

University of Warsaw  
Faculty of Mathematics, Informatics and Mechanics

**Katarzyna Kowalska**

Student no. 371053

# Approximation and Parametrized Algorithms for Geometric Set Cover

Master's thesis  
in COMPUTER SCIENCE

Supervisor:  
**dr Michał Pilipczuk**  
Instytut Informatyki

June 2020

## **Supervisor's statement**

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

Date

Supervisor's signature

## **Author's statement**

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

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

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

Date

Author's signature

## Abstract

W pracy przedstawiono prototypową implementację blabalizatora różnicowego bazującą na teorii fetorów  $\sigma$ - $\rho$  profesora Fifaka. Wykorzystanie teorii Fifaka daje wreszcie możliwość efektywnego wykonania blabalizy numerycznej. Fakt ten stanowi przełom technologiczny, którego konsekwencje trudno z góry przewidzieć.

## Keywords

blabaliza różnicowa, fetory  $\sigma$ - $\rho$ , fooizm, blarbarucja, blaba, fetoryka, baleronik

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

11.3 Informatyka

## Subject classification

D. Software

D.127. Blabalgorithms

D.127.6. Numerical blabalysis

## Tytuł pracy w języku polskim

Algorytmy parametryzowania i trudność aproksymacji problemu pokrywania zbiorów na płaszczyźnie



# Contents

<b>1. Introduction</b>	5
<b>2. Definitions</b>	7
<b>3. Geometric Set Cover with segments</b>	9
3.1. FPT for segments	9
3.1.1. Segments parallel to one of the axis	9
3.1.2. Segments in $d$ directions	9
3.1.3. Segments in arbitrary direction	9
3.2. APX-completeness for segments parallel to axis	10
3.3. Weights	10
3.3.1. FPT for segments parallel with $\delta$ -extensions	10
3.3.2. W[1]-completeness for arbitrary segments with weights	10
3.3.3. What is missing	10
<b>4. Geometric Set Cover with lines</b>	11
4.1. Lines parallel to one of the axis	11
4.2. FPT for arbitrary lines	11
4.3. APX-completeness for arbitrary lines	11
4.4. 2-approximation for arbitrary lines	12
4.5. Connection with general set cover	12
<b>5. Geometric Set Cover with polygons</b>	13
5.1. Introduction	13
5.2. FPT – ?? I don't know :(	13
5.3. APX-completeness for rectangles with $\delta$ -expansion without weights	13
5.4. $1 + \epsilon$ approximation algorithm for weighted polygons of bounded thickness $\theta$	13
5.4.1. Sparsifying the family	13
5.4.2. Dynamic programming	15
<b>6. Conclusions</b>	17
<b>Bibliografia</b>	19



# Chapter 1

## Introduction

This is some very boring and really nothing on the topic introduction.





## Chapter 2

# Definitions

Some definitions what geometric set cover is.  $\mathcal{P}$  – set of objects,  $\mathcal{C}$  – set of points. Choose  $\mathcal{R} \subset \mathcal{P}$  such that every point in  $\mathcal{C}$  is inside some element from  $\mathcal{R}$  and  $|\mathcal{R}|$  is minimal.

In parametrized setting we only look among  $|\mathcal{R}| \leq k$ . In weighted settings there is some  $f : \mathcal{P} \rightarrow \mathbb{R}$  and we minimize  $\sum_{R \in \mathcal{R}} f(R)$ .



## Chapter 3

# Geometric Set Cover with segments

### 3.1. FPT for segments

#### 3.1.1. Segments parallel to one of the axis

You can find this in Platypus book.

We'll show  $\mathcal{O}(2^k)$  branching algorithm. Let's take point  $K$  that hasn't been covered yet with the smallest coordinate in lexicographical order. We need to cover  $K$  with some of the remaining segments.

We choose one of the 2 directions on which we will cover this point. In this direction we take greedily the segment that will cover the most points (there are points in  $\mathcal{C}$  only on one side of  $K$  in this direction, so all segments covering  $K$  in this direction create monotone sequence of sets – zbiory zstępujące).

