

1

**University of Warsaw**

2

Faculty of Mathematics, Informatics and Mechanics

3

**Katarzyna Kowalska**

Student no. 371053

4

# Approximation and Parameterized Algorithms for Segment Set Cover

5

6

Master's thesis

7

in **COMPUTER SCIENCE**

8

Supervisor:

**dr Michał Pilipczuk**

Institute of Informatics

9

June 2022

10 **Supervisor's statement**

11 Hereby I confirm that the presented thesis was prepared under my supervision and  
12 that it fulfils the requirements for the degree of Master of Computer Science.

13 Date

Supervisor's signature

14 **Author's statement**

15 Hereby I declare that the presented thesis was prepared by me and none of its contents  
16 was obtained by means that are against the law.

17 The thesis has never before been a subject of any procedure of obtaining an academic  
18 degree.

19 Moreover, I declare that the present version of the thesis is identical to the attached  
20 electronic version.

21 Date

Author's signature

22

## **Abstract**

23 The work presents a study of different geometric set cover problems. It mostly focuses on  
24 segment set cover and its connection to the polygon set cover.

25

## **Keywords**

26 set cover, geometric set cover, FPT,  $W[1]$ -completeness, APX-completeness, PCP theorem,  
27 NP-completeness

28

## **Thesis domain (Socrates-Erasmus subject area codes)**

29 11.3 Informatyka

30

31

## **Subject classification**

32 D. Software

33 D.127. Blabalgorithms

34 D.127.6. Numerical blabalysis

35

## **Tytuł pracy w języku polskim**

36 Algorytmy aproksymacyjne i parametryzowane dla problemu pokrywania punktów  
37 odcinkami na płaszczyźnie



# Contents

39	<b>1. Introduction</b>	5
40	<b>2. Definitions</b>	7
41	2.1. Geometric set cover	7
42	2.2. Approximation	7
43	2.3. Problem modification with $\delta$ -extension	7
44	<b>3. APX-hardness of geometric set cover problem</b>	9
45	3.1. MAX-(3,3)-SAT and statement of reduction	9
46	3.2. Reduction	11
47	3.2.1. VARIABLE-gadget	11
48	3.2.2. OR-gadget	12
49	3.2.3. CLAUSE-gadget	14
50	3.2.4. Summary	16
51	3.2.5. Summary of construction	17
52	3.3. Construction lemmas and proof of Lemma 3.1	18
53	<b>4. Fixed-parameter tractable algorithm for geometric set cover problem</b>	21
54	4.1. Fixed-parameter tractable algorithm for unweighted segments	21
55	4.1.1. Axis-parallel segments	21
56	4.1.2. Segments in arbitrary directions	22
57	4.2. Fixed-parameter tractable algorithm for weighted segments with $\delta$ -extensions	24
58	<b>5. W[1]-hardness for weighted segments in 3 directions</b>	29
59	<b>6. Geometric Set Cover with lines</b>	39
60	6.1. Lines parallel to one of the axis	39
61	6.2. FPT for arbitrary lines	39
62	6.3. APX-completeness for arbitrary lines	39
63	6.4. 2-approximation for arbitrary lines	40
64	6.5. Connection with general set cover	40
65	<b>7. Geometric Set Cover with polygons</b>	41
66	7.1. State of the art	41
67	<b>8. Conclusions</b>	43



# Chapter 1

## Introduction

1. Set cover is NP-complete
2. Geometric set cover is NP-complete
3. Approximation of geometric set cover
  - (a) with fat polygons with  $\delta$ -extensions admits EPTAS
  - (b) with thin rectangles (segments) is APX-hard (this paper)
  - (c) if we relax it with  $\frac{1}{2}$ -extensions it is still APX-hard (this paper)
4. Geometric set cover parameterized by size of solution
  - (a) unweighted segments admit FPT algorithm (this paper)
  - (b) weighted segments relaxed with  $\delta$ -extensions admit FPT (this paper)
  - (c) weighted segments (in 3 directions) are W[1]-hard (this paper)
    - I personally think that in 2 directions they are also W[1]-hard
  - (d) with squares is W[1]-hard

The Set Cover problem is one of the most common NP-complete problems. [tutaj referencja] We are given a family of sets and have to choose the smallest subfamily of these sets that cover all their elements. This problem naturally extends to settings where we put different weights on the sets and look for the subfamily of the minimal weight. This problem is NP-complete even without weights and if we put restrictions on what the sets can be. One of such variants is Vertex Cover problem, where sets have size 2 (they are edges in a graph).

In this work we focus on another such variant where the sets correspond to some geometric shapes and only some points of the plane have to be covered. When these shapes are rectangles with edges parallel to the axis, the problem can be proven to be W[1]-complete (solution of size  $k$  cannot be found in  $n^{o(k)}$  time), APX-complete (for sufficiently small  $\epsilon > 0$ , the problem does not admit  $1 + \epsilon$ -approximation scheme) [referencje].

Some of these settings are very easy. Set cover with lines parallel to one of the axis can be solved in polynomial time.

There is a notion of  $\delta$ -expansions, which loosen the restrictions on geometric set cover. We allow the objects to cover the points after  $\delta$ -expansion and compare the result to the original setting. This way we can produce both FPT and EPTAS for the rectangle set cover with  $\delta$ -extensions [referencje].

**Our contribution.** In this work, we prove that unweighted geometric set cover with segments is fixed parameter tractable (FPT).

Moreover, we show that geometric set cover with segments is APX-complete for unweighted axis-parallel segments, even with  $1/2$ -extensions. So the problem for very thin rectangles also cannot admit PTAS. Therefore, in the efficient polynomial-time approximation scheme (EPTAS) for *fat polygons* by [Har-Peled and Lee, 2009], the assumption about polygons being fat is necessary.

Finally, we show that geometric set cover with weighted segments in 3 directions is  $W[1]$ -complete. However, geometric set cover with weighted segments is FPT if we allow  $\delta$ -extension.

This result is especially interesting, since it's counter-intuitive that the unweighed setting is FPT and the weighted setting is  $W[1]$ -complete. Most of such problems (like vertex cover or [wiecej przykladow]) are equally hard in both weighted and unweighted settings.



## Chapter 2

## Definitions

In this chapter we present some definitions, that are later used across the different chapters.

### 2.1. Geometric set cover

Everytime we refer to the geometric set cover, we consider a geometric set cover problem on 2 dimensional plane.

In the geometric set cover problem we are given  $\mathcal{P}$  – a set of objects, which are connected subsets of the plane and  $\mathcal{C}$  – a set of points in the plane. The task is to choose  $\mathcal{R} \subseteq \mathcal{P}$  such that every point in  $\mathcal{C}$  is inside some object from  $\mathcal{R}$  and  $|\mathcal{R}|$  is minimized.

In the parameterized setting for a given  $k$ , we only look for a solution  $\mathcal{R}$  such that  $|\mathcal{R}| \leq k$  or decide that there is no such set  $\mathcal{R}$ .

In the weighted setting, there is some given weight function  $f : \mathcal{P} \rightarrow \mathbb{R}^+$ , and we would like to find a solution  $\mathcal{R}$  that minimizes  $\sum_{R \in \mathcal{R}} f(R)$ .

### 2.2. Approximation

Let us recall some definitions related to optimization problems that are used in the following sections.

**Definition 2.1.** A **polynomial-time approximation scheme (PTAS)** for a minimization problem  $\Pi$  is a family of algorithms  $\mathcal{A}_\epsilon$  for every  $\epsilon > 0$  such that  $\mathcal{A}_\epsilon$  takes an instance  $I$  of  $\Pi$  and in polynomial time finds a solution that is within a factor  $(1 + \epsilon)$  of being optimal. That means the reported solution has weight at most  $(1 + \epsilon)opt(I)$ , where  $opt(I)$  is the weight of an optimal solution for  $I$ .

**Definition 2.2.** A problem  $\Pi$  is **APX-hard** if assuming  $P \neq NP$ , there exists  $\epsilon > 0$  such that there is no polynomial-time  $(1 + \epsilon)$ -approximation algorithm for  $\Pi$ .

### 2.3. Problem modification with $\delta$ -extension

Another idea that we present here, but much less versatile than the previous concepts is  $\delta$ -extensions. We define it specifically for the geometric set cover problem.

It is based on the similar idea of  $\delta$ -shrinking for the geometric independent set problem, which is presented in [Pilipczuk et al., 2016].

Intuitively, we consider a problem with slightly larger objects, which makes the instance more permissive. Although we aim to find a solution, that is not larger than the optimum

solution of the original problem, so this is substantially easier than just solving the problem for larger objects. It may even be the case, that we are able to find the solution of size smaller than the optimum solution to the original problem.

First, we formally define  $\delta$ -extended objects.

**Definition 2.3.** For any  $\delta > 0$  and a center-symmetric object  $L$  with centre of symmetry  $S = (x_s, y_s)$ , the  $\delta$ -**extension** of  $L$  is the object  $L^{+\delta} = \{(1 + \delta) \cdot (x - x_s, y - y_s) + (x_s, y_s) : (x, y) \in L\}$ , that is,  $L^{+\delta}$  is the image of  $L$  under homothety centered at  $S$  with scale  $(1 + \delta)$

The geometric set cover problem with  $\delta$ -extensions is a modified version of geometric set cover where:

- We need to cover all the points in  $\mathcal{C}$  with objects from  $\{P^{+\delta} : P \in \mathcal{P}\}$  (which always include no fewer points than the objects before  $\delta$ -extension);
- We look for a solution that is not larger than the optimum solution for the original problem. Note that it does not need to be an optimal solution in the modified problem.

Formally, we have the following.

**Definition 2.4.** The **geometric set cover problem with  $\delta$ -extension** is the problem where for an input instance  $I = (\mathcal{P}, \mathcal{C})$ , the task is to output a solution  $\mathcal{R} \subseteq \mathcal{P}$  such that the  $\delta$ -extended set  $\{R^{+\delta} : R \in \mathcal{R}\}$  covers  $\mathcal{C}$  and is not larger than the optimal solution for the problem without extensions, i.e.  $|\mathcal{R}| \leq |\text{opt}(I)|$ .

At last, we recall definition of the polynomial-time approximation scheme (PTAS) of the problem with  $\delta$ -extension.

**Definition 2.5.** We define a **PTAS for geometric set cover with  $\delta$ -extension** as a family of algorithms  $\{\mathcal{A}_{\delta, \epsilon}\}_{\delta, \epsilon > 0}$  that each takes as an input instance  $I = (\mathcal{P}, \mathcal{C})$ , and in polynomial-time outputs a solution  $\mathcal{R} \subseteq \mathcal{P}$  such that the  $\delta$ -extended set  $\{R^{+\delta} : R \in \mathcal{R}\}$  covers  $\mathcal{C}$  and is within a  $(1 + \epsilon)$  factor of the optimal solution for this problem without extensions, i.e.  $(1 + \epsilon)|\mathcal{R}| \leq |\text{opt}(I)|$ .

## Chapter 3

# APX-hardness of geometric set cover problem

In this section we analyze whether there exists a PTAS for geometric set cover for rectangles. We show that we can restrict this problem to a very simple setting: segments parallel to axes and allow  $(1/2)$ -extension, and the problem is still APX-hard. Note that segments are just degenerated rectangles with one side being very narrow.

Our results can be summarized in the following theorem and this section aims to prove it.

**Theorem 3.1.** *(axis-parallel segment set cover with  $1/2$ -extension is APX-hard). Unweighted geometric set cover with axis-parallel segments in 2D (even with  $1/2$ -extension) is APX-hard. That is, assuming  $P \neq NP$ , there does not exist a PTAS for this problem.*

Theorem 3.1 implies the following.

**Corollary 3.1.** *(rectangle set cover is APX-hard). Unweighted geometric set cover with axis-parallel rectangles (even with  $1/2$ -extension) is APX-hard.*

We prove Theorem 3.1 by taking a problem that is APX-hard and showing a reduction. For this problem we choose MAX-(3,3)-SAT which we define below.

### 3.1. MAX-(3,3)-SAT and statement of reduction

**Definition 3.1.** MAX-3SAT is the following maximization problem. We are given a 3-CNF formula, and need to find an assignment of variables that satisfies the most clauses.

**Definition 3.2.** MAX-(3,3)-SAT is a variant of MAX-3SAT with an additional restriction that every variable appears in exactly 3 clauses and every clause contains exactly 3 literals of 3 different variables. Note that thus, the number of clauses is equal to the number of variables.

In our proof of Theorem 3.1 we use hardness of approximation of MAX-(3,3)-SAT proved in [Håstad, 2001] and described in Theorem 3.2 below.

**Definition 3.3** ( $\alpha$ -satisfiable MAX-3SAT formula). MAX-3SAT formula with  $m$  clauses is at most  $\alpha$ -satisfiable, if every assignment of variables satisfies no more than  $\alpha m$  clauses.

**Theorem 3.2.** [Håstad, 2001] For any  $\epsilon > 0$ , it is NP-hard to distinguish satisfiable (3,3)-SAT formulas from at most  $(7/8 + \epsilon)$ -satisfiable (3,3)-SAT formulas.

Given an instance  $I$  of MAX-(3,3)-SAT, we construct an instance  $J$  of axis-parallel segment set cover problem such that for a sufficiently small  $\epsilon > 0$ , a polynomial time  $(1 + \epsilon)$ -approximation algorithm for  $J$  would be able to distinguish whether an instance  $I$  of MAX-(3,3)-SAT is fully satisfiable or is at most  $(7/8 + \epsilon)$ -satisfiable. However, according to Theorem 3.2 the latter problem is NP-hard. This would imply  $P = NP$ , contradicting the assumption.

The following lemma encapsulates the properties of the reduction described in this section, and it allows us to prove Theorem 3.1.

**Lemma 3.1.** *Given an instance  $S$  of MAX-(3,3)-SAT with  $n$  variables and optimum value  $opt(S)$ , we can construct an instance  $I$  of geometric set cover with axis-parallel segments in 2D such that:*

(1) *For every solution  $X$  of instance  $I$ , there exists a solution of  $S$  that satisfies at least  $15n - |X|$  clauses.*

(2) *For every solution of instance  $S$  that satisfies  $w$  clauses, there exists a solution of  $I$  of size  $15n - w$ .*

(3) *Every solution with 1/2-extensions of  $I$  is also a solution to the original instance  $I$ .*

Therefore, the optimum size of a solution of  $I$  is  $opt(I) = 15n - opt(S)$ .

