The Parameterized Complexity of the Rectangle Stabbing Problem and its Variants[1]

Michael Dom[2] and Somnath Sikdar[3]

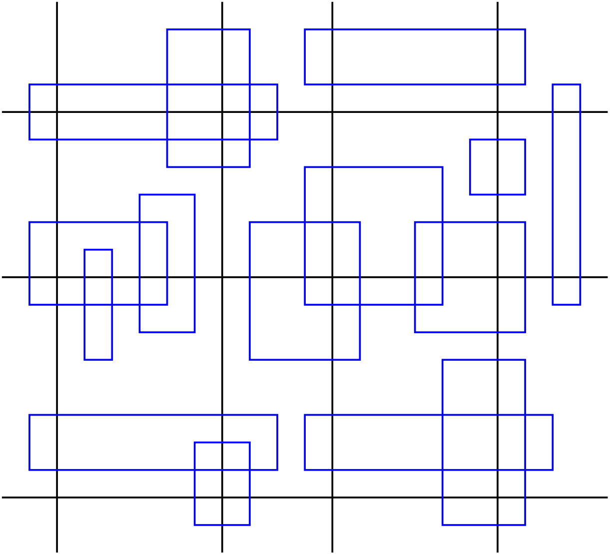
[2]Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany

[3]The Institute of Mathematical Sciences, Chennai, India

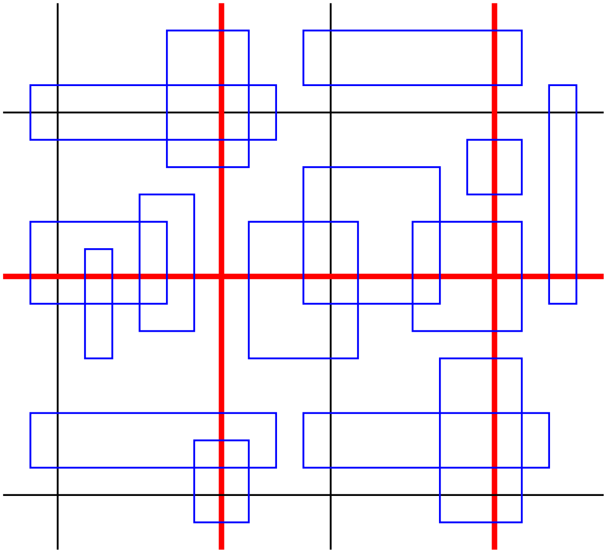
FAW 2008

[1]Supported by the DAAD-DST exchange program D/05/57666.

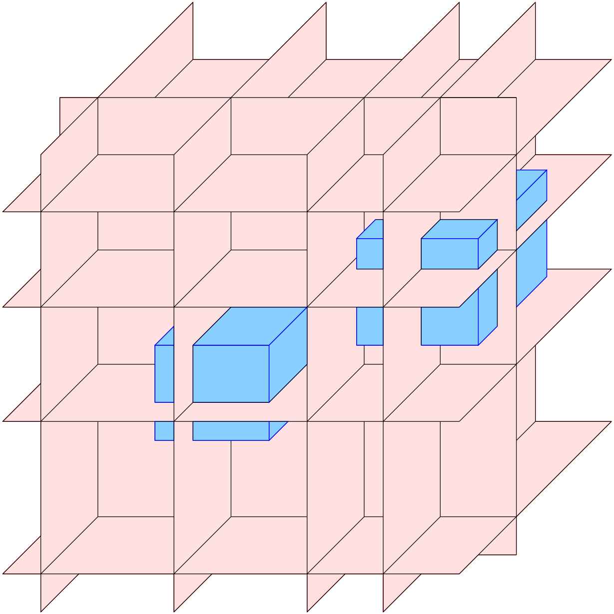
(2-Dimensional) Rectangle Stabbing



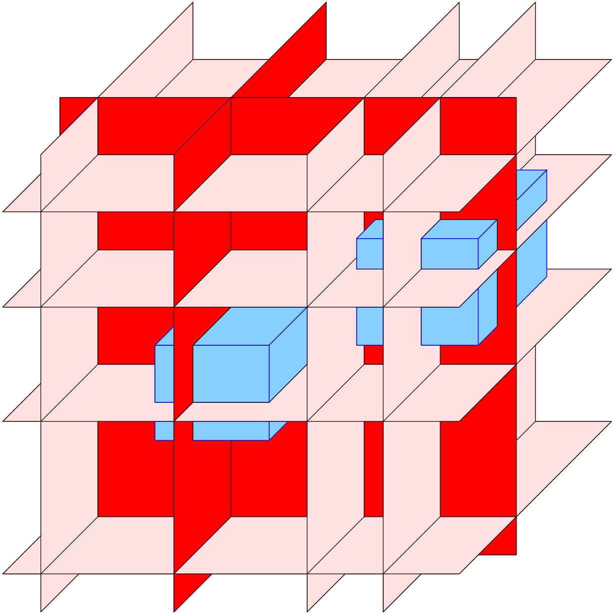
(2-Dimensional) Rectangle Stabbing



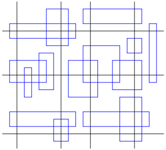
3-Dimensional Rectangle Stabbing

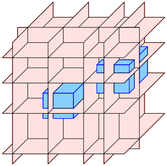


3-Dimensional Rectangle Stabbing



d-Dimensional Rectangle Stabbing



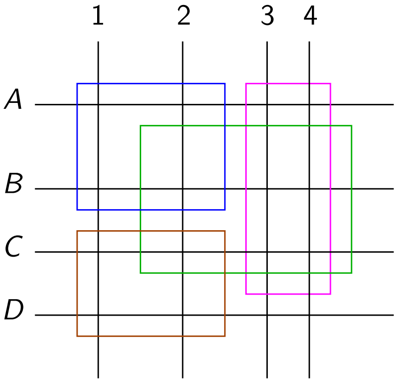


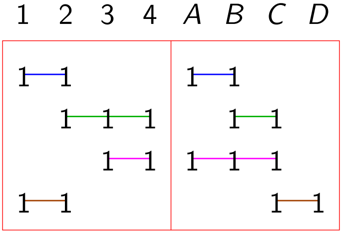
d-Dimensional Rectangle Stabbing

Input: A set R of axis-parallel d-dimensional hyperrectangles, a set L of axis-parallel $(d-1)$-dimensional hyperplanes, a positive integer k.

