**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

# 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 $f : \mathcal{P} - > \mathbb{R}$ 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 lexicograpical 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 greedly 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 exist two segments $a$ and $b$ in $\mathcal{P}$, such that any point covered by $a$ is also covered by $b$, then without loss of generality we can remove segment $a$ from $\mathcal{P}$. We repeat this process until no such $(a, b)$ pair exists.

Let us first assume that we reduced our instance to a kernel, where *any line* contains no more than $k$ points.

Since any segment covers a set of colinear points, for such a kernel $k$ segments can cover only at most $k^2$ points. Therefore, for the answer to be positive, the number of points has to be at most $k^2$. The number of segments is now bounded by $k^4$, since if we consider two *extreme* points covered by a given segment, then these pairs must be distinct, otherwise two segments would contain the same set of points. Since both the number of points and the number of segments is bounded by a function of $k$, this instance can be easily solved in time $O(f(k))$.

In remains to show how to construct the kernel.

Assume there exists a line $l$ containing points $x_1, \ldots x_t$, where $t \geq k+1$. Note that a segment that does not lie on $l$ can cover only at most one of the points $x_i$. Therefore, out of points $x_1$, ..., $x_{k+1}$, at least one has to be covered by a segment that lies on $l$, let us fix $x_i$ to be the first such point. Then, we can greedily choose a segment that lies on $l$, covers $x_i$, and also covers the largest number of points $x_j$ for $j > i$.

Since we have at most $k+1$ choices to branch over and each choice adds a segment to the constructed solution, we obtain an algorithm with complexity $O(k^k)$.

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

It works even with extensions for unit weights.

We will show reduction from MAX-(3,3)-SAT to Geometric Set Cover with segments pararell to axis.

### 3.2.1. Definition of MAX-(3,3)-SAT problem

Here we define MAXSAT problem.

### 3.2.2. Reduction construction

Let's take some instance of MAX-(3,3)-SAT with variables $x_1, x_2 \dots x_n$ and clauses $C_1, C_2 \dots C_n$.

We will create gadgets for choosing the value of variables (*true* or *false*) and checking if the clauses are met (any of the variables were chosen).

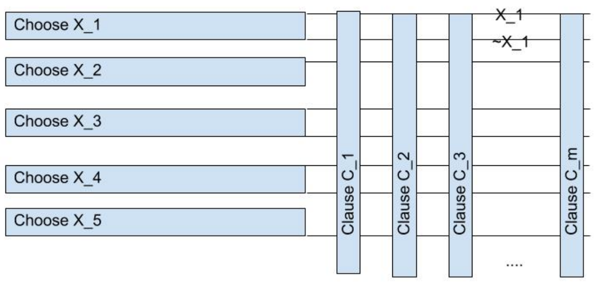



Figure 3.1: General scheme of reduction.

#### Choose $x_i$ gadget

In Figure 3.2, we show a gadget that simulates a single variable $x_i$. It consists of six points A, B, C, D, E, F, and several segments. Selecting the segment marked with $x_i$ to the solution will correspond to setting $x_i$ to *true*, while selecting the segment marked with $\neg x_i$ to setting $x_i$ to *false*. In the following lemmas, we show that this construction indeed models a binary variable.