We prove Lemma 3.1 in subsequent sections, but meanwhile let us prove Theorem 3.1 using Lemma 3.1 and Theorem 3.2.

*Proof of Theorem 3.1.* Consider any  $0 < \epsilon < 1/(15 \cdot 8)$ .

Let us assume that there exists a polynomial-time  $(1 + \epsilon)$ -approximation algorithm for unweighted geometric set cover with axis-parallel segments in 2D with (1/2)-extensions. We construct an algorithm that solves the problem stated in Theorem 3.2, thereby proving that  $P = NP$ .

Take an instance  $S$  of MAX-(3,3)-SAT to be distinguished and construct an instance of geometric set cover  $I$  using Lemma 3.1. We now use the  $(1 + \epsilon)$ -approximation algorithm for geometric set cover on  $I$ . Denote the size of the solution returned by this algorithm as  $approx(I)$ . We prove that if in  $S$  one can satisfy at most  $(\frac{7}{8} + \epsilon)n$  clauses, then  $approx(I) \geq 15n - (\frac{7}{8} + \epsilon)n$  and if  $S$  is satisfiable, then  $approx(I) < 15n - (\frac{7}{8} + \epsilon)n$ .

**Assume  $S$  satisfiable.** From the definition of  $S$  being satisfiable, we have:

$$opt(S) = n.$$

From Lemma 3.1 we have:

$$opt(I) = 14n.$$

Therefore,

$$\begin{aligned} approx(I) &\leq (1 + \epsilon)opt(I) = 14n(1 + \epsilon) = 14n + 14\epsilon \cdot n = \\ &= 14n + (15\epsilon - \epsilon)n < 14n + \left(\frac{1}{8} - \epsilon\right)n = 15n - \left(\frac{7}{8} + \epsilon\right)n. \end{aligned}$$

**Assume  $S$  is at most  $(\frac{7}{8} + \epsilon)$  satisfiable.** From the definition of  $S$  being at most  $(\frac{7}{8} + \epsilon)$  satisfiable, we have:

$$opt(S) \leq \left(\frac{7}{8} + \epsilon\right)n$$

From Lemma 3.1 we have:

$$opt(I) \geq 15n - \left(\frac{7}{8} + \epsilon\right)n$$

224 Since a solution to  $I$  with  $\frac{1}{2}$ -extension is also a solution without any extension, by Lemma  
225 3.1 (3), we have:

$$approx(I) \geq opt(I) = 15n - \left(\frac{7}{8} + \epsilon\right)n$$

226 Therefore, by using the assumed  $(1 + \epsilon)$ -approximation algorithm, it is possible to distin-  
227 guish the case when  $S$  is satisfiable: from the case when it is at most  $(\frac{7}{8} + \epsilon)n$  satisfiable,  
228 it suffices to compare  $approx(I)$  with  $15n - (\frac{7}{8} + \epsilon)n$ . Hence, the assumed approximation  
229 algorithm cannot exist, unless  $P = NP$ .  $\square$

## 230 3.2. Reduction

231 We proceed to the proof of Lemma 3.1. That is, we show a reduction from the MAX-(3,3)-SAT  
232 problem to geometric set cover with segments parallel to axis. Moreover, the obtained instance  
233 of geometric set cover will be robust to 1/2-extensions (have the same optimal solution after  
234 1/2-extension).

235 The construction will be composed of 2 types of gadgets: **VARIABLE-gadgets** and  
236 **CLAUSE-gadgets**. **CLAUSE-gadgets** will be constructed using two **OR-gadgets** connected  
237 together.

### 238 3.2.1. VARIABLE-gadget

239 **VARIABLE-gadget** is responsible for choosing the value of a variable in a CNF formula. It  
240 allows two minimum solutions of size 3 each. These two choices correspond to the two Boolean  
241 values of the variable corresponding to this gadget.

242 **Points.** Define points  $a, b, c, d, e, f, g, h$  as follows, where  $L = 22n$ :



Figure 3.1: **VARIABLE-gadget**. We denote the set of points marked with black circles as  $\text{pointsVariable}_i$ , and they need to be covered (are part of the set  $\mathcal{C}$ ). Note that some of the points are not marked as black dots and exists only to name segments for further reference. We denote the set of red segments as  $\text{chooseVariable}_i^{\text{false}}$  and the set of blue segments as  $\text{chooseVariable}_i^{\text{true}}$ .

243

$$\begin{array}{llll} a = (-3L, 0) & b = (-2L, 0) & c = (-L, 0) & d = (-3L, 1) \\ e = (-2L, 1) & f = (-2L, 2) & g = (L, 0) & h = (L, 2) \end{array}$$

Let us define:

$$\text{pointsVariable} = \{a, b, c, d, e, f\}$$

and, for any  $1 \leq i \leq n$ ,

$$\text{pointsVariable}_i = \text{pointsVariable} + (0, 4i).$$

244 We denote  $a_i := a + (0, 4i)$  etc.

245 **Segments.** Let us define:

$$\text{chooseVariable}_i^{\text{true}} := \{(a_i, d_i), (b_i, f_i), (c_i, g_i)\},$$

$$\text{chooseVariable}_i^{\text{false}} := \{(a_i, c_i), (d_i, e_i), (f_i, h_i)\},$$

$$\text{segmentsVariable}_i := \text{chooseVariable}_i^{\text{true}} \cup \text{chooseVariable}_i^{\text{false}}.$$

246 We also name two of these segment for future reference:  $\text{xTrueSegment}_i := (c_i, g_i)$ ,  
247  $\text{xFalseSegment}_i := (f_i, h_i)$ .

248 **Lemma 3.2.** *For any  $1 \leq i \leq n$ , points in  $\text{pointsVariable}_i$  can be covered using 3 segments*  
249 *from  $\text{segmentsVariable}_i$ .*

250 *Proof.* We can use either set  $\text{chooseVariable}_i^{\text{true}}$  or  $\text{chooseVariable}_i^{\text{false}}$ . □

251 **Lemma 3.3.** *For any  $1 \leq i \leq n$ , points in  $\text{pointsVariable}_i$  can not be covered with fewer than*  
252 *3 segments from  $\text{segmentsVariable}_i$ .*

253 *Proof.* No segment of  $\text{segmentsVariable}_i$  covers more than one point from  $\{d_i, f_i, c_i\}$ , therefore  
254  $\text{pointsVariable}_i$  can not be covered with fewer than 3 segments. □

255 **Lemma 3.4.** *For every set  $A \subseteq \text{segmentsVariable}_i$  such that  $A$  covers  $\text{pointsVariable}_i$  and*  
256  *$\text{xTrueSegment}_i, \text{xFalseSegment}_i \in A$ , it holds that  $|A| \geq 4$ .*

257 *Proof.* No segment from  $\text{segmentsVariable}_i$  covers more than one point from  $\{a_i, e_i\}$ , therefore  
258  $\text{pointsVariable}_i - \{c_i, f_i, g_i, h_i\}$  can not be covered with fewer than 2 segments. □

### 259 3.2.2. OR-gadget

260 OR-gadget connects input and output segments (see Figure 3.2) in a way that is supposed to  
261 simulate a binary *or* function.

262 Input segments are the only segments that cover points outside of the gadget, as their left  
263 ends lie outside of it. Point  $v_{i,j}$  is the only one that can be covered by segments that do not  
264 belong to the gadget.

265 The OR-gadget has the property that every set of segments that covers all the points in  
266 the gadget uses at least 3 segments from it.. Moreover, the output segment belongs to the  
267 solution of size 3 only if at least one of the input segments belong to the solution. Therefore,  
268 optimum solutions restricted to the OR-gadget behave like a binary *or* function for the input  
269 segments.



Figure 3.2: **OR-gadget**. Segments from  $\text{chooseOr}_{i,j}^{\text{false}}$  are **red**, segments from  $\text{chooseOr}_{i,j}^{\text{true}}$  are blue (both **light blue** and **dark blue**), segments from  $\text{orMoveVariable}_{i,j}$  are **green** and **yellow**. **Dark blue** segment is the *output* segment. Grey segments  $\text{input}_x$  and  $\text{input}_y$  are input segments that are not part of  $\text{segmentsOr}_{i,j}$ .

270 **Points.**

$$\begin{aligned}
 l_0 &:= (0, 0) & m_0 &:= (0, 1) & n_0 &:= (0, 2) & o_0 &:= (0, 3) \\
 p_0 &:= (0, 4) & q_0 &:= (1, 1) & r_0 &:= (1, 3) & s_0 &:= (2, 1) \\
 t_0 &:= (2, 2) & u_0 &:= (2, 3) & v_0 &:= (3, 2)
 \end{aligned}$$

$$\text{vec}_{i,j} := (20i + 3 + 3j, 4(n + 1) + 2j)$$

272 For integers  $i, j$ , define  $\{l_{i,j}, m_{i,j} \dots v_{i,j}\}$  as  $\{l_0, m_0 \dots v_0\}$  shifted by  $\text{vec}_{i,j}$ , i.e.  $l_{i,j} =$   
 273  $l_0 + \text{vec}_{i,j}$  etc.

274 Note that  $v_{i,0} = l_{i,1}$  (see Figure 3.3)

$$\text{pointsOr}_{i,j} := \{l_{i,j}, m_{i,j}, n_{i,j}, o_{i,j}, p_{i,j}, q_{i,j}, r_{i,j}, s_{i,j}, t_{i,j}, u_{i,j}\}$$

275 Note that  $\text{pointsOr}_{i,j}$  does not include the point  $v_{i,j}$

276 **Segments.** We define set of segments in several parts:

$$\begin{aligned}
 \text{chooseOr}_{i,j}^{\text{false}} &:= \{(q_{i,j}, r_{i,j}), (s_{i,j}, u_{i,j})\}, \\
 \text{chooseOr}_{i,j}^{\text{true}} &:= \{(m_{i,j}, s_{i,j}), (o_{i,j}, u_{i,j}), (t_{i,j}, v_{i,j})\},
 \end{aligned}$$

$$\text{orMoveVariable}_{i,j} := \{(l_{i,j}, n_{i,j}), (n_{i,j}, p_{i,j})\}.$$

277 Finally all segments in OR-gadget are defined as:

$$\text{segmentsOr}_{i,j} := \text{chooseOr}_{i,j}^{\text{false}} \cup \text{chooseOr}_{i,j}^{\text{true}} \cup \text{orMoveVariable}_{i,j}$$

278 **Lemma 3.5.** For any  $1 \leq i \leq n, j \in \{0, 1\}$  and  $x \in \{l_{i,j}, p_{i,j}\}$ , points in  $\text{pointsOr}_{i,j} - \{x\} \cup$   
 279  $\{v_{i,j}\}$  can be covered with 4 segments from  $\text{segmentsOr}_{i,j}$ .

280 *Proof.* We can do that using one segment from  $\text{orMoveVariable}_{i,j}$ , the one that does not cover  
 281  $x$ , and all segments from  $\text{chooseOr}_{i,j}^{\text{true}}$ .  $\square$

282 **Lemma 3.6.** For any  $1 \leq i \leq n, j \in \{0,1\}$ , points in  $\text{pointsOr}_{i,j}$  can be covered with 4  
 283 segments from  $\text{segmentsOr}_{i,j}$ .

284 *Proof.* We can do that using segments from  $\text{orMoveVariable}_{i,j} \cup \text{chooseOr}_{i,j}^{\text{false}}$ .  $\square$

### 285 3.2.3. CLAUSE-gadget

286 A CLAUSE-gadget is responsible for determining whether variable values assigned in variable  
 287 gadgets satisfy the corresponding clause in the input formula  $\phi$ . It has a minimum solution  
 288 of weight  $w$  if and only if the clause is satisfied, i.e. at least one of the respective variables is  
 289 assigned the correct value. Otherwise, its minimum solution has weight  $w + 1$ . In this way,  
 290 by analyzing the cost of the minimum solution for the entire constructed instance, we will be  
 291 able to tell how many clauses it was possible to satisfy in the optimum solution of  $\phi$ .



Figure 3.3: **CLAUSE-gadget for a clause  $a \vee b \vee \neg c$ .** Every green rectangle is an OR-gadget.  $y$ -coordinates of  $x_{i,0}$ ,  $y_{i,0}$  and  $z_{i,0}$  depend on the variables in the  $i$ -th clause. Grey segments corresponds to the values of variables satisfying the  $i$ -th clause.

292 **Points.** First, we define auxiliary functions for literals. For a literal  $w$ , let  $\text{idx}(w)$  be the  
 293 index of the variable in  $w$ , and  $\text{neg}(w)$  be the Boolean value whether the variable is negated  
 294 in  $w$  or not.

295 Let us assume that clause  $C_i = a \vee b \vee c$  for any literals  $a, b, c$ . Then, we define points in  
 296 the gadget as:



$$\begin{aligned}
x_{i,0} &:= (20i, 4 \cdot \text{id}x(a) + 2 \cdot \text{neg}(c)), & x_{i,1} &:= (20i, 4(n+1)), \\
y_{i,0} &:= (20i+1, 4 \cdot \text{id}x(b) + 2 \cdot \text{neg}(b)), & y_{i,1} &:= (20i+1, 4(n+1)+4), \\
z_{i,0} &:= (20i+2, 4 \cdot \text{id}x(c) + 2 \cdot \text{neg}(c)), & z_{i,1} &:= (20i+2, 4(n+1)+6).
\end{aligned}$$

We are now ready to define set of points:

$$\text{moveVariable}_i := \{x_{i,j} : j \in \{0,1\}\} \cup \{y_{i,j} : j \in \{0,1\}\} \cup \{z_{i,j} : j \in \{0,1\}\},$$

$$\text{pointsClause}_i := \text{moveVariable}_i \cup \text{pointsOr}_{i,0} \cup \text{pointsOr}_{i,1} \cup \{v_{i,1}\}.$$

Note that these two points are equal:  $v_{i,0} = l_{i,1}$ . This translates to the fact, that output of the one OR-gadget is an input to the other OR-gadget to create *or* of 3 segments.