Question: Exists $L' \subseteq L$ with $|L'| \leq k$ such that every hyperrectangle from R is intersected by at least one hyperplane from L'?

d-Dimensional Rectangle Stabbing and Set Cover



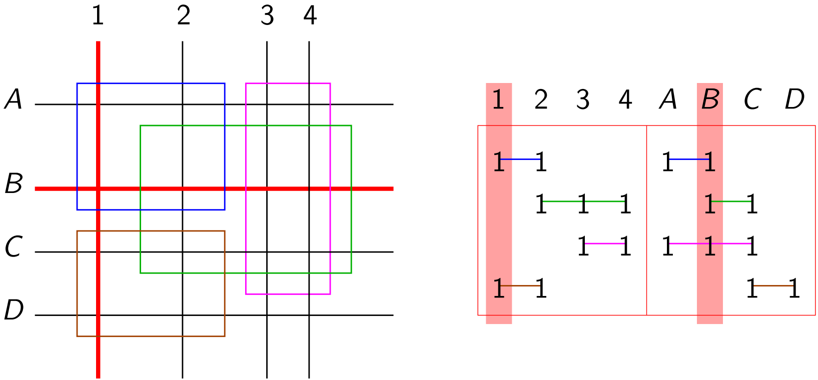


Set Cover

Input: A binary matrix M, a positive integer k.

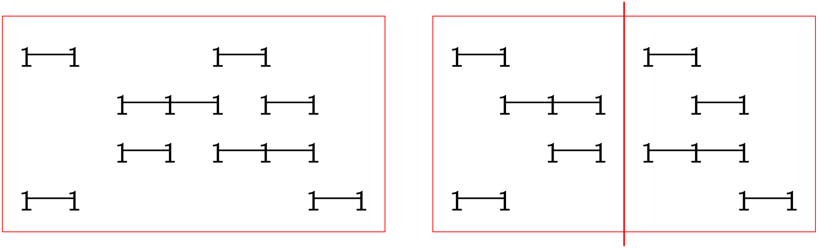
Question: Is there a set of at most k columns that hits a 1 in every row?

d-Dimensional Rectangle Stabbing and Set Cover



Set Cover
Input: A binary matrix M, a positive integer k.
Question: Is there a set of at most k columns that hits a 1 in every row?

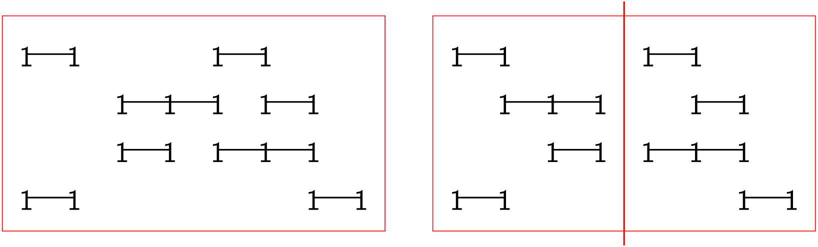
d-Consecutive-Ones Property (d-C1P) and Separated d-Consecutive-Ones Property (d-XC1P)



2-C1P:
two blocks per row

2-XC1P:
dividing line, one block per section

Restricted Variants of Set Cover



2-C1P 2-XC1P

d-C1P-Set Cover: Input Matrix has *d*-C1P.
d-XC1P-Set Cover: Input Matrix has *d*-XC1P.

d-XC1P-Set Cover and *d*-Dimensional Rectangle Stabbing are equivalent. Both are NP-complete for $d = 2$.

Known Results

- Factor-$d2^{d-1}$ approximation for d-Dimensional Rectangle Stabbing when all hyperrectangles are identical
  [Hassin and Megiddo, *Discrete Appl. Math., '91*]
- Factor-d approximation for d-Dimensional Rectangle Stabbing
  [Gaur et al., *J. Algorithms, '02*]
- Factor-d approximation for d-C1P-Set Cover
  [Mecke et al., *ATMOS '05*]
- Approximation algorithms for 2-Dimensional Rectangle Stabbing when every rectangle has height or width one
  [Hassin and Megiddo, *Discrete Appl. Math., '91*,
  Kovaleva and Spieksma, *ISAAC '01*, *SIAM J. Discrete Math., '06*]
- 3-C1P-Set Cover is W[1]-hard
  [Fellows, personal communication, 2007]
- Open: Parameterized complexity of d-Dimensional Rectangle Stabbing

Parameterized Complexity

- Main idea: Measure complexity not only in input size, but also in an additional "parameter" k.

- *Fixed parameter tractable (FPT)*
  with respect to a parameter k
  $\Leftrightarrow$ running time $f(k) \cdot n^{O(1)}$.

  Example: $O(2^k \cdot n)$
  Not FPT: $O(n^k)$

- W[1]-hardness: Basic concept for parameterized intractability.