#### 3.1.2. Segments in $d$ directions

The same algorithm as before but in complexity  $\mathcal{O}(d^k)$ .

#### 3.1.3. Segments in arbitrary direction

If there is a segment that covers no more points than some other segment (inclusion-wise), remove it. Repeat until such segment exists.

We define long segments as having more than  $k$  points on them and all others will be called short segments.

We also sort segments on their respective lines (eg. lexicographically over coordinates).

We will find the kernel where every segment is short. Then we will have at most  $k^2$  points (otherwise there is no solution). So we will have at most  $\mathcal{O}(k^4)$  different segments in the kernel. (We can choose leftmost and rightmost point for each such segment and all other points are implied).

As long as there is a long segment  $s$ , do branching.

Consider the line containing  $s$  and without loss of generality, assume  $s$  is the first (leftmost) long segment on this line.

We need to take some segment on this line, because using only segments on different lines we would use more than  $k$  segments.

We either:

- take  $s$  to solution

- first point on the line is covered by segment in this direction – cover some prefix of points on  $s$  by some earlier (therefore short) segment
- first point on the line is covered by segment in other direction – leave some of the first points to be covered by segments in different directions (no more than  $k$ ) and cover the suffix by some other segment

If we decide that we cover  $0 < i \leq k$  first point by some earlier segment, we just take the segment on this line that ends exactly in  $i$ -th point on this line (there is at most one such segment, because we removed those included in it).

If we decide that first  $0 < j < k$  points will be covered by some segments in other directions and  $j + 1$ -th will be the first to be covered by segment in this direction, we just take the segment on this line that starts exactly in  $j + 1$ -th point on this line

So we have branching over  $2k$  choices on this segment.

So the final algorithm complexity is  $\mathcal{O}(k^k)$  from branching.

## 3.2. APX-completeness for segments pararell to axis

It works even with extensions for unit weights.

We will show reduction from MAXSAT to Geometric Set Cover with segments pararell to axis.

## 3.3. Weights

### 3.3.1. FPT for segments pararell with $\delta$ -extensions

### 3.3.2. W[1]-completeness for arbitrary segments with weights

### 3.3.3. What is missing

We don't know FPT for pararell segments and arbitrary lines with  $\delta$ -extensions.

## Chapter 4

# Geometric Set Cover with lines

### 4.1. 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 parallel 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  $H$ )

### 4.2. FPT for arbitrary lines

You can find this in 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.

### 4.3. APX-completeness for arbitrary lines

We will show reduction from Vertex Cover problem. Let's take instance of Vertex Cover problem for graph  $G$ . We will create set of  $|V(G)|$  pairwise non-parallel lines that any 3 of them don't cross in the same point.

Then for every edge in  $(v, w) \in E(G)$  we put point on crossing of lines for vertices  $v$  and  $w$ . They are not parallel, so there exists exactly one such point and any other line don't cover this point (any 3 of them don't cross in the same point). So this solving geometric set cover will choose for every edge at least one vertex connected by this edge, because we need to choose at least one of lines corresponding to  $v$  or  $w$  to cover this point.

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

#### 4.4. 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 don't 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 bipartite 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 haven't done this :(

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

#### 4.5. 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.

## Chapter 5

# Geometric Set Cover with polygons

### 5.1. Introduction

The problem is APX-complete and W[1]-complete, so we introduce  $\delta$ -expansions.

### 5.2. FPT – ?? I don't know :(

### 5.3. APX-completeness for rectangles with $\delta$ -expansion without weights

It follows from APX-completeness for segments with  $\delta$ -expansion.

### 5.4. $1+\epsilon$ approximation algorithm for weighted polygons of bounded thickness $\theta$

This should be written.

