch sets $V_{h_{i,j}}$. Then for any vertex $v \in V(G)$, the collection $\{V_{h_{i,j}} \mid V_{h_{i,j}} \cap d_G^2(v) \neq \emptyset, i, j \in \{2d-1, \ldots, \ell-2d+1\}\}$ has size at most $(2 \cdot (2d-2)+1)^2 = (4d-3)^2$.

The proof is analogous of that of Lemma 3.7.
A direct result is that the largest grid minor has at most $k \cdot (4d-3)^2$ inner branch sets. This implies that the side length of the inner part is bounded by $(4d-3)\sqrt{k}$. This means the total side length is bounded by $(4d-3)\sqrt{k} + 4(d-1)$. We will combine this result with the Grid Minor Theorem 3.9 to obtain the following result:

Corollary 5.3. Let G be a planar graph with distance-$(2d-2)$-dominating set S of size k. Then G has treewidth at most $(18d-13.5)\sqrt{k}+18(d-1))$.

Then, in polynomial time a tree decomposition can be obtained with treewidth at most $\frac{3}{2}\cdot(18d-13.5)\sqrt{k}+18(d-1))=(27d-20.25)\sqrt{k}+27(d-1)$ [18]. Using this tree decomposition of width $\mathcal{O}(d\sqrt{k})$, we present the following conjecture:

Conjecture 5.4. There exists a constant c, depending on d, such that Maximum Weighted d-scattered set can be solved in $O^*(2^{c\sqrt{k}})$ time.

As stated at the start of this section, we can use a dynamic programming algorithm on graphs with bounded treewidth, to solve MWdSS. This algorithm would have a running time of $O^*(2^{c\sqrt{k}})$. This conjecture is supported by Courcelle's Theorem that states that any 'Extended Monadic Second Order Optimization problem', problems in which the goal is to find a smallest vertex subset that satisfies an 'Monadic second order'-formula, can be decided in linear time on graphs of bounded treewidth [23]. That implies that this conjecture can be proven if MWdSS is indeed an extended monadic second-order optimization problem.

Kernelization of Maximum Weighted d-Scattered Set

Similarly to the procedure in Section 4, we obtain a connected distance-$(2d-2)$-dominating set of size at most $4dk$. If we would delete this set S, the remainder of the graph would be $2d-2$-outerplanar. This implies, using Theorem 4.3, that $G-S$ has treewidth at most $3(2d-2)-1=6d-7$. Therefore S is a treewidth-$6d-7$-modulator S. Now we can construct a tree decomposition for each connected component of $G-S$ with the same treewidth in polynomial time, since we assume d to be a constant.

Now Lemma 4.6 can be applied. This will construct a protrusion decomposition with parameters $(\mathcal{O}(d^2k),\mathcal{O}(d^2k),\mathcal{O}(d))$. We will use this in the following conjecture.

Conjecture 5.5. Maximum Weighted d-scattered set admits a kernel with $\mathcal{O}(k^3)$ vertices.

We expect that a kernel with $\mathcal{O}(k^3)$ vertices can be constructed using the same principles as in Section 4.3. This holds under the assumption that a more generalized problem can be solved efficiently on graphs with bounded treewidth. We would still need to find a maximum weighted d-scattered set of size m for each subtree rooted at a neighbor of the root node for all $m\in[k]$. But this time we do not keep the union of all $W_{i,Y,m}$, the maximum weighted independent set of size exactly m in $G[\chi(T_i)\setminus N[Y]$, in which $N[Y]=\bigcup_{y\in Y}N[y]$, but we keep the union of $W_{i,Y_1,\ldots,Y_d,m}$, the maximum weighted d-scattered set of size exactly m in $G[\chi(T_i)\setminus(\bigcup_{j=1}^{d}N_j[Y_j]$, in which $N_j[Y_j]=\bigcup_{y\in Y_j}\{v\in G\mid d(v,y)\leq j\}$ for all $Y_1,\ldots,Y_d\subseteq V(G)$ such that $Y_i\cap Y_j=\emptyset$ if $i\neq j$. Intuitively, for each main subtree T_i we keep the vertices that are in any maximum weighted d-scattered set of G where we delete for all $j\in[d]$ the vertices within a distance j for any vertex in Y_j for all possibilities to partition the vertices in the root bag $\chi(i)$ in the sets $Y_1,\ldots,Y_d,Y_0$.