Our Results

Hardness:

- 3-XC1P-Set Cover (and 3-Dimensional Rectangle Stabbing) is W[1]-hard

Algorithms:

- $O((bk)^k \cdot n^{O(1)})$ time algorithm for 2-Dimensional Rectangle Stabbing when every rectangle has height or width b
- Problem kernel for 2-Dimensional Rectangle Stabbing when every horizontal line intersects at most b rectangles

Our Results

Hardness:

- **3-XC1P-Set Cover (and 3-Dimensional Rectangle Stabbing) is** W[1]**-hard**

Algorithms:

- $O((bk)^k \cdot n^{O(1)})$ time algorithm for 2-Dimensional Rectangle Stabbing when every rectangle has height or width b
- Problem kernel for 2-Dimensional Rectangle Stabbing when every horizontal line intersects at most b rectangles

3-XC1P-Set Cover is W[1]-hard

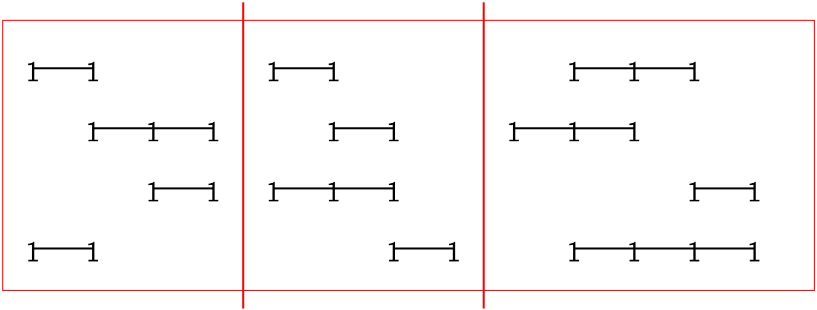
Parameterized reduction:

- Same basic idea as polynomial-time reduction: Reduce from a hard problem.

$$(x, k) \rightsquigarrow (x', k')$$

- New parameter must depend only on the old parameter: $k' = f(k)$.
- (Reduction may cost $f(k) \cdot n^{O(1)}$ time.)

3-XC1P-Set Cover is W[1]-hard



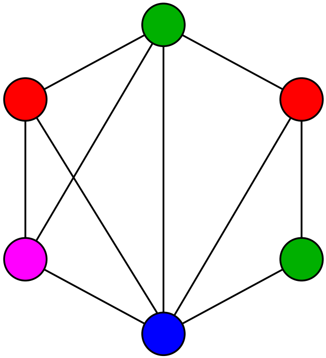
3-XC1P-Set Cover

Input: A binary matrix M with 3-XC1P, a positive integer k.

Question: Is there a set of at most k columns that hits a 1 in every row?

3-XC1P-Set Cover is W[1]-hard

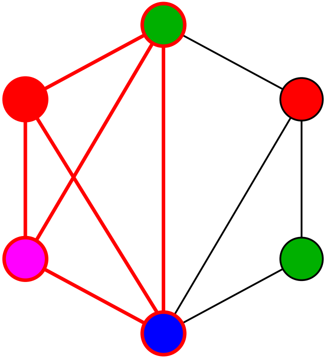
Reduction from the W[1]-hard problem Multicolored Clique.



Multicolored Clique
Input: A k-colored undirected graph and a positive integer k.
Question: Is there a clique of size k?

3-XC1P-Set Cover is W[1]-hard

Reduction from the W[1]-hard problem Multicolored Clique.

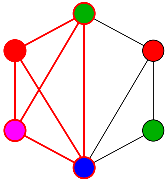


Multicolored Clique
Input: A k-colored undirected graph and a positive integer k.
Question: Is there a clique of size k?

3-XC1P-Set Cover is W[1]-hard

Reformulation of Multicolored Clique:

[Fellows et al., manuscript, 2008]



Question: Is there a set E' of $\binom{k}{2}$ edges and a set V' of k vertices such that

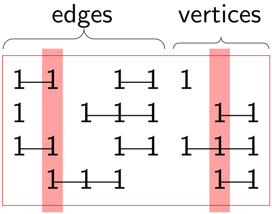
- E' contains an edge of every "edge color",
- V' contains a vertex of every color, and
- $\{v, w\} \in E' \rightarrow v, w \in V'$?

3-XC1P-Set Cover is W[1]-hard

Question: Is there a set E' of $\binom{k}{2}$ edges and a set V' of k vertices such that

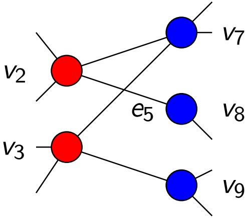
- E' contains an edge of every "edge color",
- V' contains a vertex of every color, and
- $\{v, w\} \in E' \rightarrow v, w \in V'$?

Approach for the reduction to 3-XC1P-Set Cover:



- one column for every edge and every vertex
- number of columns to select: $\binom{k}{2} + k$
- rows to enforce the three constraints

3-XC1P-Set Cover is W[1]-hard

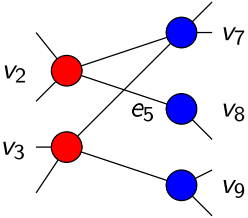


- E' contains an edge of every "edge color",
- V' contains a vertex of every color, and
- $\{v, w\} \in E' \rightarrow v, w \in V'$?

...	{red, blue}				...	...	red		...	blue			...
...	e_4	e_5	e_6	e_7	...	...	v_2	v_3	...	v_7	v_8	v_9	...
	1	1	1	1									
							1	1					
										1	1	1	
	1		1	1			1						
	1		1	1							1		

Number of columns to select: $\binom{k}{2} + k$.

3-XC1P-Set Cover is W[1]-hard



- E' contains an edge of every "edge color",
- V' contains a vertex of every color, and
- $\{v, w\} \in E' \;\rightarrow\; v, w \in V'$?

. . .	{red, blue}				. . .	. . .	red		. . .	blue			. . .
. . .	e_4	e_5	e_6	e_7	. . .	. . .	v_2	v_3	. . .	v_7	v_8	v_9	. . .
	1	1	1	1									
							1	1					
										1	1	1	
	1		1	1			1						
	1		1	1							1		

Number of columns to select: $\binom{k}{2} + k$.

3-XC1P-Set Cover is W[1]-hard

...	{red, blue}				...			...	...	red		...	blue			...
...	e_4	e_5	e_6	e_7	...			...	...	v_2	v_3	...	v_7	v_8	v_9	...
	1	1	1	1												
					1	1	1									
										1	1					
													1	1	1	
		1	1	1						1						
		1	1	1									1			
	1		1	1						1						
	1		1	1										1		
	1	1		1							1					
	1	1		1									1			
	1	1	1								1					
	1	1	1												1	
						1	1									
						1	1									
					1		1		...			...				...
					1		1									
					1	1										
					1	1										

3-XC1P-Set Cover is W[1]-hard

...	{red, blue}				...	...	{red, blue}				...	...	red		...	blue			...
...	e_4	e_5	e_6	e_7	...	...	e_4	e_5	e_6	e_7	...	...	v_2	v_3	...	v_7	v_8	v_9	...
	1	1	1	1															
							1	1	1	1									
													1	1					
																1	1	1	
	1							1	1	1									
	1	1							1	1									
	1	1	1							1									
		1	1	1			1												
			1	1			1	1											
				1			1	1	1										
	1								1	1			1						
	1								1	1							1		

Number of columns to select: $2 \cdot \binom{k}{2} + k$.