**Definition 5.4.1** *Thickness of the polygon is the ratio of the circumscribed circle's radius to the inscribed circle's radius.*

**Definition 5.4.2 (MWSCP)** *TODO: wstawić to jakoś wcześniej i inaczej Minimal Weight Set Cover for Polygons*

**Theorem 5.4.1 (EPTAS for MWSCP with bounded thickness and  $\delta$ -expansion)** *There is a randomized algorithm that given a weighted family  $\mathcal{P}$  of  $n$  polygons with thickness bounded by  $\theta$  and set  $\mathcal{C}$  of  $m$  points with total encoding size of both sets  $N$ , and parameters  $\delta, \epsilon$ , runs in time  $f(\epsilon, \delta, \theta) \cdot (nN)^c$  for some computable functions  $f$  and constant  $c$ , and outputs a subfamily  $\mathcal{S} \subseteq \mathcal{P}$  such that  $\mathcal{S}^\delta$  covers the  $\mathcal{C}$  and  $w(\mathcal{S}) \leq (1 + \epsilon)OPT(\mathcal{P})$  with probability at least  $1/2$ .*

#### 5.4.1. Sparsifying the family

Intuitively, we will create a new input family  $\mathcal{P}'$  of polygons that can cover set of points  $\mathcal{C}$  if and only if set  $\mathcal{P}$  can cover set of  $\mathcal{C}$  and  $OPT(\mathcal{P}')$  is worse only by  $\mathcal{O}(\epsilon)$ -fraction of  $OPT(\mathcal{P})$ . The polygons in  $\mathcal{P}'$  will be classified into groups of similar size of edge of their circumscribed squares.

Ogólnie wszystko tutaj będzie takie samo jak w paperze, ale wstawimy stałą dla siatki  $1/(\delta\theta\epsilon)$  zamiast  $1/\delta\epsilon$ .

$L = (1/\delta\theta\epsilon)^\ell$  for some  $\ell = \mathcal{O}(N)$  – limit for data.

Let's denote  $d_i$  as a length of edge of the circumscribed square on a polygon  $P_i \in \mathcal{P}$ .

**Partition into layers** Let's define a partition:

$$(\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_\ell)$$

of  $\mathcal{P}$  and such reals  $\nu_t, \mu_t$  for  $t = 1, 2, \dots, \ell$  with the following properties satisfied for each  $t \in \{1, 2, \dots, \ell\}$ :

- $\nu_t \leq d_i \leq \mu_t$  for each  $P_i \in \mathcal{P}_t$
- $\nu_t = \mu_{t-1}$  (expect for  $t = 1$ ) and  $\mu_t/\nu_t = (1/\delta\theta\epsilon)^{1/\epsilon}$
- $\nu_1 \geq 1, \mu_\ell \geq L$ , and all numbers  $\nu_t$  and  $\mu_t$  apart from  $\nu_1$  are integers.

How to divide these polygons and choose numbers is pretty straightforward, but we also use some shifting parameter  $0 \leq b \leq 1/\epsilon$  to be determined later.

$$\nu_t = (1/\delta\theta\epsilon)^{t/\epsilon+b} \quad \mu_t = (1/\delta\theta\epsilon)^{(t+1)/\epsilon+b}$$

**Hierarchical grid structure** Let  $a \in \{1, \dots, L-1\}$  be an integer shift parameter, to be determined later. Given  $a$  we construct a hierarchy of grid lines in the plane.

For level  $t$ , define the *level- $t$  unit* as  $u_t = \delta\nu_t/(\theta 2\sqrt{2})$ . Note that  $u_t$  is an integer.

We define a set of horizontal lines with  $y$ -coordinates from the set:

$$a + b \cdot u_t : b \in \mathbb{Z}$$

Then for every polygon  $P_i \in \mathcal{P}_t$  if the lines (horizontal or vertical) from level  $t+1$  cross the polygon  $P_i$ , we split it according to lines to at most 4 polygons with the same weights and add these to  $\mathcal{P}'$ . Otherwise  $P_i \in \mathcal{P}'$ .

