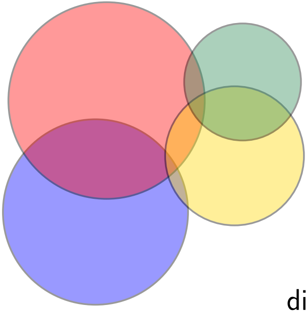
Maximum Matchings in Geometric Intersection Graphs

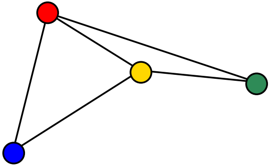
Édouard Bonnet
ENS Lyon, LIP

Sergio Cabello
U. of Ljubljana & IMFM, Slovenia

Wolfgang Mulzer
FU Berlin, Germany

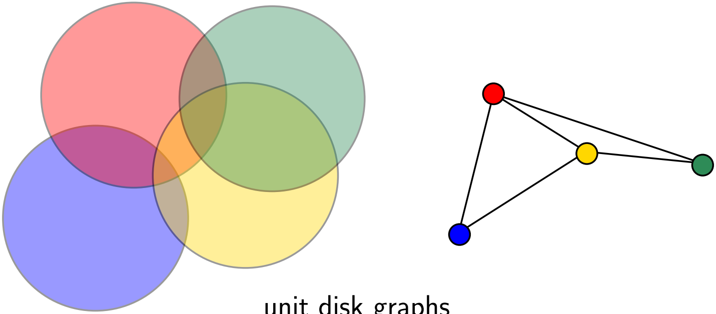
Disk intersection graphs





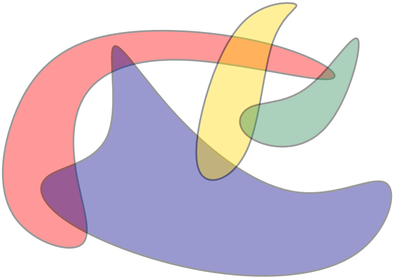
disk graphs

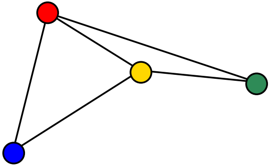
Unit disk intersection graphs



unit disk graphs

Fat object intersection graphs





Not necessarily pseudo-disks, but constant description complexity

Maximum matching for geometric intersection graphs

- Build the graph explicitly, run max matching algorithm
  - $O(\sqrt{n}m)$ [Micali and Vazirani '80]
  - $O(n^\omega)$ [Mucha and Sankowski '04]

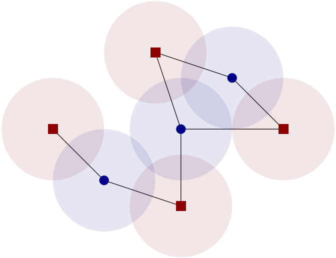
Maximum matching for geometric intersection graphs

- Build the graph explicitly, run max matching algorithm
  - $O(\sqrt{n}m)$ [Micali and Vazirani '80]
  - $O(n^\omega)$ [Mucha and Sankowski '04]

- Use geometry to speed up
  - Use geometric data structures ($\longleftarrow$ previous work)
  - Use the structure of the graph ($\longleftarrow$ our focus)

Previous works

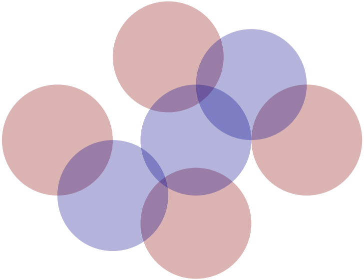
- maximum matching bicolored (unit) disk graph in $\tilde{O}(n^{3/2})$ time [Efrat, Itai, Katz '01]



- improving the $\tilde{O}(n^{3/2})$ for bicolored case would be great; remains open, even for unit squares
- max matching (unit) disk graph – nothing specialized known