3-XC1P-Set Cover is W[1]-hard

...	{red, blue}				...	...	{red, blue}				...	...	red		...	blue			...
...	e_4	e_5	e_6	e_7	...	...	e_4	e_5	e_6	e_7	...	...	v_2	v_3	...	v_7	v_8	v_9	...
	1	1	1	1															
							1	1	1	1									
													1	1					
																1	1	1	
	1							1	1	1									
	1	1							1	1									
	1	1	1							1									
		1	1	1			1												
			1	1			1	1											
				1			1	1	1										
	1								1	1			1						
	1								1	1							1		

Number of columns to select: $2 \cdot \binom{k}{2} + k$.

3-XC1P-Set Cover is W[1]-hard

...	{red, blue}				...	...	{red, blue}				...	...	red		...	blue			...
...	e_4	e_5	e_6	e_7	...	...	e_4	e_5	e_6	e_7	...	...	v_2	v_3	...	v_7	v_8	v_9	...
	1	1	1	1															
							1	1	1	1									
													1	1					
																1	1	1	
	1							1	1	1									
	1	1							1	1									
	1	1	1							1									
		1	1	1			1												
			1	1			1	1											
				1			1	1	1										
	1								1	1			1						
	1								1	1							1		

Number of columns to select: $2 \cdot \binom{k}{2} + k$.

3-XC1P-Set Cover is W[1]-hard

...	{red, blue}				...	...	{red, blue}				...	...	red		...	blue			...
...	e_4	e_5	e_6	e_7	...	...	e_4	e_5	e_6	e_7	...	...	v_2	v_3	...	v_7	v_8	v_9	...
	1	1	1	1															
							1	1	1	1									
													1	1					
																1	1	1	
	1							1	1	1									
	1	1							1	1									
	1	1	1							1									
		1	1	1			1												
			1	1			1	1											
				1			1	1	1										
	1								1	1			1						
	1								1	1							1		

Number of columns to select: $2 \cdot \binom{k}{2} + k$.

3-XC1P-Set Cover is W[1]-hard

…	{red, blue}				…	…	{red, blue}				…	…	red		…	blue			…
…	e_4	e_5	e_6	e_7	…	…	e_4	e_5	e_6	e_7	…	…	v_2	v_3	…	v_7	v_8	v_9	…
	1	1	1	1															
							1	1	1	1									
													1	1					
																1	1	1	
	1							1	1	1									
	1	1							1	1									
	1	1	1							1									
		1	1	1			1												
			1	1			1	1											
				1			1	1	1										
	1								1	1			1						
	1								1	1							1		

Number of columns to select: $2 \cdot \binom{k}{2} + k$.

3-XC1P-Set Cover is W[1]-hard

…	{red, blue}				…	…	{red, blue}				…	…	red		…	blue			…
…	e_4	e_5	e_6	e_7	…	…	e_4	e_5	e_6	e_7	…	…	v_2	v_3	…	v_7	v_8	v_9	…
	1	1	1	1															
							1	1	1	1									
													1	1					
																1	1	1	
	1							1	1	1									
	1	1							1	1									
	1	1	1							1									
		1	1	1			1												
			1	1			1	1											
				1			1	1	1										
	1								1	1			1						
	1								1	1							1		

Number of columns to select: $2 \cdot \binom{k}{2} + k$.

3-XC1P-Set Cover is W[1]-hard

Theorem

For every $d \geq 3$, d-Dimensional Rectangle Stabbing is W[1]*-hard with respect to the parameter k.*

With similar reductions:

Theorem

For every $d \geq 3$, d-Dimensional Rectangle Stabbing with hypercubes is W[1]*-hard.*

Theorem

For every $d \geq 3$, the variation of d-Dimensional Rectangle Stabbing with lines instead of hyperplanes is W[1]*-hard.*

Our Results

Hardness:

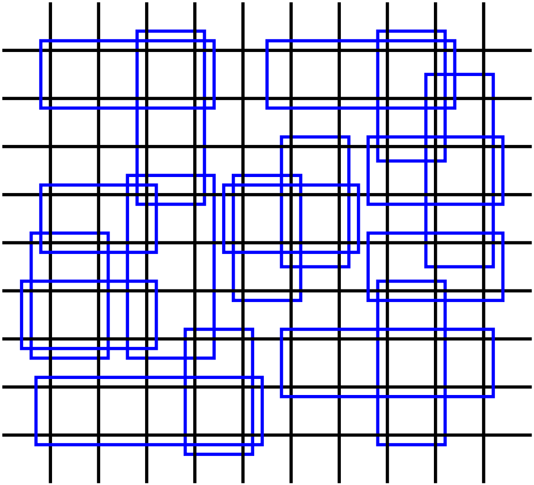
- 3-XC1P-Set Cover (and 3-Dimensional Rectangle Stabbing) is W[1]-hard

Algorithms:

- $O((bk)^k \cdot n^{O(1)})$ **time algorithm for 2-Dimensional Rectangle Stabbing when every rectangle has height or width** b
- Problem kernel for 2-Dimensional Rectangle Stabbing when every horizontal line intersects at most b rectangles

FPT Algorithm for Rectangles of Bounded Width or Height

Restriction: Every rectangle has width b or height b.