We have to delete those vertices, because the result of the sub-problems is not only influenced by the inclusion of vertices of the root bag of the subtree, but also by vertices in the root bag of the protrusion decomposition or vertices in other sub-problems. By keeping all vertices that form a solution in the graph after any of these deletions, we account for all possible inclusions for the solution outside of the subtree.

We expect that the union of $\chi(r)$ and all $W_{i,Y_1,\ldots,Y_d,m}$ will result in $\mathcal{O}(k^3)$ vertices in total. If this is the case, we can transform this instance into a kernel using a similar technique as discussed in Lemma 4.12.

5.2 Maximum Weighted Induced Connected Subgraph

Recall the definition of MWICS on planar graphs, where we use $k\cdot H$ to denote the disjoint union of k copies of the graph H. Using this notion, the problem is defined as follows:

Maximum Weighted Induced Connected Subgraph	**Parameter:** k

Input: A planar graph G equipped with a weight function $\omega : V(G) \to \mathbb{N}$, a (finite) connected graph H and integers k, t
Question: Does there exist an vertex set $X \subseteq V(G)$ of size $|V(H)| \cdot k$ such that $G[X]$ is isomorphic to $k \cdot H$ and $\omega(X) := \sum_{v \in X} w(v) \geq t$?

We denote the diameter of the subgraph by the fixed constant sd, that is naturally bounded by the constant $|H|$. We will first describe a reduction rule that can be used to obtain a similar instance on a graph with a distance-$(sd+1)$-dominating set. Then we know the treewidth is bounded, since we have proven that in the previous section. We expect that this problem can then be solved in $\mathcal{O}^*(2^{\sqrt{k}})$ time.
Finally, we will argue that we cannot directly apply the same techniques for kernelization as we have used for other problems in this paper. **Reduction rule**

Lemma 5.6. There is a reduction rule that transforms an instance (G, k, t) of MWdSS to an equivalent instance (G', k', t') in which G' contains a distance-$(sd + 1)$-dominating set of size at most $k \cdot |H|$.

Similarly to the procedure described in earlier sections, we first consider the maximum weighted induced subgraph H in $G_0 = G$. We would then construct G_1 by removing the vertices in that subgraph and every vertex within a distance of $sd + 1$. This way, there cannot be a subgraph in the solution set, that prevents at least 2 subgraphs selected by the reduction rule to be chosen and therefore, the described reduction tule is safe.
This reduction leaves us with a graph that contains a distance-$(sd + 1)$-dominating set of size $|V(H)| \cdot k$. **A bound on the Treewidth**
Using a similar argumentation as in previous sections, we obtain an equivalent instance that has treewidth bounded by $(2sd + 3)\sqrt{k \cdot |H|} + 2sd + 2$, which we assume to be $\mathcal{O}(\sqrt{k})$. This result leads to the following conjecture.

Conjecture 5.7. There exists a constant c such that Maximum Weighted Induced Connected Subgraph can be solved in $O^*(2^{c\sqrt{k}})$ time.

As stated at the start of this section, we can use a dynamic programming algorithm on graphs with bounded treewidth, to solve MWICS with an algorithm that has a running time of $O^*(2^{c\sqrt{k}})$. This is supported by Courcelle's Theorem that states that any 'Extended Monadic Second Order Optimization problem', problems in which the goal is to find a smallest vertex subset that satisfies an 'Monadic second order'-formula, can be decided in linear time on graphs of bounded treewidth [23]. That implies that this conjecture can be proven if MWICS is definable in the monadic second-order logic. **Kernelization of Maximum Weighted Induced Connected Subgraph**
Obtaining a kernel for this problem is not as straightforward as for the other discussed problems. The subgraph might be part of multiple connected components of $G - S$, in which S is the treewidth modulator of linear size. An example is provided in Figure 8. In this graph, the treewidth modulator is the 20-vertex grid in the center. The red vertices denote a solution to Maximum Weighted Induced P_3. However, there is a P_3-subgraph that uses vertices of multiple connected components of $G - S$. This implies that the problem cannot be solved by solving all smaller sub-problems, because they can now depend on each other.