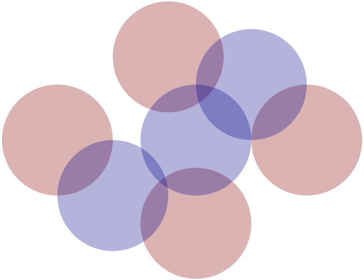
Our work I - Low density case

- each point of the plane covered by $\leqslant \rho$ disks $\Rightarrow O(\rho n)$ edges
- $\rho = O(1)$ particularly relevant



Our work I - Low density case

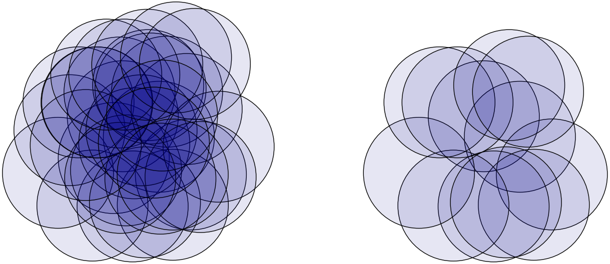
- each point of the plane covered by $\leqslant \rho$ disks $\Rightarrow O(\rho n)$ edges
- $\rho = O(1)$ particularly relevant



- maximum matching in $O(\rho^{3\omega/2} n^{\omega/2}) = O(\rho^{3.56} n^{1.19})$
  - improvement when $\rho = O(n^{0.113})$
- bicolored, unicolored; actually any given subgraph
- works for *fat shapes* or *low-density* scenarios

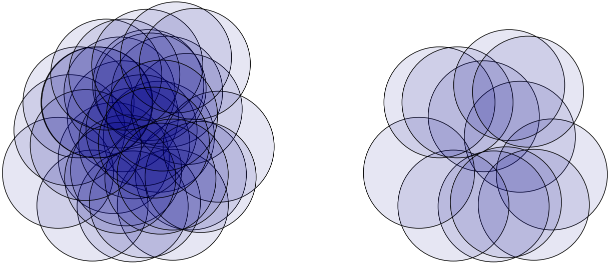
Our work II - Sparsification

- for unit disks, maximum matching reduces to the case of bounded depth



Our work II - Sparsification

- for unit disks, maximum matching reduces to the case of bounded depth



- $O(n \operatorname{polylog} n)$ time reduction
- using semi-dynamic data structures for nearest neighbors
- only works for unicolored *full* intersection graph
- works for fat shapes of comparable sizes

Our work III - Putting things together

- A maximum matching in unit disk graphs in time $O(n^{\omega/2})$ whp

Our work III - Putting things together

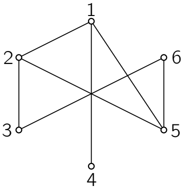
- A maximum matching in unit disk graphs in time $O(n^{\omega/2})$ whp

Same ideas (sparsification + bounded depth case), other data structures

- disks of radius in $[1, \Psi]$ in time $O(\Psi^6 n \operatorname{polylog} n + \Psi^{12\omega} n^{\omega/2})$ whp
- translates of $O(1)$ convex shapes in $\mathbb{R}^2$ in time $O(n^{\omega/2})$ whp

Low density case – Outline

- adapt [Mucha, Sankowski '06] for planar graphs in $O(n^{\omega/2})$
- adapt [Yuster, Zwick '07] for H-minor-free $O(n^{3\omega/(\omega+3)})$
- rank of an "incidence" matrix [Lovász '79, Rabin, Vazirani '89]
- Gaussian elimination, nested dissection [Lipton, Rose, Tarjan '79]
- geometric small balanced separators [Har-Peled, Quanrud '17]
- get explicit dependency on ρ.



