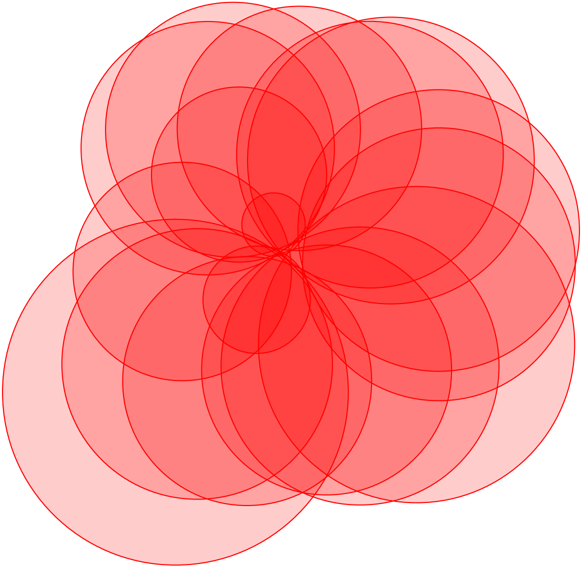
Maximum Clique on Disks

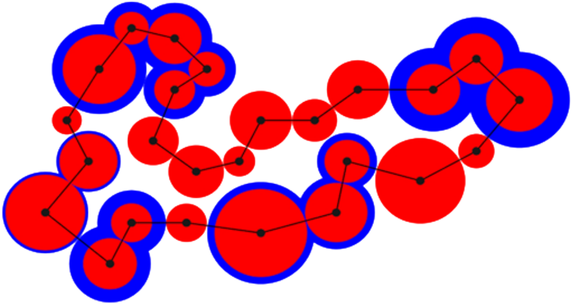
Édouard Bonnet joint work with Panos Giannopoulos, Eun Jung Kim, Paweł Rzążewski, and Florian Sikora

Middlesex University, London

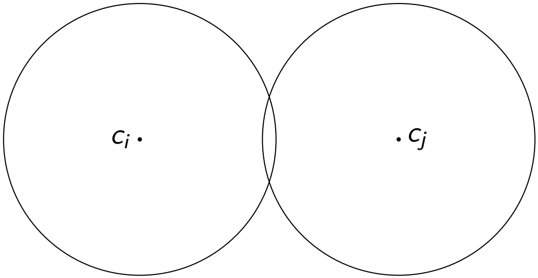
Séminaire équipe *Optimisation Combinatoire*, G-SCOP, Grenoble, 18 décembre 2017



Find a largest collection of disks that pairwise intersect

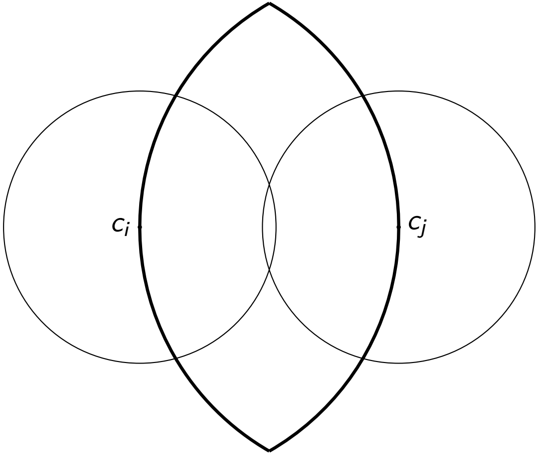


Polynomial algorithm on unit disks by Clark, Colbourn, and Johnson, 1990.



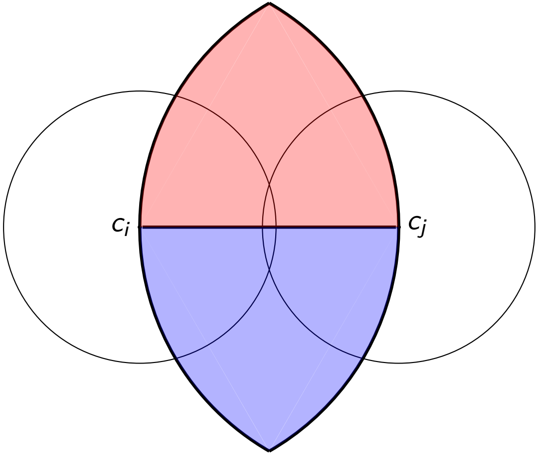
Guess two farthest disks in an optimum solution S.