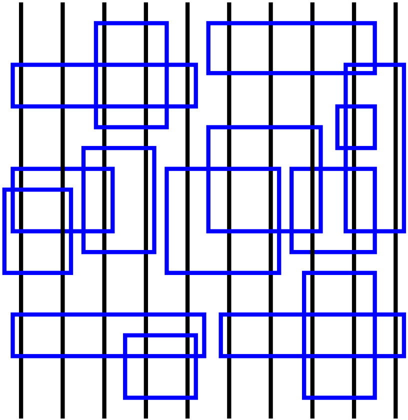


$k = 5$

FPT Algorithm for Rectangles of Bounded Width or Height

Subroutine: Polynomial time algorithm for one-directional lines.

Restriction here: The input contains only vertical lines.

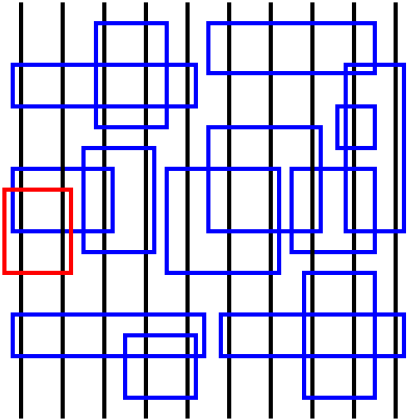


Identical to Clique Cover on Interval Graphs.

FPT Algorithm for Rectangles of Bounded Width or Height

Subroutine: Polynomial time algorithm for one-directional lines.

Restriction here: The input contains only vertical lines.

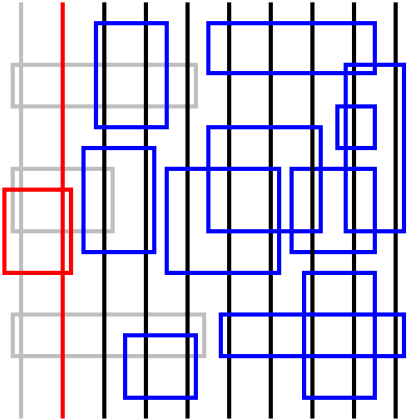


Identical to Clique Cover on Interval Graphs.

FPT Algorithm for Rectangles of Bounded Width or Height

Subroutine: Polynomial time algorithm for one-directional lines.

Restriction here: The input contains only vertical lines.

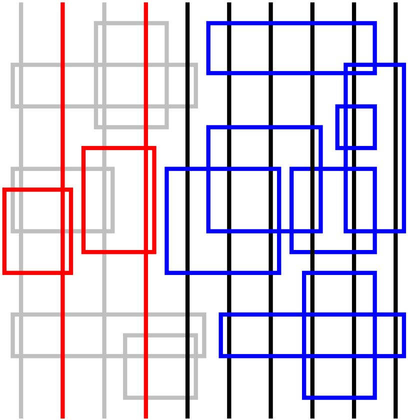


Identical to Clique Cover on Interval Graphs.

FPT Algorithm for Rectangles of Bounded Width or Height

Subroutine: Polynomial time algorithm for one-directional lines.

Restriction here: The input contains only vertical lines.

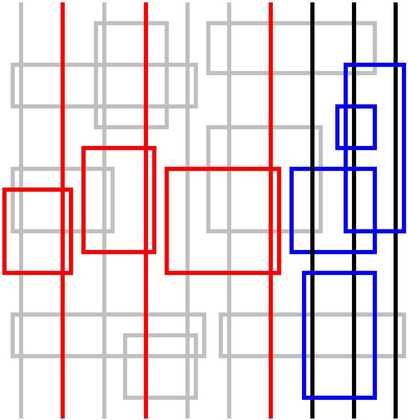


Identical to Clique Cover on Interval Graphs.

FPT Algorithm for Rectangles of Bounded Width or Height

Subroutine: Polynomial time algorithm for one-directional lines.

Restriction here: The input contains only vertical lines.

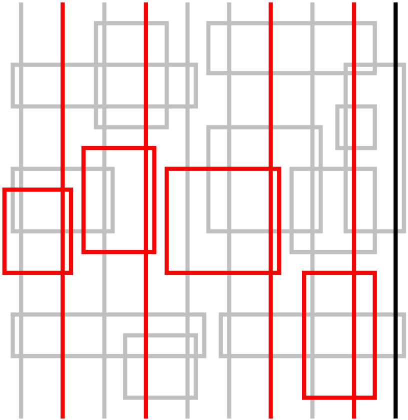


Identical to Clique Cover on Interval Graphs.

FPT Algorithm for Rectangles of Bounded Width or Height

Subroutine: Polynomial time algorithm for one-directional lines.

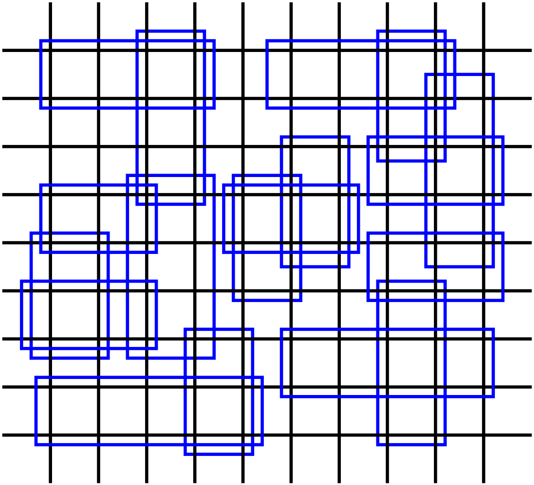
Restriction here: The input contains only vertical lines.



Identical to Clique Cover on Interval Graphs.

FPT Algorithm for Rectangles of Bounded Width or Height

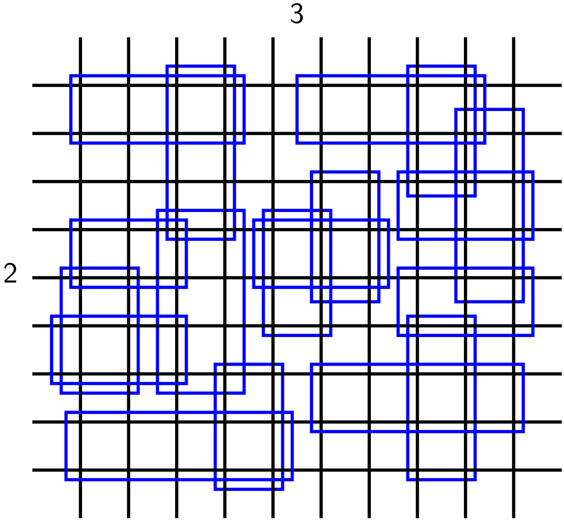
Restriction: Every rectangle has width b or height b.