$A =$

	10		−7	16	
−10		14		3	
	−14				0
7					
−16	−3				−14
		0		14	

The algebraic approach to Max Matching (1)

Skew-symmetric matrix or *Tutte matrix*:

$$\tilde{A}(G)_{i,j} = \begin{cases} x_{i,j} & \text{if } v_i v_j \in E(G) \text{ and } i < j \\ -x_{i,j} & \text{if } v_i v_j \in E(G) \text{ and } i > j \\ 0 & \text{if } v_i v_j \notin E(G) \end{cases}$$

Theorem (Tutte '47)

The determinant of $\tilde{A}(G)$ is non-zero iff G admits a perfect matching.

The algebraic approach to Max Matching (1)

Theorem (Tutte '47)

The determinant of $\tilde{A}(G)$ is non-zero iff G admits a perfect matching.

Theorem (Lovász '79)

The rank of $\tilde{A}(G)$ is twice the size of a maximum matching in G.

$A(G) =$ replace every entry $x_{i,j}$ of $\tilde{A}(G)$ by a random element of $\mathbb{Z}_p$

By DeMillo-Lipton-Zippel-Schwartz lemma,
$det(A(G)) \neq 0 \Leftrightarrow det(\tilde{A}(G)) \neq 0$ wp $\geqslant 1 - n/p$

$\rightarrow$ randomized $O(n^{\omega})$-algorithm to decide a perfect matching

The algebraic approach to Max Matching (2)

Lovász generalizes it to computing the **size** of a max matching in randomized $O(n^\omega)$.

How actually computing a perfect matching? a maximum matching?

The algebraic approach to Max Matching (2)

Lovász generalizes it to computing the **size**
of a max matching in randomized $O(n^\omega)$.

How actually computing a perfect matching? a maximum matching?

Theorem (Rabin and Vazirani '89)

Reduction of maximum matching to perfect matching in $O(n^\omega)$, whp

Gaussian elimination without pivoting, remove indices with diagonal 0

The algebraic approach to Max Matching (2)

Lovász generalizes it to computing the **size** of a max matching in randomized $O(n^\omega)$.

How actually computing a perfect matching?

Theorem (Rabin and Vazirani '89)

$A^{-1}_{i,j} \neq 0 \Leftrightarrow G - \{v_i, v_j\}$ *has a perfect matching, whp*

If $v_i v_j \in E(G)$, it is *allowed* as being extendible to a perfect matching

Theorem (Mucha and Sankowski '04)

Find a maximum matching in $O(n^\omega)$*, whp*

Batching *allowed edges* and *lazy elimination*

Small balanced separators

α-balanced separator: S such that $G - S$ is disconnected and all connected components have size at most $\alpha|V(G)|$

- Algorithmic revolution of the 80's and 90's: approximation, faster exact algorithms
- Tied to treewidth

Small balanced separators

α-balanced separator: S such that $G - S$ is disconnected and all connected components have size at most $\alpha|V(G)|$

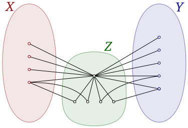
- Algorithmic revolution of the 80's and 90's: approximation, faster exact algorithms
- Tied to treewidth
- example: planar graphs, H-minor free graphs, etc.

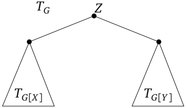
Small balanced separators

α-balanced separator: S such that $G - S$ is disconnected and all connected components have size at most $\alpha|V(G)|$

- Algorithmic revolution of the 80's and 90's: approximation, faster exact algorithms
- Tied to treewidth
- example: planar graphs, H-minor free graphs, etc.