Polynomial algorithm on unit disks by Clark, Colbourn, and Johnson, 1990.



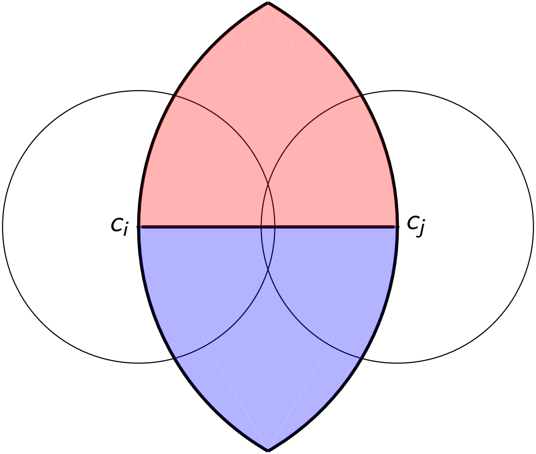
Hence, all the centers of S lie inside the bold digon.

Polynomial algorithm on unit disks by Clark, Colbourn, and Johnson, 1990.



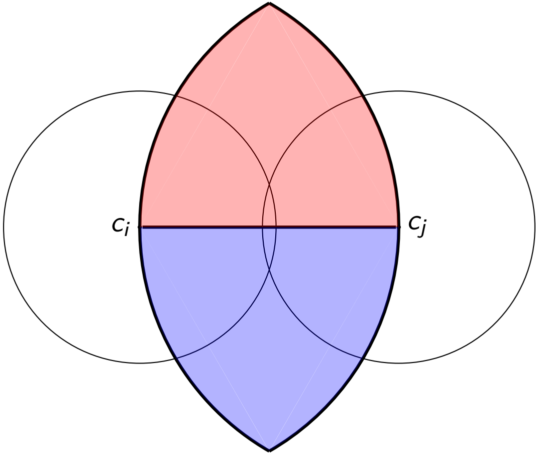
Two disks centered in the same-color region intersect.

Polynomial algorithm on unit disks by Clark, Colbourn, and Johnson, 1990.



We solve Max Clique in a co-bipartite graph.

Polynomial algorithm on unit disks by Clark, Colbourn, and Johnson, 1990.



We solve Max Independent Set in a bipartite graph.

Disk graphs

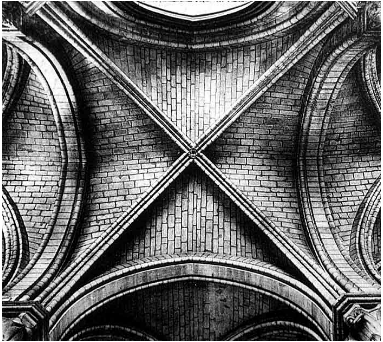
Unweighted problems

3-Colourability [?]	NP-complete	[+]Details
Clique [?]	Unknown to ISGCI	[+]Details
Clique cover [?]	NP-complete	[+]Details
Colourability [?]	NP-complete	[+]Details
Domination [?]	NP-complete	[+]Details
Feedback vertex set [?]	NP-complete	[+]Details
Graph isomorphism [?]	Unknown to ISGCI	[+]Details
Hamiltonian cycle [?]	NP-complete	[+]Details
Hamiltonian path [?]	NP-complete	[+]Details
Independent dominating set [?]	NP-complete	[+]Details
Independent set [?]	NP-complete	[+]Details
Maximum bisection [?]	NP-complete	[+]Details
Maximum cut [?]	NP-complete	[+]Details
Minimum bisection [?]	NP-complete	[+]Details
Monopolarity [?]	NP-complete	[+]Details
Polarity [?]	NP-complete	[+]Details
Recognition [?]	NP-hard	[+]Details

Inherits the NP-hardness of planar graphs.

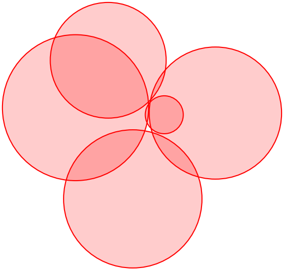
So what is known for Max Clique on disk graphs?

- Polynomial-time 2-approximation
  - For any clique there are 4 points hitting all the disks.
  - Guess those points.
  - Solve exactly in each of the $\binom{4}{2}$ co-bipartite graphs.
  - Output the best solution.
- No non-trivial exact algorithm known.