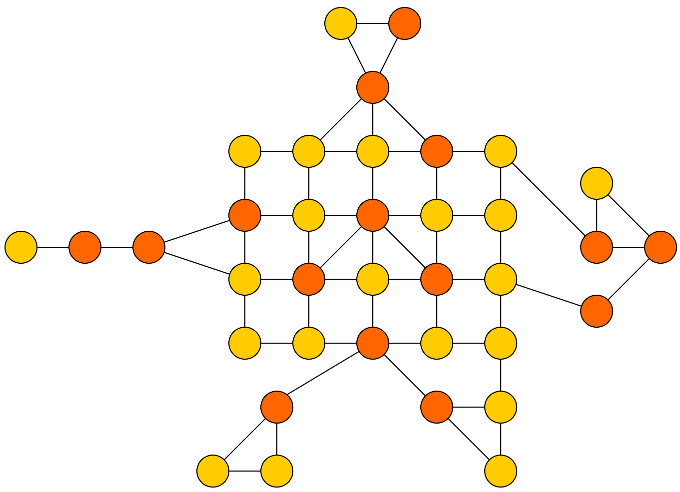


Figure 8: This graph G has 5 orange colored P_3 induced subgraphs. It contains one induced subgraph that stretches out over two different components of $G - S$, where S is the 20-vertex grid in the center.

6 Conclusions and further research

We have found that the problem Maximum Weighted Independent Set for planar graphs is in the complexity class of fixed parameter tractable problems, since in this paper we have developed an algorithm that has a running time of $\mathcal{O}^*(2^{33.75\sqrt{k}})$. Furthermore we have found a kernel for this problem with $\mathcal{O}(k^3)$ vertices.

The obtained kernel in this paper contains $\mathcal{O}(k^3)$ vertices and it is explained at the end of Section 4.3 that it is not easy to improve upon that bound. However, it is still unclear if this bound can be sharpened using other reduction rules or transformations to other problems.
Also, it might be possible to improve upon the running time of the described FPT-algorithm of $2^{\mathcal{O}(\sqrt{k})} \cdot n^c$ for a fixed constant c.

Another possibility to improve upon the kernel size is to reformulate the problem as a more general problem that has a polynomially bounded input size. We can for example reduce Maximum Weighted Independent Set to another variant, where each vertex has both a weight and a size. Then we are interested if there exist a subset of vertices of total size at most k and total weight at least t. In this so-called *compression*, the kernel of size $\Omega(k^2)$ that is described in Lemma 4.13, can be formulated with a linear number of vertices. Further research has to find out if the number of vertices can be further compressed if we reformulate the problem to a more general problem.

We know that the problem is solvable in polynomial time if the input graph has a constant treewidth. The running time might also be improved when we restrict the research to other, more specific graph types. It can also be interesting to consider more general graphs than planar graphs, for example graphs that only don't contain K_5 minors or don't contain $K_{3,3}$ minors.

Finally, this thesis might give an insight for problems of a similar nature. The problem Maximum Weighted d-Scattered Induced Subgraph was briefly mentioned in the introduction. Further research is needed to find out if this generalization admits a polynomial kernel. Of course this is only possible if MWICS also admits a polynomial kernel.
Furthermore, one might wonder if there exists a polynomial kernel for MWICS if we relax the constraint that the subgraph needs to be connected. Also, one might wonder what happens when we do allow the distance d or size of subgraph H to depend on a function of the parameter k for MWdSS respectively MWICS. I don't expect the latter problem to admit a polynomial sized kernel.