**Segments.** We also define segments for the clause gadget as below:

$$\begin{aligned}
\text{segmentsClause}_i &:= \{(x_{i,0}, x_{i,1}), (y_{i,0}, y_{i,1}), (z_{i,0}, z_{i,1}), (x_{i,1}, l_{i,0}), (y_{i,1}, p_{i,0}), (z_{i,1}, p_{i,1}), \} \cup \\
&\cup \text{segmentsOr}_{i,0} \cup \text{segmentsOr}_{i,1}.
\end{aligned}$$

The CLAUSE-gadgets consist of two OR-gadgets. Ideally, we would place the  $i$ -th CLAUSE-gadget close to the  $\text{xTrueSegment}_{j_1}$  or  $\text{xFalseSegment}_{j_1}$  segments corresponding to the literals that occur in the  $i$ -th clause. It would be inconvenient to position them there, because between these segments there may be additional  $\text{xTrueSegment}_{j_2}$  or  $\text{xFalseSegment}_{j_2}$  segments corresponding to the other literals.

Instead, we use simple auxiliary gadgets to *transfer* whether the segment is in a solution, i.e. segments  $(x_{i,0}, x_{i,1}), (y_{i,0}, y_{i,1}), (z_{i,0}, z_{i,1})$  in this gadget. Each gadget consists of two segments  $(x_{i,0}, x_{i,1}), (x_{i,1}, a)$ . These are the only segments that can cover  $x_{i,1}$ . We place  $x_{i,0}$  on a segment that we want to transfer (i.e. segment responsible for choosing the variable value satisfying the corresponding literal). If in some solution  $x_{i,0}$  is already covered by this segment, then we can cover  $x_{i,1}$  by  $(x_{i,1}, a)$ , thus also covering  $a$ . If  $x_{i,0}$  is not covered by this segment, then the only way to cover  $x_{i,0}$  is to use segment  $(x_{i,0}, x_{i,1})$ . Intuitively, in any optimal solution the two segments *transfer* the state of whether  $x_{i,0}$  is covered onto whether  $a$  is covered. Therefore, the number of segments in the optimal solution is increased by one, and we get a point  $a$  that was effectively placed on some segment  $s$ , but it can be placed anywhere on the plane instead, consequently simplifying the construction.

**Lemma 3.7.** *For any  $1 \leq i \leq n$  and  $a \in \{x_{i,0}, y_{i,0}, z_{i,0}\}$ , there is a set  $\text{solClause}_i^{\text{true},a} \subseteq \text{segmentsClause}_i$  with  $|\text{solClause}_i^{\text{true},a}| = 11$  that covers all points in  $\text{pointsClause}_i - \{a\}$ .*

*Proof.* For  $a = x_{i,0}$  (analogous proof for  $y_{i,0}$ ): First we use Lemma 3.5 twice with excluded  $x = l_{i,0}$  and  $x = l_{i,1} = v_{i,0}$ , resulting with 8 segments in  $\text{chooseOr}_{i,0}^{\text{true}} \cup \text{chooseOr}_{i,1}^{\text{true}}$  which cover all required points apart from  $x_{i,1}, y_{i,0}, y_{i,1}, z_{i,0}, z_{i,1}, l_{i,0}$ . We cover those using additional 3 segments:  $\{(x_{i,1}, l_{i,0}), (y_{i,0}, y_{i,1}), (z_{i,0}, z_{i,1})\}$

For  $a = z_{i,0}$ : Using Lemma 3.6 and Lemma 3.5 with  $x = p_{i,1}$ , we obtain 8 segments in  $\text{chooseOr}_{i,0}^{\text{false}} \cup \text{chooseOr}_{i,1}^{\text{true}}$  which cover all required points apart from  $x_{i,0}, x_{i,1}, y_{i,0}, y_{i,1}, z_{i,1}, p_{i,1}$ . We cover those using additional 3 segments:  $\{(x_{i,0}, x_{i,1}), (y_{i,0}, y_{i,1}), (z_{i,1}, p_{i,1})\}$ .  $\square$

**Lemma 3.8.** *For any  $1 \leq i \leq n$  there is a set  $\text{solClause}_i^{\text{false}} \subseteq \text{segmentsClause}_i$  with  $|\text{solClause}_i^{\text{false}}| = 12$  that covers all points in  $\text{pointsClause}_i$ .*

329 *Proof.* Using Lemma 3.6 twice we can cover  $\text{pointsOr}_{i,0}$  and  $\text{pointsOr}_{i,1}$  with 8 segments. To  
 330 cover the remaining points we additionally use:  $\{(x_{i,0}, x_{i,1}), (y_{i,0}, y_{i,1}), (z_{i,0}, z_{i,1}), (t_{i,1}, v_{i,1})\}$   
 331  $\square$

332 **Lemma 3.9.** *For any  $1 \leq i \leq n$ :*

- 333 (1) *points in  $\text{pointsClause}_i$  can not be covered using any subset of segments from  $\text{segmentsClause}_i$*   
 334 *of size smaller than 12;*
- 335 (2) *points in  $\text{pointsClause}_i - \{x_{i,0}, y_{i,0}, z_{i,0}\}$  can not be covered using any subset of segments*  
 336 *from  $\text{segmentsClause}_i$  of size smaller than 11.*

*Proof of (1).* No segment in  $\text{segmentsClause}_i$  covers more than 1 point from

$$\{x_{i,0}, y_{i,0}, z_{i,0}, l_{i,0}, p_{i,0}, q_{i,0}, u_{i,0}, v_{i,0} = l_{i,1}, p_{i,1}, q_{i,1}, u_{i,1}, v_{i,1}\}.$$

337 Therefore we need to use at least 12 segments.  $\square$

338 *Proof of (2).* We can define disjoint sets  $X, Y, Z$  such that  $X \cup Y \cup Z \subseteq \text{pointsClause}_i -$   
 339  $\{x_{i,0}, y_{i,0}, z_{i,0}\}$  such that there are no segments in  $\text{segmentsClause}_i$  covering points from dif-  
 340 ferent sets. And we prove a lower bound for each of these sets. First, let:

$$X := \{x_{i,1}, y_{i,1}, z_{i,1}\}.$$

341 No two points in  $X$  can be covered with one segment of  $\text{segmentsClause}_i$ , so it must be  
 342 covered with 3 different segments. Next we define other sets:

$$Y := \text{pointsOr}_{i,0} - \{l_{i,0}, p_{i,0}\},$$

$$Z := \text{pointsOr}_{i,1} - \{l_{i,1}, p_{i,1}\}.$$

343 For both  $Y$  and  $Z$  we can check all of the subsets of 3 segments of  $\text{segmentsClause}_i$  to  
 344 conclude that none of them cover the considered, so both  $Y$  and  $Z$  have to be covered with  
 345 disjoint sets of 4 segments each.

346 Therefore,  $\text{pointsClause}_i - \{x_{i,0}, y_{i,0}, z_{i,0}\}$  must be covered with at least  $3 + 4 + 4 = 11$   
 347 segments from  $\text{segmentsClause}_i$ .  $\square$

### 348 3.2.4. Summary

349 Add some smart lemmas that sets will be exclusive to each other.

350 **Lemma 3.10. Robustness to 1/2-extensions.** *For every segment  $s \in \mathcal{P}$ ,  $s$  and  $s^{+\frac{1}{2}}$  cover*  
 351 *the same points from  $\mathcal{C}$ .*

352 *Proof.* We can just check every segment. Most of the segments  $s$  are collinear only with points  
 353 that lay on  $s$ , so trivially  $s^{+\frac{1}{2}}$  cannot cover more points than  $s$  does.

354 Within VARIABLE-gadget for any  $1 \leq i \leq n$  after  $\frac{1}{2}$ -extension:  $(c_i, g_i)$  does not cover  $b_i$ .

355 Within OR-gadget some of the segments are collinear and share one point; specifically, for  
 356 any  $1 \leq i \leq n$  and  $j \in \{0, 1\}$ , after  $\frac{1}{2}$ -extension:

- 357 •  $(l_{i,j}, n_{i,j})$  does not cover  $o_{i,j}$ ,
- 358 •  $(n_{i,j}, p_{i,j})$  does not cover  $m_{i,j}$ ,
- 359 •  $(t_{i,j}, v_{i,j})$  does not cover  $n_{i,j}$ .



Figure 3.4: **Schema of the whole construction.**

General layout of VARIABLE-gadgets and CLAUSE-gadgets and how they interact with each other.

360 Within CLAUSE-gadget, for any  $1 \leq i \leq n$  after  $\frac{1}{2}$ -extension:

- 361 •  $(o_{i,0}, u_{i,0})$  does not cover  $m_{i,1}$ ,
- 362 •  $(m_{i,1}, s_{i,1})$  does not cover  $u_{i,0}$ ,
- 363 •  $(y_{i,1}, p_{i,0})$  does not cover  $n_{i,1}$ .

364 For two consecutive VARIABLE-gadgets, for any  $1 \leq i < n$  after  $\frac{1}{2}$ -extension:  $(b_i, f_i)$  does  
 365 not cover  $b_{i+1}$  (nor  $f_{i-1}$  for  $i > 1$ ). Similarly  $(a_i, d_i)$  does not cover  $a_{i+1}$  (nor  $d_{i-1}$  for  $i > 1$ ),  
 366 because this segment is shorter than the previous one and  $a_i$  and  $b_i$  share y-coordinate.

367 For two consecutive CLAUSE-gadgets, segments from one do not cover anything from the  
 368 other, as the gadgets have width 9 and every leftmost x-coordinate is divisible by 20. Hence  
 369 two different gadgets do not interact with each other after  $\frac{1}{2}$ -extension.

370 Next we need to check whether VARIABLE-gadget's segments do not cover any points  
 371  $x_{i,0}, y_{i,0}$  or  $z_{i,0}$  from CLAUSE-gadget. For any  $1 \leq i \leq n$  and  $1 \leq j \leq n$ , all points  $x_{j,0}, y_{j,0}$   
 372 and  $z_{j,0}$  have x-coordinate strictly positive. Segment  $(a_i, c_i)$  have length  $2L$  and  $c_i$  has x-  
 373 coordinate equal to  $-L$ , so after  $\frac{1}{2}$ -extension this segment does not cover any points with a  
 374 positive x-coordinate.

375 □

### 376 3.2.5. Summary of construction

Finally we define set of points and segments for the constructed instance:

$$\mathcal{C} := \bigcup_{1 \leq i \leq n} \text{pointsVariable}_i \cup \text{pointsClause}_i,$$

$$\mathcal{P} := \bigcup_{1 \leq i \leq n} \text{segmentsVariable}_i \cup \text{segmentsClause}_i.$$



Figure 3.5: **Schema of the whole construction.**

General layout of VARIABLE-gadgets and CLAUSE-gadgets and how they interact with each other.

### 3.3. Construction lemmas and proof of Lemma 3.1

In order to prove Lemma 3.1 we introduce several auxiliary lemmas proving properties of the construction described in the previous section.

Consider an instance  $S$  of MAX-(3,3)-SAT of size  $n$  with optimum solution satisfying  $k$  clauses. Let us construct an instance  $(\mathcal{C}, \mathcal{P})$  of geometric set cover as described in Section 3.2 for the instance  $S$  of MAX-(3,3)-SAT.

**Lemma 3.11.** *Instance  $(\mathcal{C}, \mathcal{P})$  of geometric set cover admits a solution of size  $15n - k$ .*

*Proof.* Let the clauses in  $S$  be  $c_1, c_2 \dots c_n$  and the variables be  $x_1, x_2 \dots x_n$ . Let the variable assignment in the optimum solution to  $S$  be  $\phi : \{x_1, x_2 \dots x_n\} \rightarrow \{\mathbf{true}, \mathbf{false}\}$ .

We cover every VARIABLE-gadget with solution described in Lemma 3.2, where in the  $i$ -th gadget we choose the set of segments corresponding to the value of  $\phi(x_i)$ .

For every clause that is satisfied, say  $c_i$ , let us name the variable that is **true** in it as  $x_i$  and point corresponding to  $x_i$  in  $\mathbf{pointsClause}_i$  as  $a$ . Points in  $\mathbf{pointsClause}_i$  are covered with set  $\mathbf{solClause}_i^{\mathbf{true}, a}$  described in Lemma 3.7. For every clause that is not satisfied, say  $c_j$ , points in  $\mathbf{pointsClause}_j$  are covered with set  $\mathbf{solClause}_j^{\mathbf{false}}$  described in Lemma 3.8.

Formally we define sets responsible for choosing variable assignment and satisfying clauses,  $R_i$  and  $C_i$  respectively, as following:

$$\begin{aligned}
R_i &:= \begin{cases} \text{chooseVariable}_i^{\text{true}} & \text{if } \phi(x_i) = \text{true} \\ \text{chooseVariable}_i^{\text{false}} & \text{if } \phi(x_i) = \text{false} \end{cases} \\
C_i &:= \begin{cases} \text{solClause}_i^{\text{true},a} & \text{if } c_i \text{ satisfied by literal corresponding to point } a \\ \text{solClause}_i^{\text{false}} & \text{if } c_i \text{ not satisfied} \end{cases} \\
\mathcal{R} &:= \bigcup_{i=1}^n \{R_i \cup C_i : 1 \leq i \leq n\}.
\end{aligned}$$

394 This set covers all the points from  $\mathcal{C}$ , because the sets  $R_i$ ,  $C_i$  individually cover their  
395 corresponding gadgets, as proved in the respective lemmas.

396 All of these sets are disjoint, so the size of the obtained solution is:

$$|\mathcal{R}| = \sum_{i=1}^n R_i + \sum_{i=1}^n C_i = 3n + 11k + 12(n - k) = 15n - k. \quad \square$$

397 **Lemma 3.12.** *Suppose we have a solution  $\mathcal{R}$  of the instance  $(\mathcal{C}, \mathcal{P})$  of geometric set cover.*  
398 *Then there exists a solution  $\mathcal{R}'$ , such that  $|\mathcal{R}'| \leq |\mathcal{R}|$ , and  $\mathcal{R}'$  contains at most one of the*  
399 *segments  $\text{xTrueSegment}_i$  and  $\text{xFalseSegment}_i$  from each VARIABLE-gadget.*

400 *Proof.* Assume that we have  $\{\text{xTrueSegment}_i, \text{xFalseSegment}_i\} \subseteq \mathcal{R}$  for some  $i$ . We will show  
401 how to modify  $\mathcal{R}$  into  $\mathcal{R}'$ , such that the number of such  $i$  decreases, while  $\mathcal{R}'$  is still a valid  
402 solution of  $(\mathcal{C}, \mathcal{P})$ , and  $|\mathcal{R}'| \leq |\mathcal{R}|$ . Then, by repeating this procedure, we can eventually  
403 construct a solution satisfying the property from the Lemma.