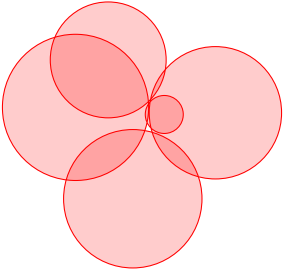
And what is known about disk graphs?

- Every planar graph is a disk graph.
- Every triangle-free disk graph is planar (centers $\rightarrow$ vertices).
- So a triangle-free non-planar graph like $K_{3,3}$ is not disk.
- A subdivision of a non-planar graph is not a disk graph (more generally not a string graph).
- ...



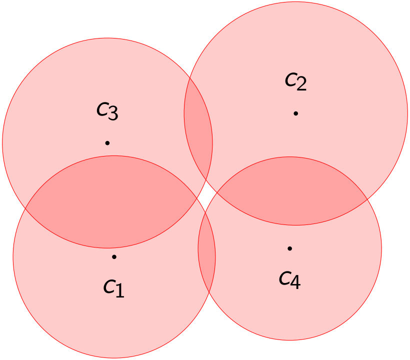
And what is known about disk graphs?

- Every planar graph is a disk graph.
- Every triangle-free disk graph is planar (centers $\rightarrow$ vertices).
- So a triangle-free non-planar graph like $K_{3,3}$ is not disk.
- A subdivision of a non-planar graph is not a disk graph (more generally not a string graph).
- ...

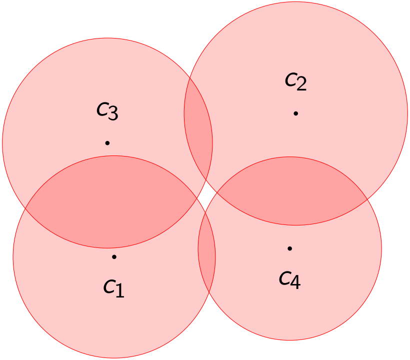


Other ways of showing that a graph is not disk?

Say the 4 centers encoding a $K_{2,2} = \overline{2K_2}$ are in convex position.

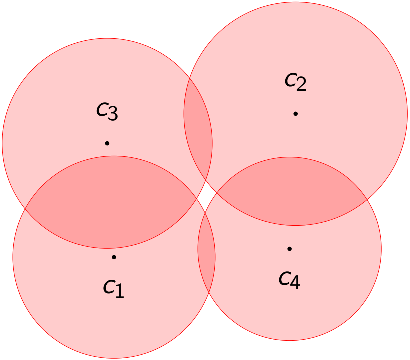


Say the 4 centers encoding a $K_{2,2} = \overline{2K_2}$ are in convex position.



Then the two non-edges should be diagonal.

Say the 4 centers encoding a $K_{2,2} = \overline{2K_2}$ are in convex position.

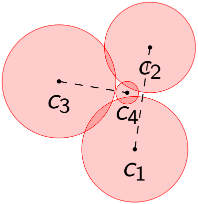


Then the two non-edges should be diagonal.

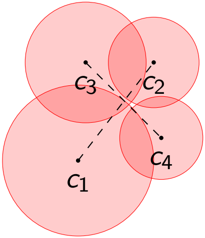
Suppose $d(c_1, c_3) > r_1 + r_3$ and $d(c_2, c_4) > r_2 + r_4$.
But $d(c_1, c_3) + d(c_2, c_4) \leqslant d(c_1, c_2) + d(c_3, c_4) \leqslant r_1 + r_2 + r_3 + r_4$, a contradiction.

Conclusion: the 4 centers of an induced $\overline{2K_2}$ are either

- not in convex position or
- in convex position with the non-edges being *diagonal*.

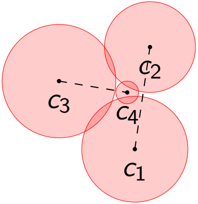


or

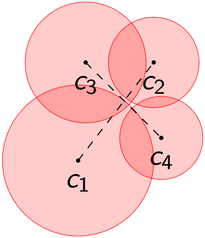


Conclusion: the 4 centers of an induced $\overline{2K_2}$ are either

- not in convex position or
- in convex position with the non-edges being *diagonal*.



or

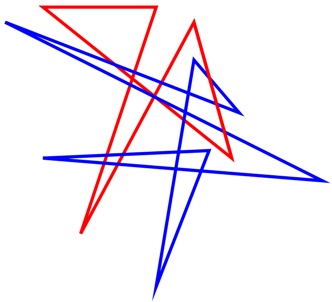


Reformulation: either

- the line $\ell(c_1, c_2)$ crosses the segment c_3c_4, or
- the line $\ell(c_3, c_4)$ crosses the segment c_1c_2, or
- both; equivalently, the segments c_1c_2 and c_3c_4 cross.