7 Acknowledgements

I would like to thank my daily supervisor Bart Jansen for initiating this project and providing very useful help and feedback throughout the duration of the project. I also would like to thank the other members of the graduation committee Frits Spieksma and Judith Keijsper for their time. Finally I would like to thank my father, Tim and my friends Henk, Steven and Sjanne for our stimulating discussions about this thesis.

8 References

[1] Sergiy Butenko and Panagote M Pardalos. *Maximum independent set and related problems, with applications*. PhD thesis, University of Florida, 2003.

[2] Richard M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, Boston, MA, 1972.

[3] Z. Li and G. He. Maximum weighted independent set based rfid reader anti-collision protocol. In *2020 39th Chinese Control Conference (CCC)*, pages 5166–5171, 2020.

[4] Dimitri Kagaris and Spyros Tragoudas. Maximum weighted independent sets on transitive graphs and applications. *Integration*, 27(1):77 – 86, 1999.

[5] Anita Saha and Madhumangal Pal. Maximum weight k-independent set problem on permutation graphs. *International Journal of Computer Mathematics*, 80(12):1477–1487, 2003.

[6] Martin Grohe. Descriptive and parameterized complexity. In Jörg Flum and Mario Rodriguez-Artalejo, editors, *Computer Science Logic*, pages 14–31, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[7] K. Appel and W. Haken. Every planar map is four colorable. Part I: Discharging. *Illinois J. Math.*, 21(3):429–490, 09 1977.

[8] H.L. Bodlaender, F.V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D.M. Thilikos. (Meta) Kernelization. *Journal of the ACM*, 63(5), November 2016.

[9] Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: theory of parameterized preprocessing*. Cambridge University Press, 2019.

[10] Rod G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness ii: On completeness for W[1]. *Theoretical Computer Science*, 141(1):109 – 131, 1995.

[11] Marek Cygan, Fedor V. Fomin, Łukasz Kowalik, Daniel Lokshtanov, Daniel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer International Publishing, 2015.

[12] W. Zheng, J. Gu, P. Peng, and J. X. Yu. Efficient weighted independent set computation over large graphs. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1970–1973, 2020.

[13] Jochen Alber, Michael R. Fellows, and Rolf Niedermeier. Polynomial-time data reduction for dominating set. *J. ACM*, 51(3):363–384, May 2004.

[14] Michael Etscheid, Stefan Kratsch, Matthias Mnich, and Heiko Röglin. Polynomial kernels for weighted problems. *Journal of Computer and System Sciences*, 84:1 – 10, 2017.

[15] Fedor V. Fomin, Daniel Lokshtanov, Saket Saurabh, and Dimitrios M. Thilikos. *Bidimensionality and Kernels*, pages 503–510. 2010.

[16] Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Elsevier*, 209:1–45, 1998.

[17] Yingfei Dong, Ding-Zhu Du, and Oscar Ibarra. *Algorithms and Computation: 20th International Symposium, ISAAC 2009, Honolulu, Hawaii, USA, December 16-18, 2009. Proceedings*, volume 5878. 01 2009.

[18] L. Marchal. *Treewidth: structural properties and algorithmic insights*. PhD thesis, Maastricht University, 2012.

[19] Cormen Thomas H., Leiserson Charles E., Rivest Ronald L., and Stein Clifford. *Introduction to Algorithms, Third Edition.*, volume 3rd ed. The MIT Press, 2009.

[20] Frank Kammer and Torsten Tholey. A lower bound for the treewidth of k-outerplanar graphs. Technical report, Universität Augsburg, 2009.

[21] Hans L. Bodlaender. A linear-time algorithm for finding tree-decompositions of small treewidth. *SIAM Journal on Computing*, 25(6):1305–1317, 1996.

[22] Hannes Moser and Somnath Sikdar. The parameterized complexity of the induced matching problem. *Discrete Applied Mathematics*, 157(4):715 – 727, 2009.

[23] Bruno Courcelle. The monadic second-order logic of graphs. I. Recognizable sets of finite graphs. *Information and Computation*, 85(1):12 – 75, 1990.