**Lemma 5.4.1** *In polynomial time one can yield a family  $\mathcal{P}'$  that satisfies*

$$OPT(\mathcal{P}') \leq (1 + 16\epsilon)OPT(\mathcal{P})$$

*with probability at least  $3/4$ . Moreover one can construct the solution  $S \subseteq \mathcal{P}$  back from the solution of  $S' \subseteq \mathcal{P}'$  such that  $w(S) \leq w(S')$ .*

*Sketch of proof* If  $\nu_t \leq d_i \leq \mu_t\epsilon$ , then there is at most  $\epsilon$  probability that with random offset  $a$ , the line will cut this polygon on the  $t$ -th level vertically. Analogically for horizontal cuts.

If  $\mu_t\epsilon < d_i < \nu_{t+1}$ , then this situation happens only for one  $b$  in set  $\{0, 1, 2, \dots, 1/\epsilon\}$ .

Then for every polygon  $P_i$  in optimal solution  $OPT$ , the expected value of sum of weights for all polygons in  $\mathcal{P}'$  corresponding to the polygon  $P_i$  is at most  $4\epsilon$ .

So with Markov inequality we can prove that  $\Pr(OPT(\mathcal{P}') > (1 + 16\epsilon)OPT(\mathcal{P})) \leq 1/4$

**Extending polygons** On every level  $t$ , for every  $P_i \in \mathcal{P}'_t$ , we will create a new polygon  $P'_i$  that consists of every cell in hierarchical grid on level  $t$ , that have non-empty intersection with  $P_i$ .

New polygon will fit inside  $P_i$  shifted to every dimension by  $u_t\sqrt{2} = \delta\nu_t/(2\theta) \leq \delta d_i/(2\theta)$ .

The larger dimension is not extended more than by  $\delta$ :

$$2 \cdot \delta d_i/(2\theta) = \delta d_i/\theta \leq \delta d_i$$

The shorter dimension is at most  $d_i/\theta$ , so it also wouldn't be extended by more than  $\delta$ .



#### 5.4.2. Dynamic programming



## Chapter 6

## Conclusions



# Bibliography

- [Bea65] Juliusz Beaman, *Morbidity of the Jolly function*, *Mathematica Absurdica*, 117 (1965) 338–9.
- [Blar16] Elizjusz Blarbarucki, *O pewnych aspektach pewnych aspektów*, *Astrolog Polski*, Zeszyt 16, Warszawa 1916.
- [Fif00] Filigran Fifak, Gizbert Gryzogrzechotalski, *O blabalii fetorycznej*, *Materiały Konferencji Euroblabal* 2000.
- [Fif01] Filigran Fifak, *O fetorach  $\sigma$ - $\rho$* , *Acta Fetorica*, 2001.
- [Głomb04] Gryzybór Głombaski, *Parazytonikacja blabiczna fetorów — nowa teoria wszystkiego*, Warszawa 1904.
- [Hopp96] Claude Hopper, *On some  $\Pi$ -hedral surfaces in quasi-quasi space*, *Omnius University Press*, 1996.
- [Leuk00] Lechoslav Leukocyt, *Oval mappings ab ovo*, *Materiały Białostockiej Konferencji Hodowców Drobiu*, 2000.
- [Rozk93] Josip A. Rozkosza, *O pewnych własnościach pewnych funkcji*, *Północnopomorski Dziennik Matematyczny* 63491 (1993).
- [Spy59] Mrowclaw Spyrpt, *A matrix is a matrix is a matrix*, *Mat. Zburp.*, 91 (1959) 28–35.
- [Sri64] Rajagopalachari Sriniswamiramanathan, *Some expansions on the Flausgloten Theorem on locally congested latches*, *J. Math. Soc.*, North Bombay, 13 (1964) 72–6.
- [Whi25] Alfred N. Whitehead, Bertrand Russell, *Principia Mathematica*, Cambridge University Press, 1925.
- [Zen69] Zenon Zenon, *Użyteczne heurystyki w blabalizie*, *Młody Technik*, nr 11, 1969.