(γ, β, α)-separator tree: T_G rooted at an α-balanced separator Z of size $\leqslant \gamma|V(G)|^\beta$, sep. X from Y, with children $T_{G[X]}$ and $T_{G[Y]}$





Nested dissection

Gaussian elimination on a sparse matrix

$$\begin{pmatrix} d_1 & v_1^T \\ v_1 & B_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ v_1/d_1 & I_{n-1} \end{pmatrix} \begin{pmatrix} d_1 & 0 \\ 0 & B_1 - {\color{red} v_1 v_1^T / d_1} \end{pmatrix} \begin{pmatrix} 1 & v_1^T/d_1 \\ 0 & I_{n-1} \end{pmatrix}$$

Interpreting the matrix as a graph with an edge *ij* iff $A_{i,j} \neq 0$,
the red term brings new non-zero entries called *fill-in*

Nested dissection

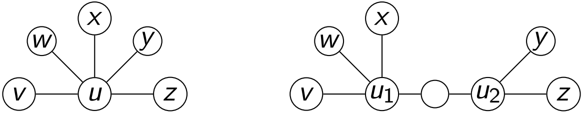
Gaussian elimination on a sparse matrix

$$\begin{pmatrix} d_1 & v_1^T \\ v_1 & B_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ v_1/d_1 & I_{n-1} \end{pmatrix} \begin{pmatrix} d_1 & 0 \\ 0 & B_1 - v_1 v_1^T/d_1 \end{pmatrix} \begin{pmatrix} 1 & v_1^T/d_1 \\ 0 & I_{n-1} \end{pmatrix}$$

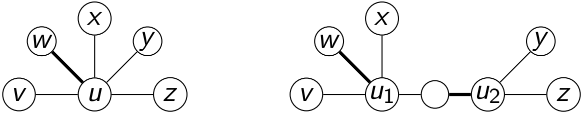
Interpreting the matrix as a graph with an edge ij iff $A_{i,j} \neq 0$,
the red term brings new non-zero entries called *fill-in*

A post-order traversal of a separator-tree gives a low fill-in

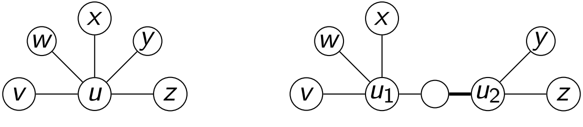
Vertex splits preserve max matchings



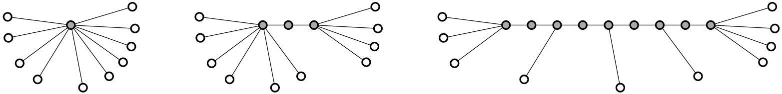
Vertex splits preserve max matchings



Vertex splits preserve max matchings



In both cases, the max matching size increases by exactly 1



Reduction to constant degree

Processing the input (1)

Lemma

The edges of n objects with density ρ in $\mathbb{R}^2$ are $O(\rho n)$ many, and can be computed in $O(\rho n \log n)$.

Plane sweep for boundary intersections + trapezoidal decomposition for inclusions

Processing the input (1)

Lemma

The edges of n objects with density ρ in $\mathbb{R}^2$ are $O(\rho n)$ many, and can be computed in $O(\rho n \log n)$.

Plane sweep for boundary intersections + trapezoidal decomposition for inclusions

Lemma (Smith and Wormald '98, Har-Peled and Quanrud '17)

A set of n objects with density ρ in $\mathbb{R}^2$ admits an α-balanced separator of size $O(\sqrt{\rho n})$.

Take the vertices intersected by a randomly scaled "balanced" circle

Processing the input (2)

$\mathbb{G}_\rho$ = class of the subgraphs of intersection graphs with density ρ

Lemma

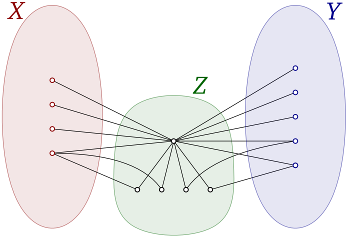
For every $G \in \mathbb{G}_\rho$, we can compute in expected $O(\rho n \log n)$ a pair (G', T') s.t.