$k = 5$

FPT Algorithm for Rectangles of Bounded Width or Height

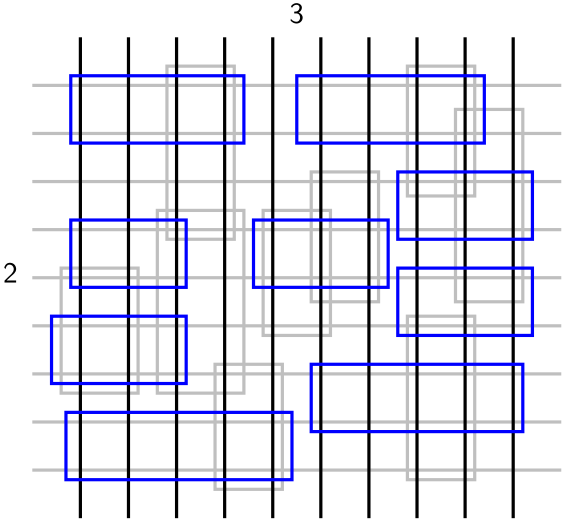
Restriction: Every rectangle has width b or height b.



$k = 5$

FPT Algorithm for Rectangles of Bounded Width or Height

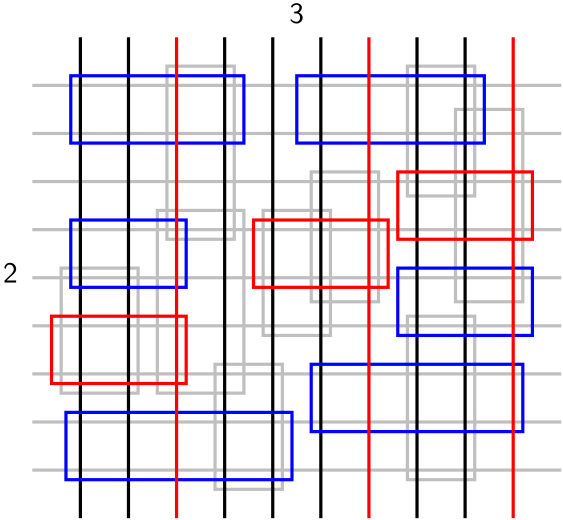
Restriction: Every rectangle has width b or height b.



$k = 5$

FPT Algorithm for Rectangles of Bounded Width or Height

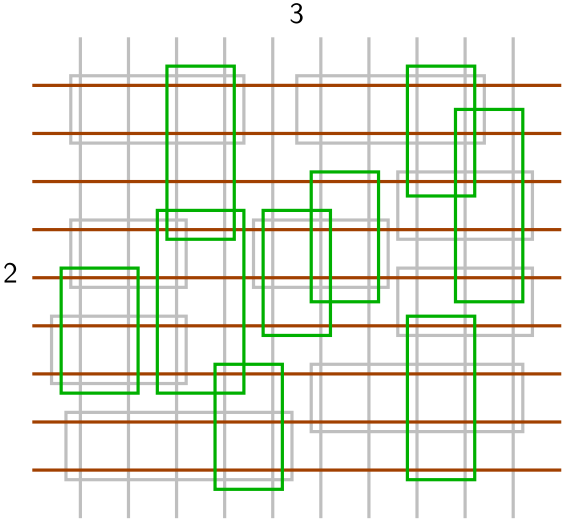
Restriction: Every rectangle has width b or height b.



$k = 5$

FPT Algorithm for Rectangles of Bounded Width or Height

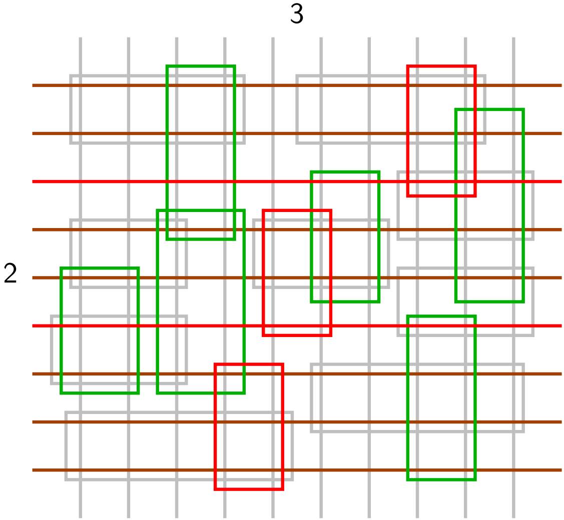
Restriction: Every rectangle has width b or height b.



$k = 5$

FPT Algorithm for Rectangles of Bounded Width or Height

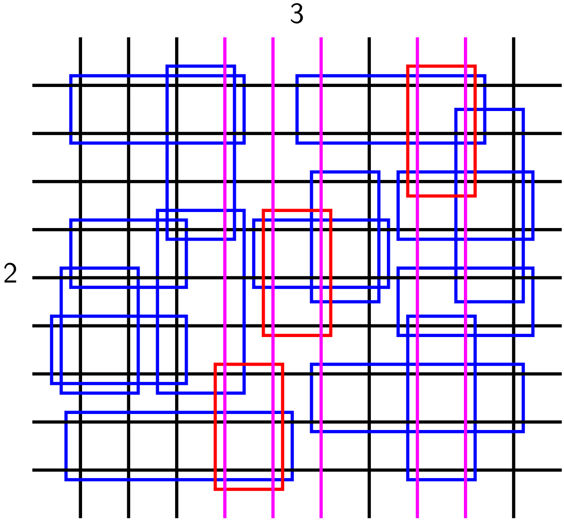
Restriction: Every rectangle has width b or height b.