Assume $\overline{C_s + C_t}$ is a disk graph.

Link consecutive centers of the two disjoint cycles (non-edges).

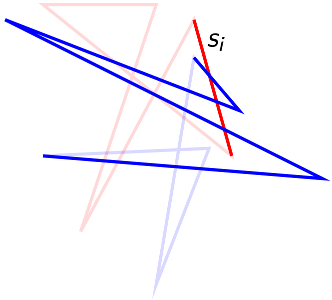


For each red segment s_i, we denote by:

- a_i the number of blue segments crossed by $\ell(s_i)$.
- b_i the number of blue segments whose extension cross s_i.
- c_i the number of blue segments intersecting s_i.

Assume $\overline{C_s + C_t}$ is a disk graph.

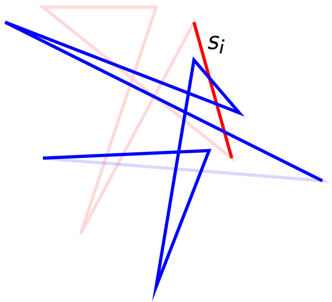
Link consecutive centers of the two disjoint cycles (non-edges).



For each red segment s_i, we denote by:

- a_i **the number of blue segments crossed by $\ell(s_i)$.**
- b_i the number of blue segments whose extension cross s_i.
- c_i the number of blue segments intersecting s_i.

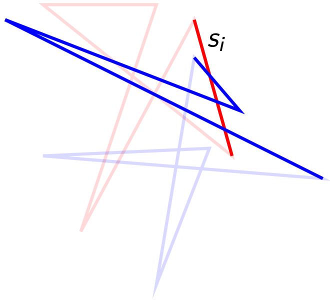
Assume $\overline{C_s + C_t}$ is a disk graph.
Link consecutive centers of the two disjoint cycles (non-edges).



For each red segment s_i, we denote by:

- a_i the number of blue segments crossed by $\ell(s_i)$.
- b_i **the number of blue segments whose extension cross** s_i**.**
- c_i the number of blue segments intersecting s_i.

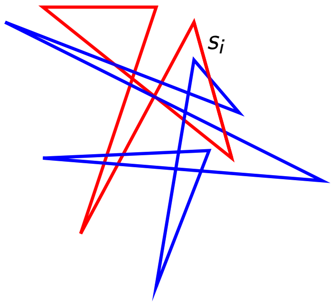
Assume $\overline{C_s + C_t}$ is a disk graph.
Link consecutive centers of the two disjoint cycles (non-edges).



For each red segment s_i, we denote by:

- a_i the number of blue segments crossed by $\ell(s_i)$.
- b_i the number of blue segments whose extension cross s_i.
- c_i **the number of blue segments intersecting** s_i**.**

Assume $\overline{C_s + C_t}$ is a disk graph.
Link consecutive centers of the two disjoint cycles (non-edges).



For each red segment s_i, we denote by:

- a_i the number of blue segments crossed by $\ell(s_i)$.
- b_i the number of blue segments whose extension cross s_i.
- c_i the number of blue segments intersecting s_i.

It should be that $a_i + b_i - c_i = t$.

$$\sum_{1 \leqslant i \leqslant s} a_i + b_i - c_i = st$$

1) a_i is even:

$$\sum_{1 \leqslant i \leqslant s} a_i + b_i - c_i = st$$

1) a_i is even: number of intersections of a line with a closed curve.

2) $\sum_{1 \leqslant i \leqslant s} b_i =$

$$\sum_{1 \leqslant i \leqslant s} a_i + b_i - c_i = st$$

1) a_i is even: number of intersections of a line with a closed curve.

2) $\sum_{1 \leqslant i \leqslant s} b_i = \sum_{1 \leqslant i \leqslant t} a'_i$ is therefore even. (a'_j, b'_j, c'_j same for blue segments)

3) $\sum_{1 \leqslant i \leqslant s} c_i$ is even:

$$\sum_{1 \leqslant i \leqslant s} a_i + b_i - c_i = st$$

1) a_i is even: number of intersections of a line with a closed curve.