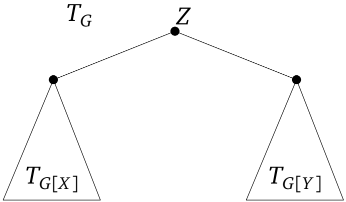
- *G' is obtained from G by a series of vertex-splits*
- *G' has degree 4 and $O(\rho n)$ vertices and edges*
- *T' is a $(O(\rho), 1/2, \alpha)$-separator tree of G'*

Follow [Yuster and Zwick '07] tracking the dependency in ρ

Input processing step

1. Separator tree; $|Z| = O(\sqrt{\rho n})$

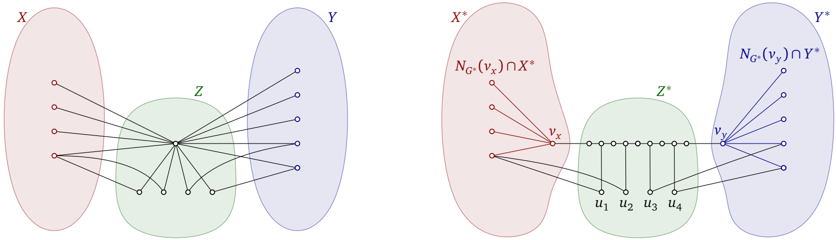




Input processing step

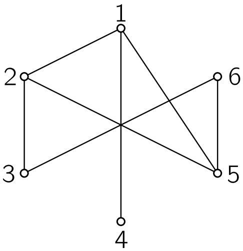
1. Separator tree; $|Z| = O(\sqrt{\rho n})$
2. Equivalent instance with bounded degree & separator tree; new instance subgraph of geometric intersection graph;

   Z^* may have larger density but only $O(\rho^{3/2} n^{1/2})$ vertices



Setting up matrix B

1. Set symbolic matrix $\tilde{A}(G')$ with variables $x_{i,j}$

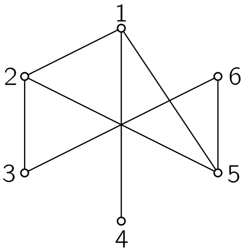


$A[X] =$

	$x_{1,2}$		$x_{1,4}$	$x_{1,5}$	
$-x_{1,2}$		$x_{2,3}$		$x_{2,5}$	
	$-x_{2,3}$				$x_{3,6}$
$-x_{1,4}$					
$-x_{1,5}$	$-x_{2,5}$				$x_{5,6}$
		$-x_{3,6}$		$-x_{5,6}$	

Setting up matrix B

1. Set symbolic matrix $\tilde{A}(G')$ with variables $x_{i,j}$
2. $A \rightarrow$ choose random values in $\mathbb{Z}_{n^4}$ for the $x_{i,j}$
3. $\text{rank}(A) = 2 \cdot (\text{size max matching}(G'))$, whp

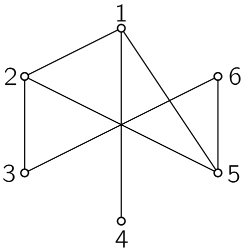


$A =$

	10		−7	16	
−10		14		3	
	−14				0
7					
−16	−3				−14
		0		14	

Setting up matrix B

1. Set symbolic matrix $\tilde{A}(G')$ with variables $x_{i,j}$
2. $A \rightarrow$ choose random values in $\mathbb{Z}_{n^4}$ for the $x_{i,j}$
3. $\text{rank}(A) = 2 \cdot (\text{size max matching}(G'))$, whp
4. $B := AA^T$ is symmetric and has same rank as A
5. Gaussian elimination in B without pivoting, whp

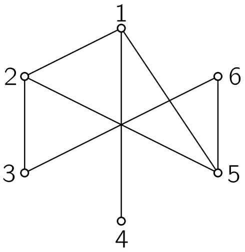


$A =$

	10		−7	16	
−10		14		3	
	−14				0
7					
−16	−3				−14
		0		14	

Setting up matrix B

1. Set symbolic matrix $\tilde{A}(G')$ with variables $x_{i,j}$
2. $A \rightarrow$ choose random values in $\mathbb{Z}_{n^4}$ for the $x_{i,j}$
3. $\text{rank}(A) = 2 \cdot (\text{size max matching}(G'))$, whp
4. $B := AA^T$ is symmetric and has same rank as A
5. Gaussian elimination in B without pivoting, whp
6. G'^2 also has nice tree separator because of bounded degree