404 To construct  $\mathcal{R}'$ , we first remove from  $\mathcal{R}$  all segments belonging to  $\text{segmentsVariable}_i$ .  
405 Recall that the  $i$ -th VARIABLE-gadget corresponds to variable  $x_i$  in  $S$ . As every variable in  
406  $S$  is used in exactly 3 clauses, then one literal  $x_i$  or  $\neg x_i$  must appear in at least 2 clauses. If  
407 that literal is  $x_i$ , then we add to the constructed solution all segments from  $\text{chooseVariable}_i^{\text{true}}$ ,  
408 otherwise we add all segments from  $\text{chooseVariable}_i^{\text{false}}$ .

409 Now, there exists at most one CLAUSE-gadget which needs adjustment to make  $\mathcal{R}'$  valid;  
410 assuming it is the  $j$ -th clause, then one of the points  $x_{j,0}, y_{j,0}$  or  $z_{j,0}$  for this CLAUSE-gadget  
411 might be not covered, say  $y_{j,0}$ . We amend the solution by adding  $(y_{j,0}, y_{j,1})$  to  $\mathcal{R}'$ .

412 By Lemma 3.4 we know that  $\mathcal{R}$  used at least 4 segments from  $\text{segmentsVariable}_i$ . Therefore,  
413 we removed at least 4 segments and added at most 4 segments, so  $|\mathcal{R}'| \leq |\mathcal{R}|$ .  $\square$

414 **Lemma 3.13.** *Suppose we have a solution  $\mathcal{R}$  of the instance  $(\mathcal{C}, \mathcal{P})$  of geometric set cover*  
415 *that is of size  $w$ . Then there exists a solution of  $S$  that satisfies at least  $15n - w$  clauses.*

416 *Proof.* Let the clauses in  $S$  be  $c_1, c_2 \dots c_n$  and the variables be  $x_1, x_2 \dots x_n$ . Given a solution  
417  $\mathcal{R}$  of the instance  $(\mathcal{C}, \mathcal{P})$  of geometric set cover, we use Lemma 3.12 to modify  $\mathcal{R}$  such that  
418 for any  $i$  it contains at most one of  $\text{xTrueSegment}_i$  and  $\text{xFalseSegment}_i$ ; this may decrease the  
419 cost of  $\mathcal{R}$ , but that does not matter in the subsequent construction. To simplify notation, in  
420 the remainder of this proof we use  $\mathcal{R}$  to refer to the modified solution.

Given  $\mathcal{R}$ , we construct a solution of  $S$  by defining an assignment of variables:

$$\phi : \{x_1, x_2 \dots x_n\} \rightarrow \{\text{true}, \text{false}\}$$

421 that satisfies at least  $15n - w$  clauses in  $S$ .

**Definition of  $\phi$ .** Recall that due to Lemma 3.12,  $\mathcal{R}$  contains at most one of  $\text{xTrueSegment}_i$  and  $\text{xFalseSegment}_i$ .

We define the value  $\phi(x_i)$  for the variable  $x_i$  as follows:

$$\begin{cases} \phi(x_i) = \text{true} & \text{if } \text{xTrueSegment}_i \in \mathcal{R} \\ \phi(x_i) = \text{false} & \text{otherwise} \end{cases}$$

Moreover, from Lemma 3.3 we get  $|\text{segmentsVariable}_i \cap \mathcal{R}| \geq 3$  for every  $i$ .

**Clauses satisfied with the chosen variable assignment.** For a clause  $c_i$ ,  $\mathcal{R}$  needs to use at least 11 segments to cover  $\text{pointsClause}_i - \{x_{i,0}, y_{i,0}, z_{i,0}\}$  in the  $i$ -th CLAUSE-gadget (Lemma 3.9).

Moreover, if none of the points  $\{x_{i,0}, y_{i,0}, z_{i,0}\}$  are covered by the segments from  $\mathcal{R} \cap \text{segmentsVariable}_i$ , then  $\mathcal{R}$  needs to cover  $\text{pointsClause}_i$  with at least 12 segments by Lemma 3.9.

Let us denote  $a$  as the amount of such clauses  $c_i$  for which none of the points  $x_{i,0}, y_{i,0}, z_{i,0}$  in  $\text{pointsClause}_i$  were covered by segments from  $\mathcal{R} \cap \text{segmentsVariable}_j$  for any  $1 \leq j \leq n$ .

Consider a clause  $c_i$  for which at least one of the points  $x_{i,0}, y_{i,0}, z_{i,0}$  in  $\text{pointsClause}_i$  were covered by segments from  $\mathcal{R} \cap \text{segmentsVariable}_j$  for some  $1 \leq j \leq n$ , then denote this point as  $t$  and say it corresponds to literal  $q$  and variable  $x_j$ . Point  $t$  can be only covered in  $\text{segmentsVariable}_j$  by a corresponding segment  $\text{xTrueSegment}_j$  or  $\text{xFalseSegment}_j$  (depending on whether the literal  $q$  is negated or not). From the definition of  $\phi$  and the fact that one of this segment is in  $\mathcal{R}$ , we know that  $\phi(j)$  has the value that evaluates  $w$  to be true. Therefore, clause  $c_i$  is satisfied.

Consequently,  $\phi$  satisfies all but at most  $a$  clauses in  $S$ .

To conclude, given a solution of  $(\mathcal{C}, \mathcal{P})$  of size  $w$  we constructed a variable assignment  $\phi$  that satisfies at least  $n - a$  clauses of  $S$ . Finally, note that

$$w \geq 3n + 11(n - a) + 12a = 3n + 11n + a = 14n + a,$$

hence

$$15n - w \leq 15n - 14n - a = n - a.$$

Therefore  $\phi$  satisfies at least  $15n - w$  clauses of  $S$ . □

We are ready to conclude the proof of Lemma 3.1.

*Proof of Lemma 3.1.* By Lemma 3.11, we know that there exists a solution to  $(\mathcal{C}, \mathcal{P})$  of size  $15n - k$ , so:

$$\text{opt}((\mathcal{C}, \mathcal{P})) \leq 15n - k.$$

Since the optimum solution of  $S$  satisfies  $k$  clauses, then according to Lemma 3.13:

$$\text{opt}((\mathcal{C}, \mathcal{P})) \geq 15n - k.$$

Therefore, the solution given by Lemma 3.11 of size  $15n - k$  is an optimum solution to the instance  $(\mathcal{C}, \mathcal{P})$ . □

## Chapter 4

# Fixed-parameter tractable algorithm for geometric set cover problem

In this chapter we show fixed-parameter tractable algorithms for the geometric set cover problem in two different settings. Section 4.1 shows a fixed-parameter tractable algorithm for geometric set cover with unweighted segments. The remainder of the chapter presents a fixed-parameter tractable algorithm for geometric set cover with weighted segments with  $\delta$ -extensions. We show an algorithm for the setting with  $\delta$ -extensions, because the original problem with weights is W[1]-hard, as we show in Chapter 5.

We start with a shared definition for this problem. We define *extreme points* for a set of collinear points.

**Definition 4.1.** For a set of collinear points  $C$  in the plane, **extreme points** of  $C$  are the endpoints of the smallest segment that covers all points from set  $C$ .

If  $C$  consists of one point or is empty, then there are 1 or 0 extreme points respectively.

### 4.1. Fixed-parameter tractable algorithm for unweighted segments

In this section we consider fixed-parameter tractable algorithms for unweighted geometric set cover with segments. The setting where segments are required to be axis-parallel (or limited to a constant number of directions) has an FPT algorithm already present in literature in the Parametrized Algorithms book [Cygan et al., 2015]. We present an FPT algorithm for geometric set cover with unweighted segments, where segments are in arbitrary directions.

#### 4.1.1. Axis-parallel segments

**Theorem 4.1.** (*FPT for segment cover with axis-parallel segments*). *There exists an algorithm that given a family  $\mathcal{P}$  of axis-parallel segments, a set of points  $\mathcal{C}$  and a parameter  $k$ , runs in time  $O(2^k)$ , and outputs a solution  $\mathcal{R} \subseteq \mathcal{P}$  such that  $|\mathcal{R}| \leq k$  and  $\mathcal{R}$  covers all points in  $\mathcal{C}$ , or determines that such a set  $\mathcal{R}$  does not exist.*

We present here a simple algorithm from [Cygan et al., 2015] for completeness.

We show an  $O(2^k)$ -time branching algorithm. In each step, the algorithm selects a point  $a$  which is not yet covered, branches to choose one of the two directions, and greedily chooses a segment  $a$  in that direction to cover. This proceeds until either all points are covered or  $k$  segments are chosen.

Let us take the point  $a = (x_a, y_a)$  which is the smallest among points that are not yet covered in the lexicographic ordering of points in  $\mathbb{R}^2$ . We need to cover  $a$  with some of the remaining segments.

Branch over the choice of one of the coordinates ( $x$  or  $y$ ); without loss of generality, let us assume we chose  $x$ . Among the segments lying on line  $x = x_a$ , we greedily add to the solution the one that covers the most points. As  $a$  was the smallest in the lexicographical order, all points on the line  $x = x_a$  have the  $y$ -coordinate larger than  $y_a$ . Therefore, if we denote the greedily chosen segment as  $s$ , then any other segment on the line  $x = x_a$  that covers  $a$  can only cover a (possibly strict) subset of points covered by  $s$ . Thus, greedily choosing  $s$  is optimal.

In each step of the algorithm we add one segment to the solution, thus the recursion can be stopped at depth  $k$ . If no branch finds a solution, then this means that a solution of size at most  $k$  does not exist.

Note that, the same algorithm can be used for segments in  $d$  directions, where we branch over  $d$  choices of directions, and it runs in complexity  $\mathcal{O}(d^k)$ .

#### 4.1.2. Segments in arbitrary directions

In this section we consider the setting where segments are not constrained to a constant number of directions. We present a fixed-parameter tractable algorithm, parameterized by the size of the solution.

**Theorem 4.2. (FPT for segment cover).** *There exists an algorithm that given a family  $\mathcal{P}$  of segments (in any direction), a set of points  $\mathcal{C}$  and a parameter  $k$ , runs in time  $k^{O(k)} \cdot (|\mathcal{C}| \cdot |\mathcal{P}|)^2$ , and outputs a solution  $\mathcal{R} \subseteq \mathcal{P}$  such that  $|\mathcal{R}| \leq k$  and  $\mathcal{R}$  covers all points in  $\mathcal{C}$ , or determines that such a set  $\mathcal{R}$  does not exist.*

We will need the following lemmas proving properties of any instance of the problem.

**Lemma 4.1.** *Given an instance  $(\mathcal{P}, \mathcal{C})$  of the segment cover problem, without a loss of generality we can assume that no segment covers a superset of what another segment covers. That is, for any distinct  $A, B \in \mathcal{P}$ , we have  $A \cap \mathcal{C} \not\subseteq B \cap \mathcal{C}$  and  $A \cap \mathcal{C} \not\supseteq B \cap \mathcal{C}$ .*

*Proof.* If there are two distinct subsets of  $\mathcal{P}$ ,  $A, B$ , such that  $A \cap \mathcal{C} \subseteq B \cap \mathcal{C}$ .

We construct a set  $\mathcal{P}' := \mathcal{P} - \{A\}$ . We prove that for any solution  $\mathcal{R}$  of  $(\mathcal{P}, \mathcal{C})$ , we can construct a solution  $\mathcal{R}' \subseteq \mathcal{P}'$ , such that  $|\mathcal{R}'| \leq |\mathcal{R}|$ . Let us take any solution  $\mathcal{R}$  of  $(\mathcal{P}, \mathcal{C})$ . If  $A \in \mathcal{R}$ , then  $\mathcal{R}' := \mathcal{R} \cup \{B\} - \{A\}$ , otherwise  $\mathcal{R}' := \mathcal{R}$ . Let us consider the case when  $A \in \mathcal{R}$ , because the other case is trivial. Since  $A \cap \mathcal{C} \subseteq B \cap \mathcal{C}$ , then  $\mathcal{R} \cup \{B\} - \{A\}$  covers the same points from  $\mathcal{C}$  as  $\mathcal{R}$ . Also  $|\mathcal{R} \cup \{B\} - \{A\}| \leq |\mathcal{R}|$ .  $\square$

**Lemma 4.2.** *Given an instance  $(\mathcal{P}, \mathcal{C})$  of the segment cover problem transformed by Lemma 4.1, if there exists a line  $L$  with at least  $k + 1$  points on it, then there exists a subset  $A \subseteq \mathcal{P}$ , of size at most  $k$ , such that every solution  $\mathcal{R}$  with  $|\mathcal{R}| \leq k$  satisfies  $|A \cap \mathcal{R}| \geq 1$ . Moreover, such a subset can be found in polynomial time.*

*Proof.* Let us enumerate the points from  $\mathcal{C}$  that lie on  $L$  as  $x_1, x_2, \dots, x_t$  in the order in which they appear on  $L$ . Our proposed set is defined as:

$$A := \{\text{segment that have the leftmost point in } x : x \in x_1, x_2, \dots, x_k\}.$$

If such segment does not exist for any of these points, then set  $A$  has smaller size. We prove the lemma by contradiction. Let us assume that there exists a solution  $\mathcal{R}$  of size at most  $k$ , such that  $\mathcal{R} \cap A = \emptyset$ .



Every segment that is not collinear with  $L$  can cover at most one of the points that lie on this line. Hence if all segments from  $\mathcal{R}$  were not collinear with  $L$ , then  $\mathcal{R}$  would cover at most  $k$  points on line  $L$ , but  $L$  had at least  $k + 1$  different points from  $\mathcal{C}$  on it.

Therefore we know that one of the segments from  $\mathcal{R}$  must be collinear with  $L$  and at most  $k - 1$  segments can be not collinear with  $L$ . Segments from  $\mathcal{R}$ , that are not collinear with  $L$  can cover at most  $k - 1$  points among  $\{x_1, x_2, \dots, x_k\}$ , therefore at least one of these points must be covered by segments from  $\mathcal{R}$ . We take leftmost point from  $\{x_1, x_2, \dots, x_k\}$  that is covered in  $\mathcal{R}$  by segment collinear with  $L$  and name it  $a$ . After transformation from Lemma 4.1 there is only one segment that starts in  $a$ , therefore this segment must be in both  $\mathcal{R}$  and  $A$ .

This contradiction concludes the proof that  $|A \cap \mathcal{R}| \geq 1$  for any solution  $\mathcal{R}$  of size at most  $k$ .  $\square$

We are now ready to prove Theorem 4.2.