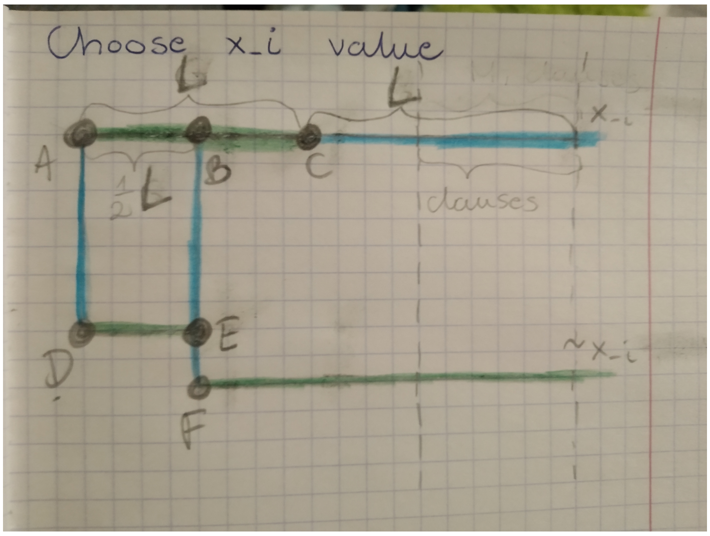



Figure 3.2: Scheme of choose $x_i$ gadget.

First, note that in the gadget there are exactly two sets of three segments that cover all points $A, B, C, D, E, F$. These two sets of segments are marked in Figure 3.2 in blue and green, respectively.

**Lemma 3.2.1** *Points $A, B, C, D, E, F$ cannot be covered using less than 3 segments (even with $1/2$-extensions).*

**Proof.** We need to take at least one segment on line $ABC$, because it's the only way to cover $C$. All other points $(D, E, F)$ are not colinear, so we need at least 2 other segments to cover them.

**Lemma 3.2.2** *If we choose both segments $x_i$ and $\neg x_i$, we need to use at least 4 segments to cover all points $A, B, C, D, E, F$ (even with $1/2$-extensions).*

**Proof.** Choosing both segments $x_i$ and $\neg x_i$ we only cover points $C$ (becuase $B$ is too far away to be covered with 1/2-extension) and $F$.

The remaining points $(A, B, D, E)$ are not colinear, so we need at least two more segments to cover them.

**Robustness to $1/2$-extension.** Take a look at Figure 3.1. The points will be included in choose gadgets (horizontal boxes) and clause gadgets (vertical boxes).

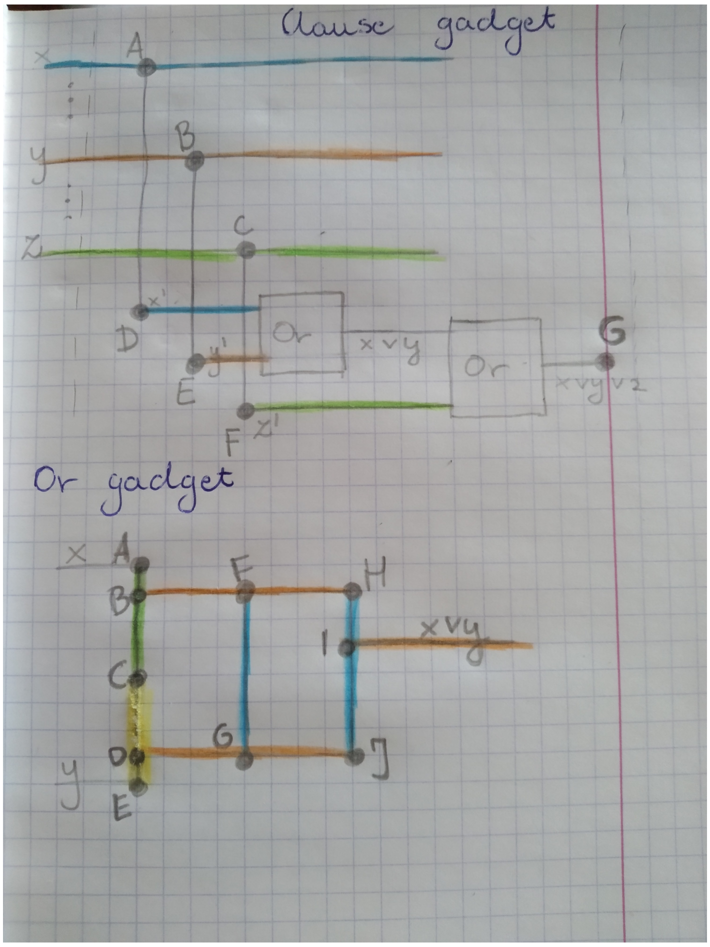
Since segment $AC$ is very long and colinear with $x_i$, after 1/2-extension it will cover a significant part of segment $x_i$, even though $x_i$ will not be chosen.