$A =$

	10		-7	16	
-10		14		3	
	-14				0
7					
-16	-3				-14
		0		14	

Balanced Separators/Nested dissection come in

- Gaussian elimination in $B = AA^T$ in $O(\rho^{3\omega/2} n^{\omega/2})$ time using nested dissection & separator tree
  - as for planar graphs, but with parameter ρ
- This gives size of max matching

Balanced Separators/Nested dissection come in

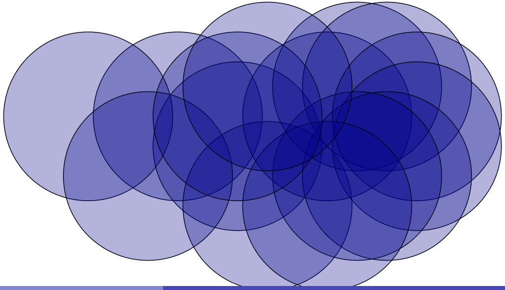
- Gaussian elimination in $B = AA^T$ in $O(\rho^{3\omega/2} n^{\omega/2})$ time using nested dissection & separator tree
  - as for planar graphs, but with parameter ρ
- This gives size of max matching
- Also useful for identifying vertices in maximum matching
  - searching max matching $\rightarrow$ searching perfect matching

Balanced Separators/Nested dissection come in

- Gaussian elimination in $B = AA^T$ in $O(\rho^{3\omega/2} n^{\omega/2})$ time using nested dissection & separator tree
  - as for planar graphs, but with parameter ρ
- This gives size of max matching
- Also useful for identifying vertices in maximum matching
  - searching max matching $\rightarrow$ searching perfect matching
- For the separator Z^*, use $O(1)$ Gaussian eliminations to find partial matching
  - included in a perfect matching of G', and
  - spanning all the vertices of Z^*

Sparsification – How?

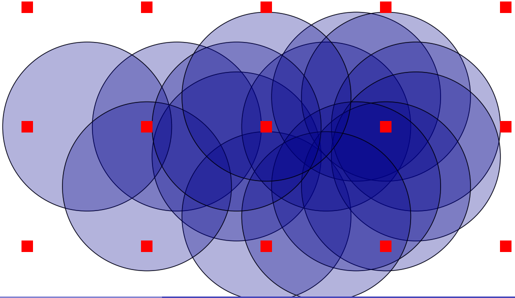
Consider the case of unit disks



Sparsification – How?

Consider the case of unit disks

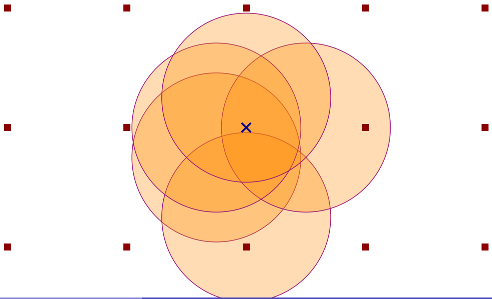
- a grid P of points; cluster the disks into cliques $\{\mathcal{D}_p \mid p \in P\}$



Sparsification – How?

Consider the case of unit disks

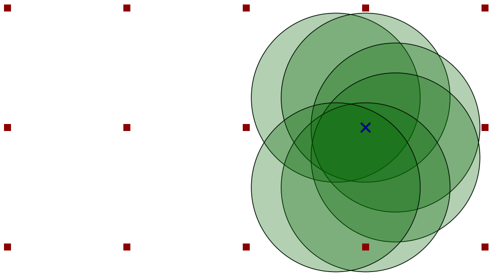
- a grid P of points; cluster the disks into cliques $\{\mathcal{D}_p \mid p \in P\}$



Sparsification – How?