*Proof of Theorem 4.2.*

We will prove this theorem by presenting a branching algorithm that works in desired complexity. It branches over the choice of segments to cover the lines with *a lot* of points, then solves a small instance (where every line has at most  $k$  points) by checking all possible solutions.

**Algorithm.** First we use Lemma 4.1.

Next, we present a recursive algorithm. Given an instance of the problem:

- (1) If there exist a line with at least  $k + 1$  points from  $\mathcal{C}$ , we branch over choice of adding to the solution one of the at most  $k$  possible segments provided by Lemma 4.2; name this segment  $s$  and name set of points from  $\mathcal{C}$  that lie on  $s$  as  $S$ . Then we find a solution  $\mathcal{R}$  for the instance  $(\mathcal{C} - S, \mathcal{P} - \{s\})$ , and parameter  $k - 1$ . We return  $\mathcal{R} \cup \{s\}$ .
- (2) If every line has at most  $k$  points on it and  $|\mathcal{C}| > k^2$ , then answer NO.
- (3) If  $|\mathcal{C}| \leq k^2$ , solve the problem by brute force: check all subsets of  $\mathcal{P}$  of size at most  $k$ .

**Correctness.** Lemma 4.2 proves that at least one segment that we branch over in (1) must be present in every solution  $\mathcal{R}$  with  $|\mathcal{R}| \leq k$ . Therefore, the recursive call can find a solution, provided there exists one.

In (2) the answer is no, because every line covers no more than  $k$  points from  $\mathcal{C}$ , which implies the same about every segment from  $\mathcal{P}$ . Under this assumption we can cover only  $k^2$  points with a solution of size  $k$ , which is less than  $|\mathcal{C}|$ .

Checking all possible solutions in (3) is trivially correct.

**Complexity.** In the leaves of the recursion we have  $|\mathcal{C}| \leq k^2$ , so  $|\mathcal{P}| \leq k^4$ , because every segment can be uniquely identified by the two extreme points it covers (by Lemma 4.1). Therefore, there are  $\binom{k^4}{k}$  possible solutions to check, each can be checked in time  $O(k|\mathcal{C}|)$ . Thus, (3) takes time  $k^{O(k)}$ .

In this branching algorithm our parameter  $k$  is decreased with every recursive call, so we have at most  $k$  levels of recursion with branching over  $k$  possibilities. Candidates to branch over can be found on each level in time  $O((|\mathcal{C}| \cdot |\mathcal{P}|)^2)$ .

Reduction from Lemma 4.1 can be implemented in time  $O(|\mathcal{C}|^2|\mathcal{P}|)$ .

It follows that the overall complexity is  $O((|\mathcal{C}| \cdot |\mathcal{P}|)^2 \cdot k^{O(k)})$   $\square$

## 4.2. Fixed-parameter tractable algorithm for weighted segments with $\delta$ -extensions

In this section we consider a geometric set cover problem for weighted segments relaxed with  $\delta$ -extensions. We show that this problem admits an FPT algorithm when parameterized with the size of the solution and  $\delta$ . In the next chapter we show that the assumption about the problem being relaxed with  $\delta$ -extensions is necessary: we prove that geometric set cover problem for weighted segments is W[1]-hard, which means there does not exist an FPT algorithm parameterized by solution size for it.

**Theorem 4.3** (FPT for weighted segment cover with  $\delta$ -extensions). *There exists an algorithm that given a family  $\mathcal{P}$  of  $n$  weighted segments (in any direction), a set of  $m$  points  $\mathcal{C}$ , and parameters  $k$  and  $\delta > 0$ , runs in time  $f(k, \delta) \cdot (nm)^c$  for some computable function  $f$  and a constant  $c$ , and outputs a set  $\mathcal{R} \subseteq \mathcal{P}$  such that  $|\mathcal{R}| \leq k$  and  $\mathcal{R}^{+\delta}$  covers all points in  $\mathcal{C}$  and weight of  $\mathcal{R}$  is not greater than weight of optimum solution of size at most  $k$  for this problem without  $\delta$ -extensions, or determines that such a set  $\mathcal{R}$  does not exist.*

To solve this problem we will introduce a lemma about choosing a *dense* subset of points. Dense subset of points for a set of collinear points  $C$  and parameters  $k$  and  $\delta$  is a subset of  $C$ , such that if we cover it with at most  $k$  segments, these segments after  $\delta$ -extensions will cover all of the points from  $C$ .

We will prove that such set of size bounded by some function  $f(k, \delta)$  always exists. In later part of the section Lemma 4.3 will allow us to find a kernel for our original problem.

**Definition 4.2.** For a set of collinear points  $C$ , a subset  $A \subseteq C$  is  $(k, \delta)$ -**dense** if for any set of segments  $R$  that covers  $A$  and such that  $|R| \leq k$ , it holds that  $R^{+\delta}$  covers  $C$ .

**Lemma 4.3.** *For any set of collinear points  $C$ ,  $\delta > 0$  and  $k \geq 1$ , there exists a  $(k, \delta)$ -dense set  $A \subseteq C$  of size at most  $(2 + \frac{2}{\delta})^k$ . Moreover, there exists an algorithm that computes the  $(k, \delta)$ -dense set in time  $O(|C| \cdot (2 + \frac{2}{\delta}))$ .*

*Proof of Lemma 4.3.* We prove this for a fixed  $\delta$  by induction over  $k$ .

**Inductive hypothesis.** For any set of collinear points  $C$ , there exists a set  $A$  such that:

- $A$  is subset of  $C$ ,
- $A$  is  $(\ell, \delta)$ -dense for every  $1 \leq \ell \leq k$ ,
- $|A| \leq (2 + \frac{2}{\delta})^k$ ,
- extreme points from  $C$  are in  $A$ .

**Base case for  $k = 1$ .** It is sufficient that  $A$  consists of extreme points of  $C$ .

If they are covered with one segment, it must be a segment that includes the extreme points from  $C$ , so it covers the whole set  $C$ .

There are at most 2 extreme points in  $C$  and  $2 < 2 + \frac{2}{\delta}$ .

594 **Inductive step.** Assuming inductive hypothesis for any set of collinear points  $C$  and  
 595 for parameter  $k$ , we will prove it for  $k + 1$ .

596 Let  $s$  be the minimal segment that includes all points from  $C$ . That is, the extreme points  
 597 of  $C$  are endpoints of  $s$ .

598 We define  $M = \lceil 1 + \frac{2}{\delta} \rceil$  subsegments of  $s$  by splitting  $s$  into  $M$  closed segments of equal  
 599 length. We name these segments  $v_i$  and  $|v_i| = \frac{|s|}{M}$  for each  $1 \leq i \leq M$ .

600 Let  $C_i$  be the subset of  $C$  consisting of points laying on  $v_i$ .

601 Let  $t_i$  be the segment with endpoints being the extreme points of  $C_i$ . It might be a  
 602 degenerate segment if  $C_i$  consists of one point or empty if  $C_i$  is empty.

603 You can see an example of such segments  $v_i$  and  $t_i$  in Figure 4.1.

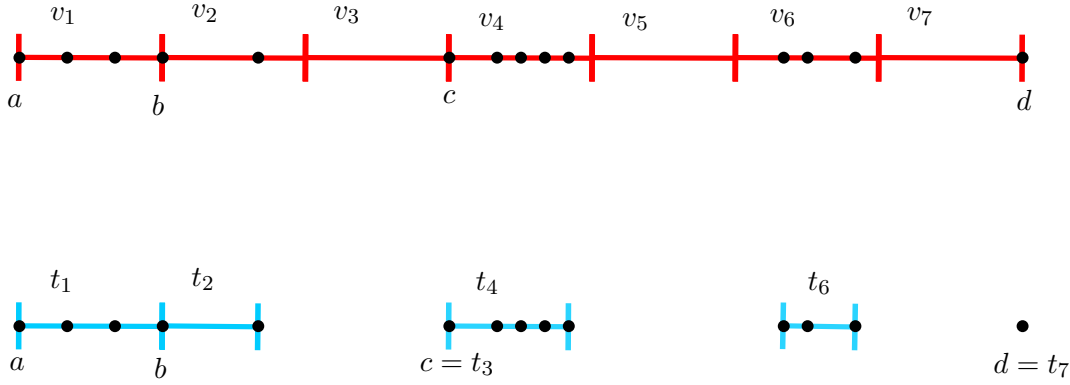


Figure 4.1: **Example of segments  $v_i$  and  $t_i$ .**

Example for  $M = 7$  and some set of points (marked with black circles). Upper picture shows segments  $v_i$  and lower picture shows segments  $t_i$  on the same set of points.  $a$  and  $b$  are extreme points and therefore segment  $s$  ends in  $a$  and  $b$ . Red segments denote split into  $M$  segments of equal length  $v_i$ . Blue segments denote segments  $t_i$ .  $t_5$  is an empty segment, because there are no points that lie on segment  $v_5$ . Segments  $t_3$  and  $t_7$  are degenerated to one point –  $c$  and  $d$  respectively. Segments  $t_1$  and  $t_2$  share one point  $b$ .

604 We use the inductive hypothesis to choose  $(k, \delta)$ -dense sets  $A_i$  for sets  $C_i$ . Note that if  
 605  $|C_i| \leq 1$ , then  $A_i = C_i$  and it is still a  $(k, \delta)$ -dense set for  $C_i$ .

606 Then we define  $A = \bigcup_{i=1}^M A_i$ . Thus  $A$  includes the extreme points of  $C$ , because they are  
 607 included in the sets  $A_1$  and  $A_M$ .

Size of each  $A_i$  is at most  $(2 + \frac{2}{\delta})^k$  from the inductive hypothesis, therefore size of  $A$  is at most:

$$M \left(2 + \frac{2}{\delta}\right)^k = \left\lceil 1 + \frac{2}{\delta} \right\rceil \cdot \left(2 + \frac{2}{\delta}\right)^k \leq \left(2 + \frac{2}{\delta}\right)^{k+1}.$$

608 **Proof that  $A$  is  $(k, \delta)$ -dense for  $C$ .** Let us take any cover of  $A$  with  $k + 1$  segments  
 609 and call it  $\mathcal{R}$ .

610 For every segment  $t_i$ , if there exists a segment  $x$  in  $\mathcal{R}$  that is disjoint with  $t_i$ , then we have  
 611 a cover of  $A_i$  with at most  $k$  segments using  $\mathcal{R} - \{x\}$ . Since  $A_i$  is  $(k, \delta)$ -dense for  $t_i$  and  $C_i$ ,  
 612 then  $(\mathcal{R} - \{x\})^{+\delta}$  covers  $C_i$ . So  $\mathcal{R}^{+\delta}$  covers  $C_i$  as well.

613 If there exists a segment  $t_i$  for which a segment  $x$  as defined above does not exist, then all  
 614  $k + 1$  segments that cover  $A_i$  intersect with  $t_i$ . You can see an example of such segments in  
 615 Figure 4.2. Note that there may exist only one such segment  $t_i$ . From the inductive hypothesis

616 endpoints of  $s$  are in  $A_1$  and  $A_M$  respectively, so  $\mathcal{R}$  must cover them. For each endpoint of  $s$ ,  
 617 there exists a segment that starts in this endpoint and ends somewhere in  $t_i$ . Let us call these  
 618 two segments  $y$  and  $z$ . It follows that:  $|y| + |z| + |t_i| \geq |s|$ . Since  $|t_i| \leq |v_i| = \frac{|s|}{M} \leq \frac{|s|}{1+\frac{2}{\delta}} = \frac{|s|\delta}{\delta+2}$ ,  
 619 we have  $\max(|y|, |z|) \geq |s|(1 - \frac{\delta}{\delta+2})/2 = \frac{|s|}{\delta+2}$ .

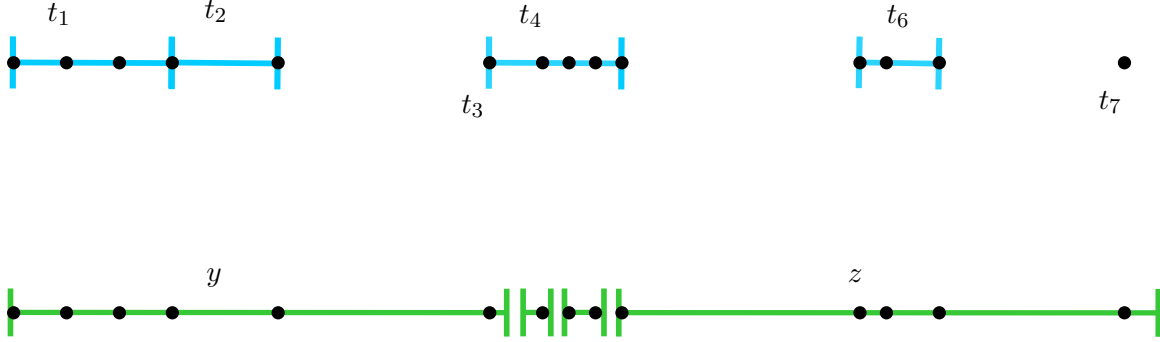


Figure 4.2: **Example of all  $k+1$  segments intersecting with one segment  $t_i$ .** Both pictures show the same set  $\mathcal{C}$  (black circles), the same as in Figure 4.1. The upper picture shows blue segments  $t_i$  for  $M = 7$ . The lower picture shows green segments – solution  $\mathcal{R}$  of size 4. All segments from  $\mathcal{R}$  intersect with  $t_4$ . Segments  $z$  and  $y$  are named on the picture.

After  $\delta$ -extension, the longer of these segments will expand at both ends by at least:

$$\max(|y|, |z|)\delta \geq \frac{|s|\delta}{\delta+2} = \frac{|s|}{1+\frac{2}{\delta}} \geq \frac{|s|}{M} = v_i \geq t_i.$$

620 Therefore, the longer of segments  $y$  and  $z$  will cover the whole segment  $t_i$  after  $\delta$ -extension.  
 621 We conclude that  $\mathcal{R}^{+\delta}$  covers  $C_i$ .  
 622 Since  $C = \bigcup_{i=1}^M C_i$ , it follows that  $\mathcal{R}^{+\delta}$  covers  $C$ .

**Algorithm.** We can simulate the inductive proof by a recursive algorithm with the following complexity:

$$O\left(|C| + \frac{1}{\delta}\right) + O\left(k(2 + \frac{1}{\delta})^k\right).$$

623

□

624 Let us now formulate some claims about the properties for the problem parameterized  
 625 by the solution size. These properties provide bounds for different objects in the problem  
 626 instance, that help us to find small kernel of the problem or claim that the minimal solution  
 627 of this instance must be above some threshold.