If we put all the clause gadgets in the area marked with **clauses** at gadget scheme in Figure 3.2, it is enough to prove that $AC$ will not cover any points in the **clauses** area even with 1/2-extensions.

**Lemma 3.2.3** *No points in **clauses** area can be covered by $AC$ with $1/2$-extension.*

**Proof.** Bear in mind that length of $AC$ is equal to length of $x_i$. Area **clauses** takes a second half of the segment $x_i$ and $AC$ after extension will cover the first half of segment $x_i$.

**Clause gadget**




**Lemma 3.2.4** *In order to cover $D$ $(E, F)$ point at least one of the segments $AD$ $(BE, CF)$ or $x'$ $(y', z')$.*

**Lemma 3.2.5** *Points $A$ and $D$ can be covered with one additional segment $x'$ only if $x$ is already chosen. Otherwise they can be covered with one segment only by using $AD$.*

**Lemma 3.2.6** *Points $A, B, C, D, E, F, G$ can be covered with 3 or 4 segments, depending if at least one of the segments $x, y, z$ was previously chosen.*

**Or gadget**

**Lemma 3.2.7** *Points $A, B, C, D, E, F, G, H, I, J$ can be covered using at least 4 segments even with 1/2-extension.*

**Lemma 3.2.8** *Points $A, B, C, D, E, F, G, H, I, J$ can be covered using 4 segments and segment $x \vee y$ can be chosen even with 1/2-extension only if at least one of the segments $x$ or $y$ is chosen.*

### 3.2.3. Proof that construction is sound

**Lemma 3.2.9** *If there exists setting of values of variables that exactly $k$ clauses are satisfied, we can cover all the points with $3n + 11m + (m - k)$ segments.*

**Lemma 3.2.10** *If there exists cover with $k$ segments, then also there exists solution for MAX-(3,3)-SAT.*

*TODO: Formulate this lemma better.*

## 3.3. Weights

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

### 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 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 $H$)

## 4.2. 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.

## 4.3. 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 don't cover this point (any three of them don't 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 in 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 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.

Unfortunaly 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. State of the art

Covering points with unit discs or squares is APX-hard [Marx07].

Covering points with squares is W[1]-hard [Marx05].

## 5.2. APX-completness for rectangles with $\delta$-expansion without weights

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

## 5.3. $1+\epsilon$ approximation algorithm for weighted polygons of bounded thickness $\theta$

This should be written.

**Definition 5.3.1** ***Thickness*** *of the polygon is the ratio of the circumsribed circle's radius to the inscribed circle's radius.*

**Definition 5.3.2 (MWSCP)** *TODO: wstawić to jakoś wcześniej i inaczej Minimal Weight Set Cover for Polygons*

**Theorem 5.3.1 (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 bouded 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.*

### 5.3.1. 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 $\mathrm{OPT}(\mathcal{P}')$ is worse only by $\mathcal{O}(\epsilon)$-fraction of $\mathrm{OPT}(\mathcal{P})$. 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.

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

**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 horiontal 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 5.3.1** *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 $\mathcal{P}'$ corresponsing 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 $\delta$.

#### 5.3.2. Dynamic programming

# Chapter 6

# Conclusions

# Bibliography


[Marx05] D. Marx, *Efficient Approximation Schemes for Geometric Problems?*, In *Brodal G.S., Leonardi S. (eds) Algorithms – ESA 2005. ESA 2005.* Lecture Notes in Computer Science, vol 3669. Springer, Berlin, Heidelberg

[Marx07] D. Marx, *On the optimality of planar and geometric approximation schemes*, In *FOCS 2007*, pages 338–348. IEEE Computer Society, 2007