2) $\sum_{1 \leqslant i \leqslant s} b_i = \sum_{1 \leqslant i \leqslant t} a'_i$ is therefore even. (a'_j, b'_j, c'_j same for blue segments)

3) $\sum_{1 \leqslant i \leqslant s} c_i$ is even: number of intersections of two closed curves.

$$\sum_{1 \leqslant i \leqslant s} a_i + b_i - c_i =$$

$$\sum_{1 \leqslant i \leqslant s} a_i + b_i - c_i = st$$

1) a_i is even: number of intersections of a line with a closed curve.

2) $\sum_{1 \leqslant i \leqslant s} b_i = \sum_{1 \leqslant i \leqslant t} a'_i$ is therefore even. (a'_j, b'_j, c'_j same for blue segments)

3) $\sum_{1 \leqslant i \leqslant s} c_i$ is even: number of intersections of two closed curves.

$\sum_{1 \leqslant i \leqslant s} a_i + b_i - c_i = \sum_{1 \leqslant i \leqslant s} a_i + \sum_{1 \leqslant i \leqslant t} a'_i - \sum_{1 \leqslant i \leqslant s} c_i$ is even.

$$\sum_{1\leqslant i\leqslant s} a_i + b_i - c_i = st$$

1) a_i is even: number of intersections of a line with a closed curve.

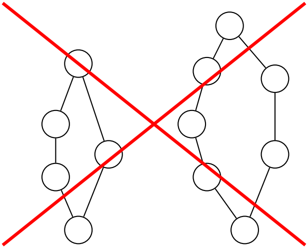
2) $\sum_{1\leqslant i\leqslant s} b_i = \sum_{1\leqslant i\leqslant t} a'_i$ is therefore even. (a'_j, b'_j, c'_j same for blue segments)

3) $\sum_{1\leqslant i\leqslant s} c_i$ is even: number of intersections of two closed curves.

$$\sum_{1\leqslant i\leqslant s} a_i + b_i - c_i = \sum_{1\leqslant i\leqslant s} a_i + \sum_{1\leqslant i\leqslant t} a'_i - \sum_{1\leqslant i\leqslant s} c_i \text{ is even.}$$

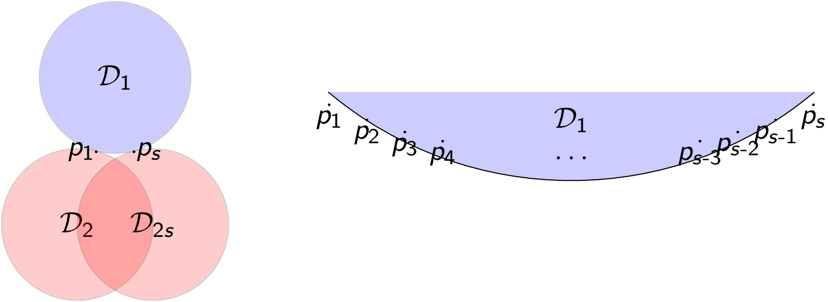
Hence s and t cannot be both odd.

The complement of two odd cycles is not a disk graph.



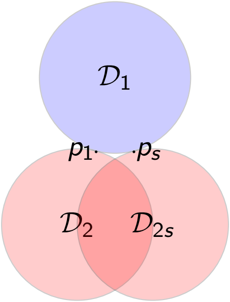
Are there other graphs of co-degree 2 which are not disk?

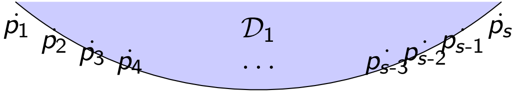
Complement of an even cycle $1, 2, \ldots, 2s$



We start by positioning $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_{2s}$.
We draw a convex chain between p_1 and p_s.

Complement of an even cycle $1, 2, \ldots, 2s$

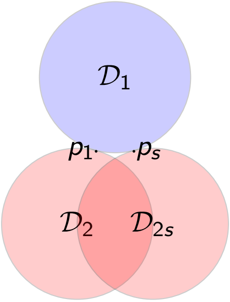


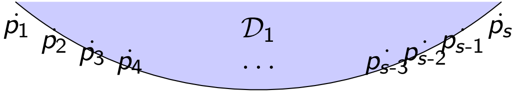


We start by positioning $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_{2s}$.
We draw a convex chain between p_1 and p_s.