628 **Definition 4.3.** A line in  $\mathbb{R}$  is **long** if there are at least  $k+1$  points from  $\mathcal{C}$  on it.

629 **Claim 4.1.** *If there are more than  $k$  different long lines, then  $\mathcal{C}$  can not be covered with  $k$*   
 630 *segments.*

631 *Proof.* We prove the claim by contradiction. Let us assume that we have at least  $k+1$  different  
 632 long lines in our instance of the problem and solution  $\mathcal{R}$  of size at most  $k$  covering points  $\mathcal{C}$ .

633 Choose any long line  $L$ . Every segment from  $\mathcal{R}$ , which is not collinear with  $L$ , covers at  
 634 most one point that lies on  $L$ .

635  $L$  is long, so there are at least  $k + 1$  points from  $\mathcal{C}$  that lie on  $L$ . That implies that there  
 636 must be a segment in  $\mathcal{R}$  that is collinear with  $L$ .

637 Since we have at least  $k + 1$  different long lines, then there are at least  $k + 1$  segments in  
 638  $\mathcal{R}$  collinear with different lines. It contradicts with the assumption that  $|\mathcal{R}| \leq k$ .  $\square$

639 **Claim 4.2.** *If there are more than  $k^2$  points from  $\mathcal{C}$  that do not lie on any long line, then  $\mathcal{C}$*   
 640 *can not be covered with  $k$  segments.*

641 *Proof.* We prove the claim by contradiction. Let us assume that we have at least  $k^2 + 1$  points  
 642 from  $\mathcal{C}$  that do not lie on any long line, call this set  $A$ , and a solution  $\mathcal{R}$  of size at most  $k$   
 643 covering points  $\mathcal{C}$ .

644 For every segment  $s$  from  $\mathcal{R}$  it covers at most  $k$  points from  $A$ . It holds because if  $s$   
 645 covered at least  $k + 1$  points from  $A$ , then the line in the direction of  $s$  would be a long line  
 646 and that contradicts of definition of  $A$ .

647 If every segment from  $\mathcal{R}$  covers at most  $k$  points from  $A$  and  $|\mathcal{R}| \leq k$ , then at most  $k^2$   
 648 points from  $A$  are covered by  $\mathcal{R}$  and that contradicts the fact that  $\mathcal{R}$  is a solution of the given  
 649 geometric set cover instance.  $\square$

650 We are now ready to give a proof of Theorem 4.3.

651 *Proof of Theorem 4.3.* Applying the claims 4.1 and 4.2, if we have more than  $k$  different long  
 652 lines or more than  $k^2$  points from  $\mathcal{C}$  that do not lie on any long line, then we answer that  
 653 there is no solution of size at most  $k$ .

654 Otherwise, we can split  $\mathcal{C}$  into at most  $k + 1$  sets:

- 655 •  $D$ , at most  $k^2$  points that do not lie on any long line;
- 656 •  $C_i$ , points that lie on the  $i$ -th long line.

657 Sets  $C_i$  do not need to be disjoint.

658 Then for every set  $C_i$  we can use Lemma 4.3 to obtain a  $(k, \delta)$ -dense set  $A_i$  for  $C_i$  with  
 659  $|A_i| \leq (2 + \frac{2}{\delta})^k$ .

660 Then we have a set  $\mathcal{C}' = D \cup (\bigcup A_i)$  of size at most  $k^2 + k(2 + \frac{2}{\delta})^k$ . Observe that if we  
 661 have a solution  $\mathcal{R}$  of size at most  $k$  that covers  $\mathcal{C}'$ , then  $\mathcal{R}^{+\delta}$  covers  $\mathcal{C}$ .

662  $\mathcal{C}$  is separated into several parts – sets  $D$  and  $C_i$ . Points from  $D$  are covered by  $\mathcal{R}$ , because  
 663  $D$  is part of  $\mathcal{C}'$ . Each  $A_i$  is covered, because  $A_i$  is part of  $\mathcal{C}'$ ;  $A_i$  is a  $(k, \delta)$ -dense set for  $C_i$ ,  
 664 therefore  $\mathcal{R}^{+\delta}$  covers  $C_i$ .

665 After that we shrunk down  $\mathcal{C}$  to  $\mathcal{C}'$  of size  $f(k, \delta)$  for some computable function  $f$ . Then  
 666 we would like to shrink down  $\mathcal{P}$  to some set of relevant segments of bounded size as well.

667 For every pair of points  $\mathcal{C}'$ , we can choose one segment from  $\mathcal{P}$  that have the lowest weight  
 668 among segments that cover these points or decide there is no segment that cover them. Call  
 669 this set  $\mathcal{P}'$  and name these segments **interesting**. There are at most  $|\mathcal{C}'|^2$  different segments  
 670 in  $\mathcal{P}'$ .

671 We need to show that when we cover  $\mathcal{C}'$  with segments from  $\mathcal{P}'$  we achieve the same  
 672 minimal solution as when we cover them with segments from  $\mathcal{P}$ . In order to prove this,  
 673 consider a minimal solution  $\mathcal{R}$  that covers  $\mathcal{C}'$  with segments from  $\mathcal{P}'$  and take any segment  $s$   
 674 from  $\mathcal{R}$ . Let us look at the points from  $\mathcal{C}'$  that lie on  $s$  and call this set of points  $F$ .  $F$  is  
 675 a set of collinear points for course. We can cover  $F$  with any segment that covers extreme  
 676 points of  $F$ , because all other points lay on the segment between these points. Therefore we  
 677 can change  $s$  to an interesting segment  $s'$  and interesting segments are defined in such a way,  
 678 that  $s'$  has weight no larger than weight of  $s$ .

679 This has complexity  $O(|\mathcal{C}'|^2|\mathcal{P}|)$  and produces shrunk down set of segments  $\mathcal{P}'$  of size  
680  $f(k, \delta)$  for some computable function  $f$ .

681 Then we can iterate over all subsets of  $\mathcal{P}'$  and choose the set with the lowest sum of  
682 weights that cover  $\mathcal{C}'$ . This solution would have weight not larger than optimal solution for  
683 the problem without extension, because we iterate over all possibilities of covering the subset  
684 of  $\mathcal{C}'$ . □

## Chapter 5

# W[1]-hardness for weighted segments in 3 directions

In this chapter we consider geometric set cover problem with weighted segments. Theorem 5.1 proves that this problem is W[1]-hard when parameterized by the size of the solution. We additionally restrict the problem to only use segments in three directions to achieve a stronger result. W[1]-hardness is proved by reduction to a grid tiling problem, which was introduced in [Marx, 2007].

**Definition 5.1.** Line is **right-diagonal** if it is described by linear function  $y = -x + d$  for any  $d \in \mathbb{R}$ . Segment is **right-diagonal** if its direction is a right-diagonal line.

**Theorem 5.1.** *Consider the problem of covering a set  $\mathcal{C}$  of points by selecting at most  $k$  segments from a set of segments  $\mathcal{P}$  with non-negative weights  $w : \mathcal{P} \rightarrow \mathbb{R}^+$  so that the weight of the cover is minimal. Then this problem is W[1]-hard when parameterized by  $\sqrt{k}$  and assuming ETH, there is no algorithm for this problem with running time  $f(k) \cdot (|\mathcal{C}| + |\mathcal{P}|)^{o(\sqrt{k})}$  for any computable function  $f$ . Moreover, this holds even if all segments in  $\mathcal{P}$  are axis-parallel or right-diagonal.*

Theorem 5.1 is also true for less restricted problem where segments have any direction. In order to prove Theorem 5.1 we will show reduction from a W[1]-hard problem. We introduce the grid tiling problem, which is proven to be W[1]-complete in literature.

**Definition 5.2.** We define **powerset** of a set  $A$ , denoted as  $\text{Pow}(A)$ , as the set of all subsets of  $A$ , ie.  $\text{Pow}(A) = \{B : B \subseteq A\}$ .

**Definition 5.3.** In the **grid tiling** problem we are given integers  $n$  and  $k$ , and a function  $f : \{1 \dots k\} \times \{1 \dots k\} \rightarrow \text{Pow}(\{1 \dots n\} \times \{1 \dots n\})$  specifying the set of allowed tiles for each cell of a  $k \times k$  grid. The task is to find functions  $x, y : \{1 \dots k\} \rightarrow \{1 \dots n\}$  that assign colors from  $\{1 \dots n\}$  to respectively columns and rows of the grid, so that  $(x(i), y(j)) \in f(i, j)$  for all valid  $i$  and  $j$ , or conclude that such an assignment does not exist.

In short, in grid tiling problem you need to assign numbers to rows and columns in such a way, that for every pair of a row and a column, the pair of colors assigned to the row and column belongs to the allowed set tiles for this pair. The next theorem describes the complexity of this problem, which is W[1]-hard when parameterized by the size of the grid.

**Theorem 5.2.** [Marx, 2007] *Grid tiling is W[1]-hard when parameterized by  $k$  and assuming ETH, there is no  $f(k) \cdot n^{o(\sqrt{k})}$ -time algorithm solving the grid tiling problem for any computable function  $f$ .*

	$x(1) = 3$	$x(2) = 1$	$x(3) = 3$	$x(4) = 7$
$y(4) = 1$	$(\mathbf{2}, \mathbf{1}); (2, 2);$ $(\mathbf{3}, \mathbf{1}); (3, 9)$	$(1, 1); (3, 1)$	$(\mathbf{3}, \mathbf{1}); (7, 2)$	$(\mathbf{2}, \mathbf{1}); (\mathbf{7}, \mathbf{1})$
$y(3) = 1$	$(\mathbf{2}, \mathbf{1}); (\mathbf{3}, \mathbf{1});$ $(4, 2); (8, 2)$	$(1, 1); (1, 3)$	$(\mathbf{3}, \mathbf{1}); (4, 3)$	$(\mathbf{2}, \mathbf{2}); (\mathbf{7}, \mathbf{1})$
$y(2) = 6$	$(\mathbf{2}, \mathbf{6}); (\mathbf{3}, \mathbf{6})$	$(1, 2); (\mathbf{1}, \mathbf{6});$ $(2, 6)$	$(2, 6); (\mathbf{3}, \mathbf{6})$	$(\mathbf{2}, \mathbf{6}); (\mathbf{7}, \mathbf{6})$
$y(1) = 4$	$(\mathbf{2}, \mathbf{4}); (2, 6);$ $(\mathbf{3}, \mathbf{4}); (\mathbf{3}, \mathbf{9})$	$(1, 4); (\mathbf{1}, \mathbf{9})$	$(\mathbf{3}, \mathbf{4}); (\mathbf{3}, \mathbf{9})$	$(\mathbf{2}, \mathbf{9}); (\mathbf{7}, \mathbf{4})$

Figure 5.1: **Example of a grid tiling instance with its solution.**

In the first row and column of the table you can see the solution: functions  $x$  and  $y$ . The tiles used in this solution are marked in **bold**. If we instead chose the tiles marked in blue (whenever there is one, taking the tile marked in **bold** otherwise), then that corresponds to setting  $x(1) = 2$ , and would also form a correct solution. On the other hand, if we instead chose the tiles marked in red (as before), then that corresponds to setting  $y(1) = 9$  and  $x(4) = 2$  and that would **not** form a correct solution. Even though the first row is correct, tile with coordinates  $(3, 4)$  requires tile  $(2, 1)$ , not  $(2, 2)$ .

The reminder of this section is proving Theorem 5.1 by reduction of a grid tiling problem instance to a geometric set cover instance. That proves the  $W[1]$ -hardness of the geometric set cover problem, because if we could solve it with an FPT algorithm, then we could also solve the grid tiling problem (which we reduced to the geometric set cover). Therefore geometric set cover with setting described in Theorem 5.1 is at least as hard as the grid tiling problem.

**Construction.** We start with an instance of the grid tiling problem  $(n, k, f)$ . The instance consists of:

- size of the grid  $k$ ,
- number of colors  $n$ ,
- function of allowed tiles  $f : \{1, \dots, k\} \times \{1, \dots, k\} \rightarrow \text{Pow}(\{1, \dots, n\} \times \{1, \dots, n\})$ .

We construct an instance  $(\mathcal{C}, \mathcal{P}, w, 3k^2 + 2k)$  of geometric set cover as follows.

First, let us choose any bijection  $\text{order} : \{1, \dots, n^2\} \rightarrow \{1, \dots, n\} \times \{1, \dots, n\}$ .

Define  $\text{match}_v(i, j)$  and  $\text{match}_h(i, j)$  as boolean functions denoting whether two points share x or y coordinate:

$\text{match}_v(i, j)$  is **true**  $\iff \text{order}(i)$  and  $\text{order}(j)$  have the same x coordinate,

$\text{match}_h(i, j)$  is **true**  $\iff \text{order}(i)$  and  $\text{order}(j)$  have the same y coordinate.

**Points.** For  $1 \leq i, j \leq k$  and  $1 \leq t \leq n^2$  define points:

$$h_{i,j,t} := (i \cdot (n^2 + 1) + t, j \cdot (n^2 + 1)),$$

$$v_{i,j,t} := (i \cdot (n^2 + 1), j \cdot (n^2 + 1) + t).$$

Let us define sets  $H$  and  $V$  as:

$$H := \{h_{i,j,t} : 1 \leq i, j \leq k, 1 \leq t \leq n^2\},$$



$$V := \{v_{i,j,t} : 1 \leq i, j \leq k, 1 \leq t \leq n^2\}.$$

Let  $\epsilon = \frac{1}{2k^2}$ . For a point  $p = (x, y)$  we define points:

$$p^L := (x - \epsilon, y),$$

$$p^R := (x + \epsilon, y),$$

$$p^U := (x, y + \epsilon),$$

$$p^D := (x, y - \epsilon).$$

Then we define the point set as follows:

$$\mathcal{C} := H \cup \{p^L : p \in H\} \cup \{p^R : p \in H\} \cup V \cup \{p^U : p \in V\} \cup \{p^D : p \in V\}.$$

**Definition 5.4.** For every point  $p \in H$ , we name point  $p^L$  its **left guard** and point  $p^R$  its **right guard**.

Similarly for every points  $p \in V$ , we name point  $p^D$  its **lower guard** and point  $p^U$  its **upper guard**.