$k = 5$

FPT Algorithm for Rectangles of Bounded Width or Height

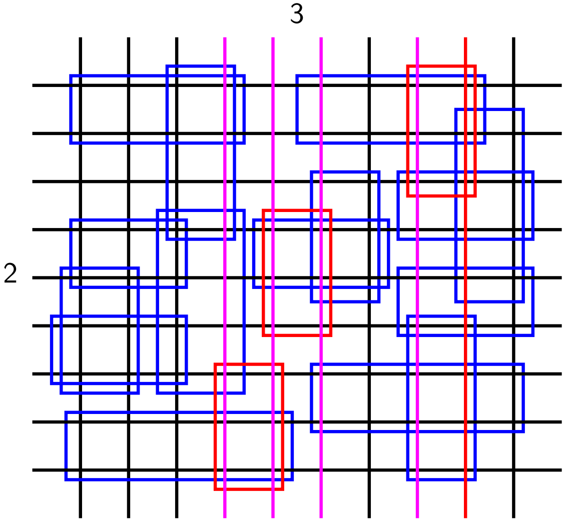
Restriction: Every rectangle has width b or height b.



$k = 5$

FPT Algorithm for Rectangles of Bounded Width or Height

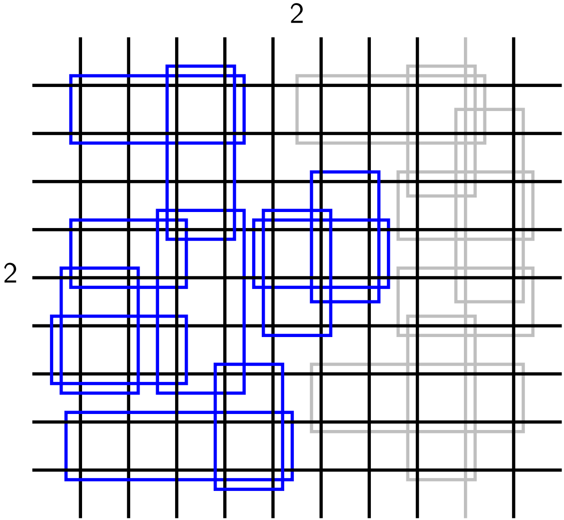
Restriction: Every rectangle has width b or height b.



$k = 5$

FPT Algorithm for Rectangles of Bounded Width or Height

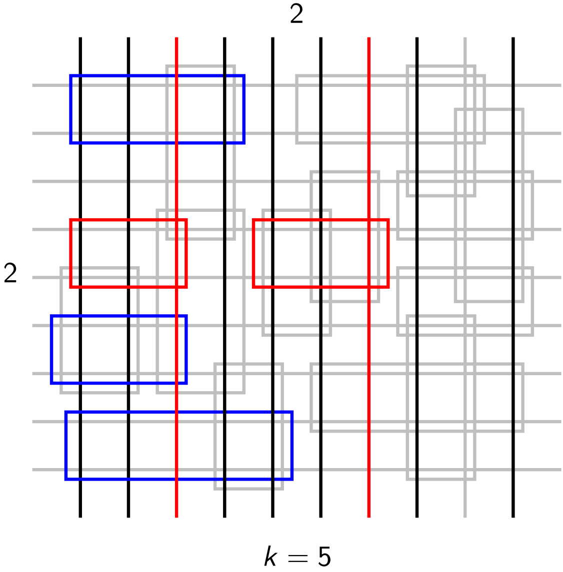
Restriction: Every rectangle has width b or height b.



$k = 5$

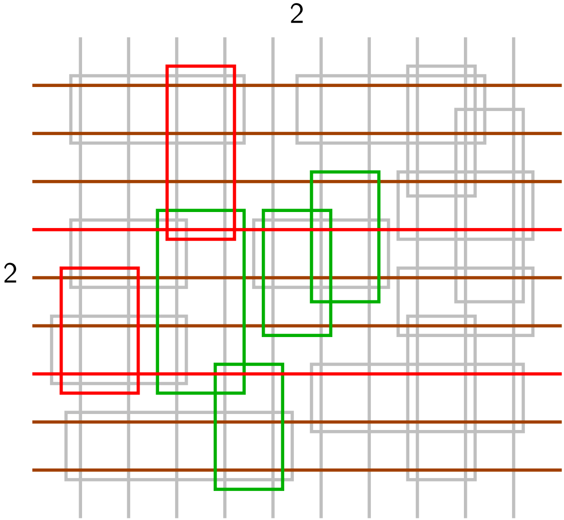
FPT Algorithm for Rectangles of Bounded Width or Height

Restriction: Every rectangle has width b or height b.



FPT Algorithm for Rectangles of Bounded Width or Height

Restriction: Every rectangle has width b or height b.



$k = 5$

FPT Algorithm for Rectangles of Bounded Width or Height

Theorem

2-Dimensional Rectangle Stabbing can be solved in $O((bk)^k \cdot n^{O(1)})$ time when the width or the height of every rectangle in R is bounded from above by b.

Our Results

Hardness:

- 3-XC1P-Set Cover (and 3-Dimensional Rectangle Stabbing) is W[1]-hard

Algorithms:

- $O((bk)^k \cdot n^{O(1)})$ time algorithm for 2-Dimensional Rectangle Stabbing when every rectangle has height or width b
- **Problem kernel for 2-Dimensional Rectangle Stabbing when every horizontal line intersects at most b rectangles**

Problem Kernel for Horizontal Lines with Few Intersections

Problem kernel:

- Main idea: Polynomial-time data reduction leads to a small problem instance.

$$(x, k) \rightsquigarrow (x', k')$$

- $|x'| \leq f(k)$
- $|k'| \leq k$
- (x', k') can be computed in polynomial time.

Problem Kernel for Horizontal Lines with Few Intersections

Restriction: Every horizontal line intersects at most b rectangles.