D_{2i}: same radius and boundary crosses p_i with tangent $\ell(p_{i-1}p_{i+1})$

Complement of an even cycle $1, 2, \ldots, 2s$

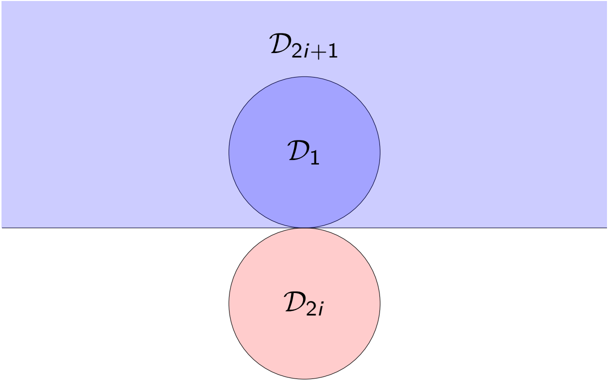




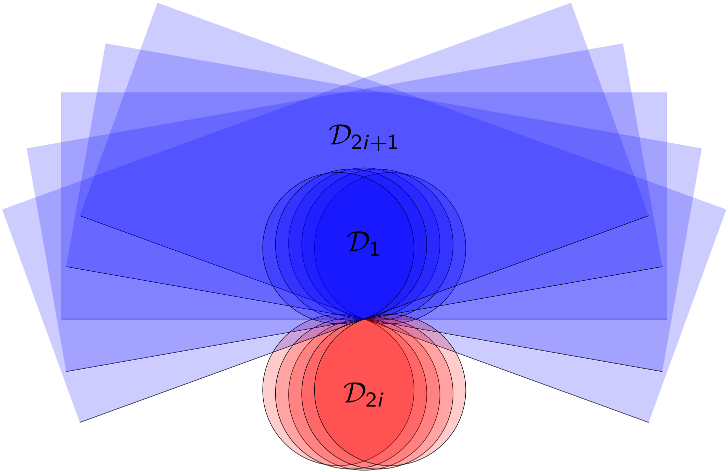
We start by positioning $\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_{2s}$.
We draw a convex chain between p_1 and p_s.

D_{2i}: same radius and boundary crosses p_i with tangent $\ell(p_{i-1}p_{i+1})$
D_{2i+1}: larger radius and "co-tangent" to D_{2i} and D_{2i+2}.