**Segments.** For  $1 \leq i, j \leq k$  and  $1 \leq t_1, t_2 \leq n^2$  define segments:

$$\text{hor}_{i,j,t_1,t_2} := (h_{i,j,t_1}^R, h_{i+1,j,t_2}^L),$$

$$\text{ver}_{i,j,t_1,t_2} := (v_{i,j,t_1}^U, v_{i,j+1,t_2}^D),$$

$$\text{horBeg}_{i,t} := (h_{1,i,1}^L, h_{1,i,t}^L),$$

$$\text{horEnd}_{i,t} := (h_{k,i,t}^R, h_{k,i,n^2}^R),$$

$$\text{verBeg}_{i,t} := (v_{i,1,1}^D, v_{i,1,t}^D),$$

$$\text{verEnd}_{i,t} := (v_{i,k,t}^U, v_{i,k,n^2}^U).$$

Next, we define sets of vertical and horizontal segments:

$$\begin{aligned} \text{HOR} &:= \{\text{hor}_{i,j,t_1,t_2} : 1 \leq i < k, 1 \leq j \leq k, 1 \leq t_1, t_2 \leq n^2, \text{match}_h(t_1, t_2) \text{ holds}\} \\ &\cup \{\text{horBeg}_{i,t} : 1 \leq i \leq k, 1 \leq t \leq n^2\} \\ &\cup \{\text{horEnd}_{i,t} : 1 \leq i \leq k, 1 \leq t \leq n^2\}, \end{aligned}$$

$$\begin{aligned} \text{VER} &:= \{\text{ver}_{i,j,t_1,t_2} : 1 \leq i \leq k, 1 \leq j < k, 1 \leq t_1, t_2 \leq n^2, \text{match}_v(t_1, t_2) \text{ holds}\} \\ &\cup \{\text{verBeg}_{i,t} : 1 \leq i \leq k, 1 \leq t \leq n^2\} \\ &\cup \{\text{verEnd}_{i,t} : 1 \leq i \leq k, 1 \leq t \leq n^2\}. \end{aligned}$$

You can see an example of these segments in Figure 5.3.

Finally, we also define a set of right-diagonal segments:

$$\text{DIAG} := \{(h_{i,j,t}, v_{i,j,t}) : 1 \leq i, j \leq k, 1 \leq t \leq n^2, \text{order}(t) \in f(i, j)\}.$$

You can see an example of such segments in Figure 5.2.

Every segment in **DIAG** connects points  $(i(n^2+1)+t, j \cdot (n^2+1))$  and  $(i \cdot (n^2+1), j(n^2+1) + t)$  for some  $1 \leq i, j \leq k, 1 \leq t \leq n^2$ . The line on which it lies can be described by linear equation  $y = -x + (t + (i + j)(n^2 + 1))$ , thus these segments are in fact right-diagonal.



Figure 5.2: **Vertices and segments in DIAG.**

This is an example of constructed points any  $1 \leq i, j \leq k$ . Points from  $H$  and  $V$  are marked in black, their guards are marked in blue. You can also see segments from DIAG with their weights (equal to  $\delta$ ).



Figure 5.3: **Vertices and segments in HOR.**

This is an example for  $n = 2$  and any  $1 \leq j \leq k$ . Points from  $H$  are marked in black, their guards are marked in light blue.  $t_{i,j}$  is a notation that we use for  $\text{order}^{-1}(i,j)$ . Segments are represented as arcs between endpoints. You can see  $\text{horBeg}_{j,t}$  segments in red.  $\text{horBeg}_{j,1}$  is degenerated to a single point at  $h_{1,1,t_{1,1}}^L$ . Segments  $\text{hor}_{i,j,t_{x_1,y},t_{x_2,y}}$  are marked in blue and green. Blue segments connect  $t_{x_1,y}$  and  $t_{x_2,y}$  such that they share y-coordinate equal to 1, for green segments it is equal to 2.

742 The constructed segment set is defined as:

$$\mathcal{P} := \text{HOR} \cup \text{VER} \cup \text{DIAG}.$$

743 The weight of each segment in  $\text{HOR} \cup \text{VER}$  is equal to its length, while every segment in  
744  $\text{DIAG}$  has weight  $\delta := \frac{1}{4k^4}$ .

$$w(s) = \begin{cases} \text{length}(s) & \text{if } s \in \text{HOR} \cup \text{VER} \\ \delta & \text{if } s \in \text{DIAG} \end{cases}$$

745 Now, we prove that the constructed instance of geometric set cover with weighted segments  
746 is indeed a correct and sound reduction of the grid tiling problem. Lemma 5.1 proves that if  
747 the solution of the instance of the grid tiling instance exists, then there exists a solution with  
748 bounded size and weight of the constructed instance of geometric set cover problem.

749 Then Lemma 5.5 proves that if the solution of the geometric set cover instance with  
750 bounded weight exists, then there exists a solution to the original grid tiling instance.

751 **Lemma 5.1.** *If there exists a solution of the grid tiling instance  $(f_{i,j})$ , then there exists*  
752 *a solution of the instance  $(\mathcal{C}, \mathcal{P}, w, 3k^2 + 2k)$  of geometric set cover with weight  $2k^2(n^2 + 1) -$*   
753  *$4k^2\epsilon - 4k(1 - \epsilon) + k^2\delta$ .*

754 *Proof.* Suppose there exists a solution  $x, y$  of the instance  $(f_{i,j})$  of the grid tiling problem.

755 We define the proposed solution  $\mathcal{R} \subset \mathcal{P}$  of the instance of geometric set cover in three  
756 parts  $D \subset \text{DIAG}$ ,  $A \subset \text{HOR}$  and  $B \subset \text{VER}$ :

$$\begin{aligned} D &:= \{(v_{i,j,t}, h_{i,j,t}) : 1 \leq i, j \leq k, t = \text{order}^{-1}(x(i), y(j))\}, \\ A &:= \{\text{horBeg}_{i, \text{order}^{-1}(x(1), y(i))} : 1 \leq i \leq k\} \\ &\quad \cup \{\text{horEnd}_{i, \text{order}^{-1}(x(k), y(i))} : 1 \leq i \leq k\} \\ &\quad \cup \{\text{hor}_{i,j, \text{order}^{-1}(x(i), y(j)), \text{order}^{-1}(x(i+1), y(j))} : 1 \leq i < k, 1 \leq j \leq k\}, \\ B &:= \{\text{verBeg}_{i, \text{order}^{-1}(x(i), y(1))} : 1 \leq i \leq k\} \\ &\quad \cup \{\text{verEnd}_{i, \text{order}^{-1}(x(i), y(k))} : 1 \leq i \leq k\} \\ &\quad \cup \{\text{ver}_{i,j, \text{order}^{-1}(x(i), y(j)), \text{order}^{-1}(x(i), y(j+1))} : 1 \leq i \leq k, 1 \leq j < k\}, \\ \mathcal{R} &:= D \cup A \cup B. \end{aligned}$$

Since  $\mathcal{C} = H \cup V$ , we show that  $\mathcal{R}$  covers the whole set  $H$ , proof for  $V$  is analogous.

Take any  $1 \leq j \leq k$  and define  $t_i := \text{order}^{-1}(x(i), y(j))$ . The two leftmost segments in  $A$  for this  $j$  are  $\text{horBeg}_{j,t_1} = (h_{1,j,1}^L, h_{1,j,t_1}^L)$  and  $\text{hor}_{1,j,t_1,t_2} = (h_{1,j,t_1}^R, h_{2,j,t_2}^L)$ . Therefore points  $h_{1,j,x}^L$  and  $h_{1,j,x}^R$  for all  $1 \leq x \leq n^2$  are covered by  $\text{horBeg}_{j,t_1}$  and  $\text{hor}_{1,j,t_1,t_2}$ , excluding point  $h_{1,j,t_1}$ .

Analogously for  $2 \leq i \leq k-1$  for two consecutive segments  $\text{hor}_{i-1,j,t_{i-1},t_i}$  and  $\text{hor}_{i,j,t_i,t_{i+1}}$  we prove that all points  $h_{i,j,x}^L$  and  $h_{i,j,x}^R$  for all  $1 \leq x \leq n^2$  are covered by these segments excluding point  $h_{i,j,t_i}$ .

Finally  $\text{hor}_{k-1,j,t_{k-1},t_k}$  and  $\text{horEnd}_{j,t_k}$  cover all points  $h_{k,j,x}^L$  and  $h_{k,j,x}^R$  for all  $1 \leq x \leq n^2$  excluding point  $h_{k,j,t_k}$ .

$D$  covers all points  $h_{i,j,t_i}$  and  $v_{i,j,t_i}$ , therefore all points in  $H$  are covered.

Size of this proposed solution is:

$$|\mathcal{R}| = |D| + |A| + |B| = k^2 + (k+1)k + (k+1)k = 3k^2 + 2k.$$

Then, we need to compute the total weight of the solution  $\mathcal{R}$ . First, we compute the sum of weights of segments in  $A$ . Fix  $1 \leq j \leq k$  and compute segments collinear with the  $j$ -th line. All points  $h_{i,j,t}^L$  and  $h_{i,j,t}^R$  for every  $1 \leq i \leq k$  and  $1 \leq t \leq n^2$  are covered by  $A$  excluding points  $h_{i,j,\text{order}^{-1}(x(i),y(j))}^L$  and  $h_{i,j,\text{order}^{-1}(x(i),y(j))}^R$ . Every such point leaves a gap of length  $2\epsilon$  between  $h_{i,j,\text{order}^{-1}(x(i),y(j))}^L$  and  $h_{i,j,\text{order}^{-1}(x(i),y(j))}^R$ . Therefore, the total weight of segments in  $A$  that lie on the line in question equals the length of the segment  $(h_{i,1,1}^L, h_{i,k,n^2}^R)$  minus  $2\epsilon k$ , which is  $k(n^2 + 1) - 2(1 - \epsilon) - 2k\epsilon$ . We need to multiply that by  $k$ , as we consider all possible values of  $j$ .

Calculation for vertical segments is analogous and has the same result. Every segment in  $D$  has weight  $\delta$ , therefore the sum of all weights is equal to:

$$2k(k(n^2 + 1) - 2(1 - \epsilon) - 2k\epsilon) + k^2\delta = 2k^2(n^2 + 1) - 4k^2\epsilon - 4k(1 - \epsilon) + k^2\delta$$

□

**Claim 5.1.** *In any solution of the instance  $(\mathcal{C}, \mathcal{P}, w, 3k^2 + 2k)$ :*

- *left and right guards of points in  $H$  (points in  $\{p^L : p \in H\} \cup \{p^R : p \in H\}$ ) have to be covered with segments from **HOR**,*
- *lower and upper guards of points in  $V$  (points in  $\{p^D : p \in V\} \cup \{p^U : p \in V\}$ ) have to be covered with segments from **VER**.*

*Proof.* We prove the claim for the points from  $H$  as the proof for points from  $V$  is analogous.

Every segment in **VER** is vertical and has x-coordinate equal to  $i(n^2 + 1)$  for some  $1 \leq i \leq k$ , so they all have different x-coordinate than any left or right guard of points in  $H$ .

Every point  $x$ , which is a left or right guard of points in  $H$  have  $kn^2$  segments from **DIAG** that intersect with the horizontal line that goes through  $x$ . All of these segments intersect with this line in points from set  $H$ , therefore none of them cover any of the guards.

Therefore none of the segments from **VER** or **DIAG** cover any of the guards of the points in  $H$ . □

Now we present a few additional properties of the constructed instance of the geometric set cover that help us to prove Lemma 5.5.

794 **Claim 5.2.** *For any  $1 \leq i, j \leq n$  and any solution of the instance  $(\mathcal{C}, \mathcal{P}, w, 3k^2 + 2k)$  all,*  
795 *but at most one point  $h_{i,j,t}$  and at most one point  $v_{i,j,t}$  for  $1 \leq t \leq n^2$  must be covered with*  
796 *segments from HOR or VER.*

797 *Proof.* We prove the claim for horizontal segments, as the proof for vertical segments is ana-  
798 loguous.

799 We prove this by contradiction. Assume that we have two points  $h_{i,j,t_1}, h_{i,j,t_2}$  such that  
800 they are not covered with segments from HOR for any  $1 \leq t_1 < t_2 \leq n^2$ .

801 Point  $h_{i,j,t_1}^R$  has to be covered with HOR by Claim 5.1. Every segment in HOR cover-  
802 ing  $h_{i,j,t_1}^R$ , but not  $h_{i,j,t_1}$  must start at  $h_{i,j,t_1}^R$  and all such segments cover also  $h_{i,j,t_2}$ . This  
803 contradicts the assumption, which concludes the proof.  $\square$

804 **Lemma 5.2.** *For every solution of the instance  $(\mathcal{C}, \mathcal{P}, w, 3k^2 + 2k)$ , the sum of weights of*  
805 *segments chosen from sets HOR and VER is at least  $2k^2(n^2 + 1) - 4k^2\epsilon - 4k(1 - \epsilon)$ .*

806 *Proof.* We prove the lemma for vertical lines, as the proof for horizontal segments is analogous.

807 Let us fix  $1 \leq i \leq k$ .

808 We provide a lower bound for the sum of lengths of vertical segments from  $\mathcal{R} \cap \text{VER}$ . This  
809 bound is the same for each  $i$  and is the same for horizontal lines, thus we need to multiply  
810 such bound by  $2k$ .

(1) The total length between  $v_{i,1,1}^D$  and  $v_{i,k,n^2}^U$  is:

$$(k(n^2 + 1) + n^2 + \epsilon) - ((n^2 + 1) + 1 - \epsilon) = k(n^2 + 1) - 2(1 - \epsilon).$$

811 (2) For every  $1 \leq j \leq k$  there exists at most one  $1 \leq t \leq n^2$  such that  $v_{i,j,t}$  is not covered  
812 by segments from VER (Claim 5.2). Its guards (see Definition 5.4)  $v_{i,j,t}^U$  and  $v_{i,j,t}^D$  have  
813 to be covered in VER (Claim 5.1). Therefore, at most  $k$  spaces of length  $2\epsilon$  can be left  
814 not covered by segments from VER between  $v_{i,1,1}^D$  and  $v_{i,k,n^2}^U$ .

The sum of these lower bounds for vertical and horizontal lines is:

$$2k(k(n^2 + 1) - 2k\epsilon - 2(1 - \epsilon)) = 2k^2(n^2 + 1) - 4k^2\epsilon - 4k(1 - \epsilon)$$

815  $\square$

816 Let us name the bound from the previous lemma as  $W_{hv} := 2k^2(n^2 + 1) - 4k^2\epsilon - 4k(1 - \epsilon)$   
817 for future reference.