Consider the case of unit disks

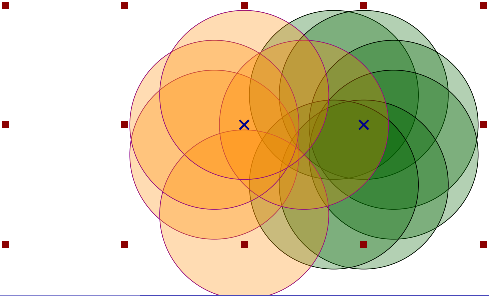
- a grid P of points; cluster the disks into cliques $\{\mathcal{D}_p \mid p \in P\}$



Sparsification – How?

Consider the case of unit disks

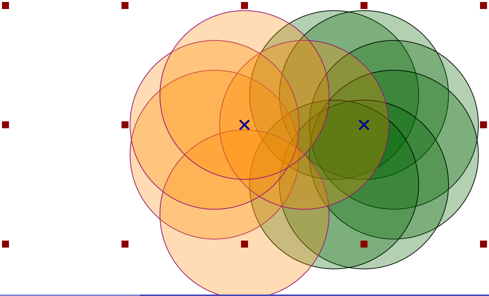
- a grid P of points; cluster the disks into cliques $\{\mathcal{D}_p \mid p \in P\}$
- each $\mathcal{D}_p$ interacts with $O(1)$ different clusters $\mathcal{D}_q$



Sparsification – How?

Consider the case of unit disks

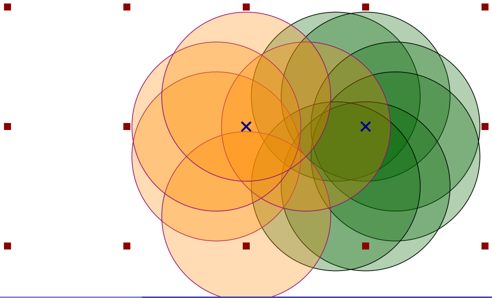
- a grid P of points; cluster the disks into cliques $\{\mathcal{D}_p \mid p \in P\}$
- each $\mathcal{D}_p$ interacts with $O(1)$ different clusters $\mathcal{D}_q$
- maximum matching with $\leqslant 1$ edge in $\mathcal{D}_p \times \mathcal{D}_q$



Sparsification – How?

Consider the case of unit disks

- a grid P of points; cluster the disks into cliques $\{\mathcal{D}_p \mid p \in P\}$
- each $\mathcal{D}_p$ interacts with $O(1)$ different clusters $\mathcal{D}_q$
- maximum matching with $\leqslant 1$ edge in $\mathcal{D}_p \times \mathcal{D}_q$
- in each D_p, keep $\bigcup_q \Big(\min\{\text{maximal}, O(1)\} \text{ matching in } \mathcal{D}_p \times \mathcal{D}_q\Big)$ to keep enough candidates
- data structures (shape dependent) for maximal matching



Conclusion

- Combination of graphs, geometry, algebra, algorithms
- State for unit disks/squares:
  - $O(n^{1.19})$ for general
  - $O(n^{1.5})$ for bicolored case
- near-linear?
- or deciding existence of perfect matching
- bicolored unit square perfect matching
  relevant for persistence diagrams
- Computer Algebra for Computational Geometry vs.
  Algebraic Methods for Discrete Geometry

Conclusion

- Combination of graphs, geometry, algebra, algorithms
- State for unit disks/squares:
  - $O(n^{1.19})$ for general
  - $O(n^{1.5})$ for bicolored case
- near-linear?
- or deciding existence of perfect matching
- bicolored unit square perfect matching
  relevant for persistence diagrams
- Computer Algebra for Computational Geometry vs.
  Algebraic Methods for Discrete Geometry

Thank you for your attention!