- Polynomial time solvable for $b = 1$.
- NP-hard for $b \geq 2$.
- Easy size-$O(k^k)$ search tree for $b = 2$.
- Difficult for $b \geq 3 \ldots$

Observation

For every b, to intersect all rectangles with vertical lines only, we need at most bk vertical lines.

Problem Kernel for Horizontal Lines with Few Intersections

Some Reduction Rules:

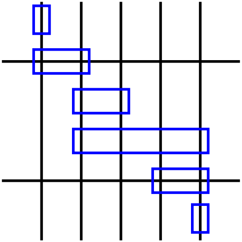
1. Delete “dominated” lines.
2. Delete “dominated” rectangles.

Problem Kernel for Horizontal Lines with Few Intersections

Some Reduction Rules:

1. Delete "dominated" lines.
2. Delete "dominated" rectangles.

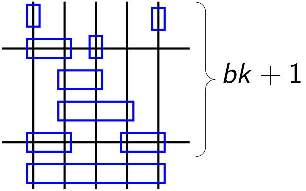
⇒ For every vertical line v there is a rectangle r_1 such that v is the leftmost line intersecting r_1, and a rectangle r_2 such that v is the rightmost line intersecting r_2.



Problem Kernel for Horizontal Lines with Few Intersections

Additional reduction rules:

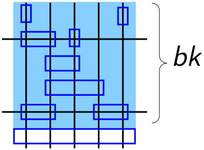
3. If there are $bk + 2$ rectangles $r_1, \ldots, r_{bk+2} \in R$ such that for each $i \in \{1, \ldots, bk+1\}$ it holds that every vertical line that intersects r_i also intersects r_{bk+2}, then delete r_{bk+2}.



Problem Kernel for Horizontal Lines with Few Intersections

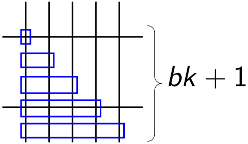
Observation

For every rectangle r in a reduced instance there are at most bk rectangles r' contained "in the vertical space" of r.



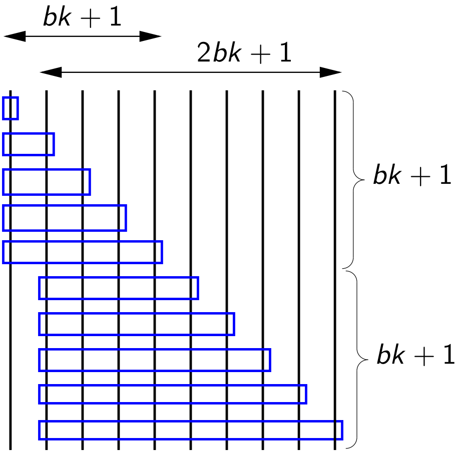
Observation

For every vertical line v there are at most $bk + 1$ rectangles "starting" at v.



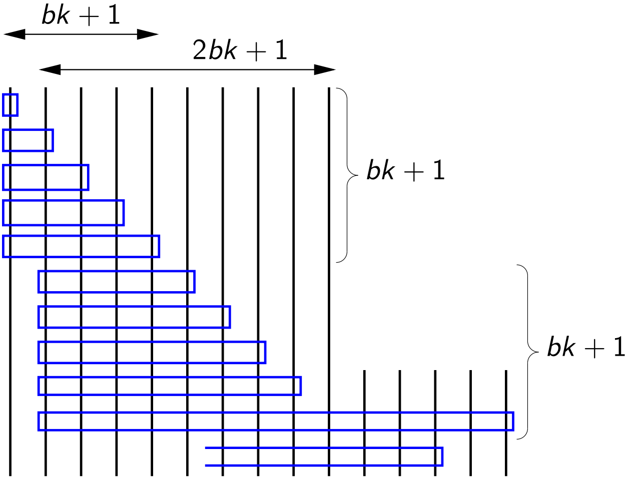
Problem Kernel for Horizontal Lines with Few Intersections

⇒ The possible width of a rectangle depends on the position of its left end: $w(r) \leq l(r) \cdot bk + 1$



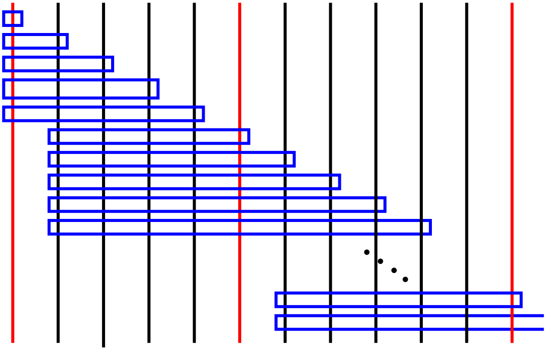
Problem Kernel for Horizontal Lines with Few Intersections

$\Rightarrow$ The possible width of a rectangle depends on the position of its left end: $w(r) \leq l(r) \cdot bk + 1$

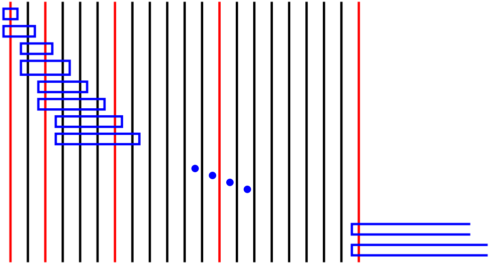


Problem Kernel for Horizontal Lines with Few Intersections

- The possible width of a rectangle depends on the position of its left end.
- We know: For every b, to intersect all rectangles with vertical lines only, we need at most bk vertical lines.
- $\Rightarrow$ The ith vertical line in such a set of bk vertical lines is at position at most $((bk+1)^i - 1)/bk$.



Problem Kernel for Horizontal Lines with Few Intersections



Theorem

2-Dimensional Rectangle Stabbing has a kernel of size $O((bk+1)^{bk})$ when every horizontal line intersects at most b rectangles.

Open Questions

- ~~Parameterized complexity of 2-Dimensional Rectangle Stabbing?~~
- Parameterized complexity of 2-Dimensional Rectangle Stabbing when no two rectangles overlap?
- Faster FPT algorithms for the restrictions considered here?
- Weighted versions of the restrictions considered here?