818 **Lemma 5.3.** *Let  $\mathcal{R}$  be a solution of a constructed instance  $(\mathcal{C}, \mathcal{P}, w, 3k^2 + 2k)$  with weight*  
819 *at most  $2k^2(n^2 + 1) - 4k^2\epsilon - 4k(1 - \epsilon) + k^2\delta$ . Then for every  $1 \leq i, j \leq k$  there exists such*  
820  *$1 \leq t \leq n^2$  that:*

- 821 (1)  $v_{i,j,t}, h_{i,j,t}$  are not covered by segments from VER or HOR;
- 822 (2) segment  $(v_{i,j,t}, h_{i,j,t})$  is in solution  $\mathcal{R}$ ;
- 823 (3)  $\text{order}(t) \in f(i, j)$ , that is,  $\text{order}(t)$  is an allowed tile for  $(i, j)$ ;
- 824 (4) for every  $1 \leq s \leq n^2$ ,  $s \neq t$ ,  $v_{i,j,s}$  is covered in VER;
- 825 (5) for every  $1 \leq s \leq n^2$ ,  $s \neq t$ ,  $h_{i,j,s}$  is covered in HOR.

*Proof.* At most one of points  $\{h_{i,j,t_x} : 1 \leq t_x \leq n^2\}$  and one of points  $\{v_{i,j,t_y} : 1 \leq t_y \leq n^2\}$  is covered with **DIAG** (Claim 5.2).

Moreover, exactly one such point  $h_{i,j,t_x}$  and one such point  $v_{i,j,t_y}$  is covered with **DIAG**, because if none of them were covered, then the solution would have to have weight at least  $W_{hv} + 2\epsilon$  (Lemma 5.2), which is more than  $2k^2(n^2 + 1) - 4k^2\epsilon - 4k(1 - \epsilon) + k^2\delta$ .

We observe that points  $h_{i,j,t_x}$  and  $v_{i,j,t_y}$  have to be covered with the same segment from **DIAG**. Indeed we need to use at least  $k^2$  of them to use exactly one **DIAG** segment for every pair of  $1 \leq i, j \leq k$ , if we used 2 segments from **DIAG** for one pair  $(i, j)$ , then we would have used  $W_{hv} + k^2\delta + \delta$  (Lemma 5.2), which is more than  $2k^2(n^2 + 1) - 4k^2\epsilon - 4k(1 - \epsilon) + k^2\delta$ . Since points  $h_{i,j,t_x}$  and  $v_{i,j,t_y}$  are covered by a single segment from **DIAG**, we have  $t_x = t_y$ .

Therefore  $t_x = t_y$  and  $\text{order}(t_x)$  is an allowed tile for  $(i, j)$  because the corresponding segment is in **DIAG**.  $\square$

We refer to the function mapping  $1 \leq x \leq k$  to  $t_x$  from Lemma 5.3 as **diagonal** :  $\{1 \dots k\} \times \{1 \dots k\} \rightarrow \{1 \dots n^2\}$ .

**Lemma 5.4.** *For any solution  $\mathcal{R}$  of a constructed instance  $(\mathcal{C}, \mathcal{P}, w, 3k^2 + 2k)$  with weight at most  $2k^2(n^2 + 1) - 4k^2\epsilon - 4k(1 - \epsilon) + k^2\delta$ :*

1. for any  $1 \leq i < k, 1 \leq j \leq k$ ,  $\text{match}_h(\text{diagonal}(i, j), \text{diagonal}(i + 1, j))$  is **true**;
2. for any  $1 \leq i \leq k, 1 \leq j < k$ ,  $\text{match}_v(\text{diagonal}(i, j), \text{diagonal}(i, j + 1))$  is **true**.

*Proof.* We prove (1) by contradiction, the proof of (2) is analogous.

Let us take any  $1 \leq i < k, 1 \leq j \leq k$  and name  $t_1 = \text{diagonal}(i, j)$  and  $t_2 = \text{diagonal}(i + 1, j)$ . We also assume that  $\text{match}_h(t_1, t_2)$  is **false**, which is equivalent to the fact that segment  $(h_{i,j,t_1}^R, h_{i+1,j,t_2}^L)$  is not in set **HOR**.

Therefore  $h_{i,j,t_1}$  and  $h_{i+1,j,t_2}$  are not covered by segments from **HOR** (Lemma 5.3), while  $h_{i,j,t_1}^R$  and  $h_{i+1,j,t_2}^L$  have to be covered by segments from **HOR** (Claim 5.1).

Every segment from **HOR** starts at point  $h_{x,y,z_1}^R$  and ends at point  $h_{x+1,y,z_2}^L$  for some  $1 \leq x < k, 1 \leq y \leq k$  and  $1 \leq z_1, z_2 \leq n^2$ . All of the points between  $h_{i,j,t_1}^R$  and  $h_{i+1,j,t_2}^L$  are covered by segments in **HOR** and there is no segment  $(h_{i,j,t_1}^R, h_{i+1,j,t_2}^L)$  in **HOR**. Hence, there are at least two different segments covering them. One of them must begin at  $h_{i,j,t_1}^R$  and end at  $h_{i+1,j,z_2}^L$  and there must be other one that begins at  $h_{i,j,z_1}^R$  and ends at  $h_{i+1,j,t_2}^L$  for some  $1 \leq z_1, z_2 \leq n^2$ .

Thus, the space between  $h_{i,j,z_1}^R$  and  $h_{i,j+1,z_2}^L$  would be covered twice and is longer than  $\epsilon$ . By Lemma 5.2, the lower bound for weight of such a solution is  $W_{hv} + \epsilon$  which is more than  $2k^2(n^2 + 1) - 4k^2\epsilon - 4k(1 - \epsilon) + k^2\delta$ .

Therefore  $h_{i,j,t_1}^R$  and  $h_{i+1,j,t_2}^L$  must be covered by one segment from **HOR**,  $(h_{i,j,t_1}^R, h_{i+1,j,t_2}^L)$  is a segment in **HOR** and  $\text{match}_h(t_1, t_2)$  is **true**.  $\square$

**Lemma 5.5.** *If there exists solution of instance  $(\mathcal{C}, \mathcal{P}, w, 3k^2 + 2k)$  with weight at most  $2k^2(n^2 + 1) - 4k^2\epsilon - 4k(1 - \epsilon) + k^2\delta$ , then there exists a solution for the grid tiling instance  $(f_{i,j})$ .*

*Proof.* Take **diagonal** function from Lemma 5.3.

To define the  $x$  function for every  $1 \leq i \leq k$  set  $x(i) := x_i$  where  $(x_i, a) = \text{order}(v_{i,1})$ . Similarly, to define the  $y$  function, for every  $1 \leq i \leq k$  set  $y(i) := y_i$  where  $(b, y_i) = \text{order}(h_{1,i})$ .

To prove that it is a correct solution for grid tiling, we need to prove that for every  $1 \leq i, j \leq k$   $(x(i), y(j))$  is in allowed tiles set  $f(i, j)$ .

869 Let us take any  $1 \leq i, j \leq k$ . By Lemma 5.4 and simple induction, we know that  
 870  $\text{match}_h(\text{diagonal}(1, j), \text{diagonal}(i, j))$  and  $\text{match}_v(\text{diagonal}(i, 1), \text{diagonal}(i, j))$  are **true**. There-  
 871 fore  $\text{order}(\text{diagonal}(i, j)) = (x(i), y(j))$ . By Lemma 5.3 we know that  $\text{order}(\text{diagonal}(i, j))$  is in  
 872  $f(i, j)$ . Therefore  $(x(i), y(j))$  is in  $f(i, j)$ .  $\square$

873 *Proof of Theorem 5.1.* Follows from Lemmas 5.1 and 5.5.  $\square$

874 TODO: Add reference when known In proof of reduction we did not use the assumption  
 875 that the solution is of bounded size. Thus this reduction proves that the problem is not only  
 876 W[1]-hard, but assuming ETH there also does not exist permissive FPT algorithm for this  
 877 problem.





## Chapter 6

# Geometric Set Cover with lines

### 6.1. Lines parallel to one of the axis

When  $\mathcal{R}$  consists only of lines parallel to one of the axis, the problem can be solved in polynomial time.

We create bipartial graph  $G$  with node for every line on the input split into sets:  $H$  – horizontal lines and  $V$  – vertical lines. If any two lines cover the same point from  $\mathcal{C}$ , then we add edge between them.

Of course there will be no edges between nodes inside  $H$ , because all of them are pararell and if they share one point, they are the same lines. Similar argument for  $V$ . So the graph is bipartial.

Now Geometric Set Cover can be solved with Vertex Cover on graph  $G$ . Since Vertex Cover (even in weighted setting) on bipartial graphs can be solved in polynomial time.

Short note for myself just to remember how to this in polynomial time:

Non-weighted setting - Konig theorem + max matching

Weighted setting - Min cut in graph of  $\neg A$  or  $\neg B$  (edges directed from  $V$  to  $H$ )

### 6.2. FPT for arbitrary lines

You can find this is Platypus book. We will show FPT kernel of size at most  $k^2$ .

(Maybe we need to reduce lines with one point/points with one line).

For every line if there is more than  $k$  points on it, you have to take it. At the end, if there is more than  $k^2$  points, return NO. Otherwise there is no more than  $k^4$  lines.

In weighted settings among the same lines with different weights you leave the cheapest one and use the same algorithm.

### 6.3. APX-completeness for arbitrary lines

We will show a reduction from Vertex Cover problem. Let's take an instance of the Vertex Cover problem for graph  $G$ . We will create a set of  $|V(G)|$  pairwise non-pararell lines, such that no three of them share a common point.

Then for every edge in  $(v, w) \in E(G)$  we put a point on crossing of lines for vertices  $v$  and  $w$ . They are not pararell, so there exists exactly one such point and any other line do not cover this point (any three of them do not cross in the same point).

Solution of Geometric Set Cover for this instance would yield a sound solution of Vertex Cover for graph  $G$ . For every point (edge) we need to choose at least one of lines (vertices)  $v$  or  $w$  to cover this point.

Vertex Cover for arbitrary graph is APX-complete, so this problem is also APX-complete.

## 6.4. 2-approximation for arbitrary lines

Vertex Cover has an easy 2-approximation algorithm, but here very many lines can cross through the same point, so we can do  $d$ -approximation, where  $d$  is the biggest number of lines crossing through the same point. So for set where any 3 lines do not cross in the same point it yields 2-approximation.

The problematic cases are where through all points cross at least  $k$  points and all lines have at least  $k$  points on them. It can be created by casting  $k$ -grid in  $k$ -D space on 2D space.

Greedy algorithm yields  $\log |\mathcal{R}|$ -approximation, but I have example for this for bipartial graph and reduction with taking all lines crossing through some point (if there are no more than  $k$ ) would solve this case. So maybe it works.

Unfortunately I have not done this :(

I can link some papers telling it's hard to do.

## 6.5. Connection with general set cover

Problem with finite set of lines with more dimensions is equivalent to problem in 2D, because we can project lines on the plane which is not perpendicular to any plane created by pairs of (point from  $\mathcal{C}$ , line from  $\mathcal{P}$ ).

Of course every two lines have at most one common point, so is every family of sets that have at most one point in common equivalent to some geometric set cover with lines?

No, because of Desargues's theorem. Have to write down exactly what configuration is banned.

## 932 Chapter 7

# 933 Geometric Set Cover with polygons

### 934 7.1. State of the art

935 Covering points with weighted discs admits PTAS [Li and Jin, 2015] and with fat polygons  
936 with  $\delta$ -extensions with unit weights admits EPTAS [Har-Peled and Lee, 2009].

937 Although with thin objects, even if we allow  $\delta$ -expansion, the Set Cover with rectangles is  
938 APX-complete (for  $\delta = 1/2$ ), it follows from APX-completeness for segments with  $\delta$ -expansion  
939 in Section ??.

940 Covering points with squares is W[1]-hard [Marx, 2005]. It can be proven that assuming  
941 *SETH*, there is no  $f(k) \cdot (|\mathcal{C}| + |\mathcal{P}|)^{k-\epsilon}$  time algorithm for any computable function  $f$  and  
942  $\epsilon > 0$  that decides if there are  $k$  polygons in  $\mathcal{P}$  that together cover  $\mathcal{C}$ , *Theorem 1.9* in [Marx  
943 and Pilipczuk, 2015].



## 944 Chapter 8

## 945 Conclusions

946 We do not know FPT for axis-parallel segments without  $\delta$ -extensions.



## 947 Bibliography

- 948 [Cygan et al., 2015] Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D.,  
949 Pilipczuk, M., Pilipczuk, M., and Saurabh, S. (2015). *Parameterized Algorithms*. Springer.
- 950 [Har-Peled and Lee, 2009] Har-Peled, S. and Lee, M. (2009). Weighted geometric set cover  
951 problems revisited. *Journal of Computational Geometry*, 3.
- 952 [Håstad, 2001] Håstad, J. (2001). Some optimal inapproximability results. *J. ACM*,  
953 48(4):798–859.
- 954 [Li and Jin, 2015] Li, J. and Jin, Y. (2015). A PTAS for the weighted unit disk cover problem.  
955 *CoRR*, abs/1502.04918.
- 956 [Marx, 2005] Marx, D. (2005). Efficient approximation schemes for geometric problems? In  
957 Brodal, G. S. and Leonardi, S., editors, *Algorithms – ESA 2005*, pages 448–459, Berlin,  
958 Heidelberg. Springer Berlin Heidelberg.
- 959 [Marx, 2007] Marx, D. (2007). On the optimality of planar and geometric approximation  
960 schemes. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS*  
961 *2007), October 20-23, 2007, Providence, RI, USA, Proceedings*, pages 338–348. IEEE Com-  
962 puter Society.
- 963 [Marx and Pilipczuk, 2015] Marx, D. and Pilipczuk, M. (2015). Optimal parameterized algo-  
964 rithms for planar facility location problems using voronoi diagrams. *CoRR*, abs/1504.05476.
- 965 [Pilipczuk et al., 2016] Pilipczuk, M., van Leeuwen, E. J., and Wiese, A. (2016). Approxima-  
966 tion and parameterized algorithms for geometric independent set with shrinking. *CoRR*,  
967 abs/1611.06501.