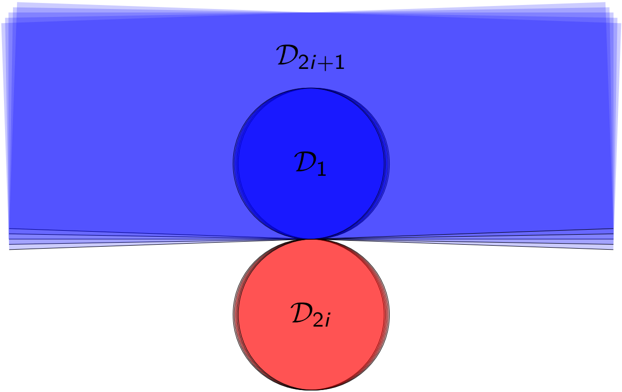
Stacking complements of even cycles



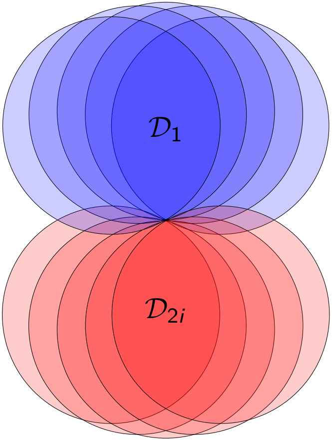
Stacking complements of even cycles

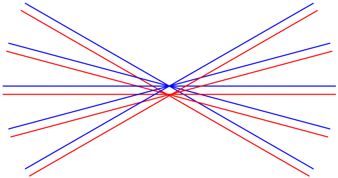


Stacking complements of even cycles

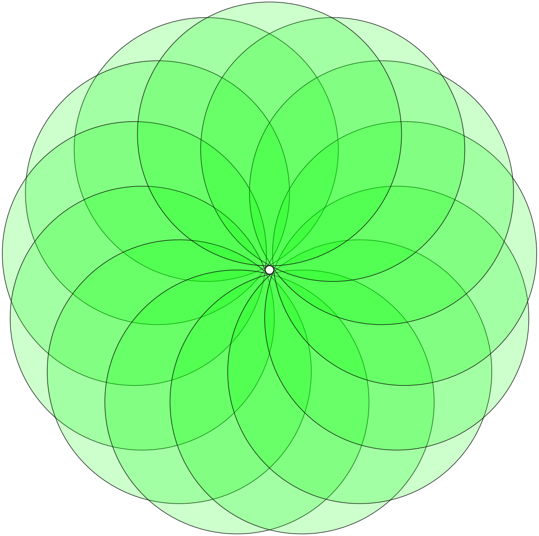


Disks of different cycle complements intersect

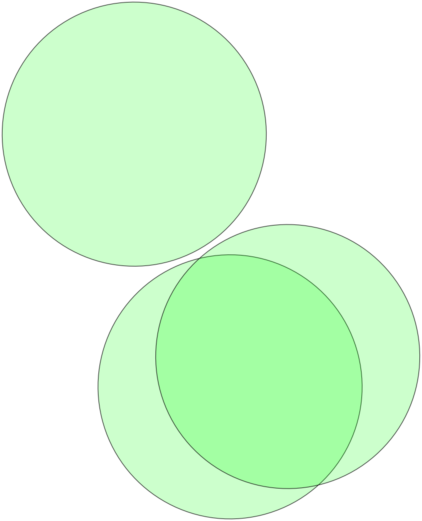




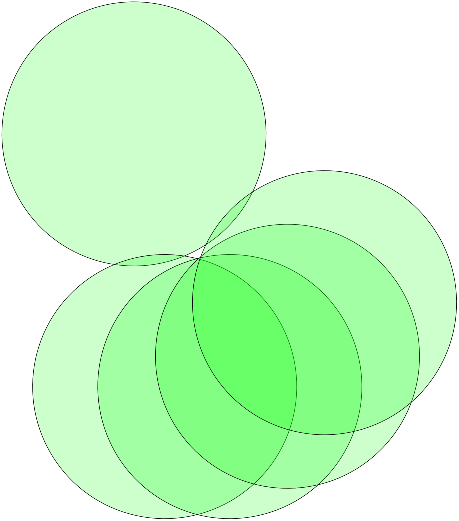
Complement of odd cycle by unit disks (Atminas & Zamaraev)



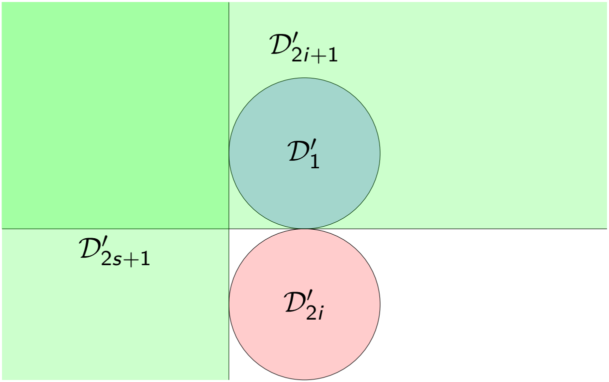
Complement of odd cycle by unit disks (Atminas & Zamaraev)



Complement of odd cycle by unit disks (Atminas & Zamaraev)

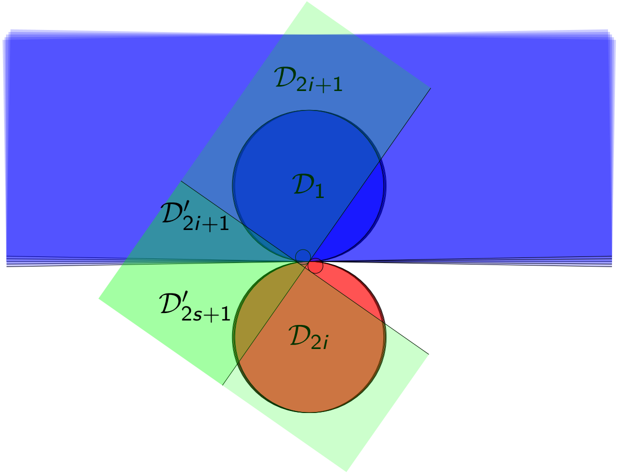


Different representation with non-unit disks

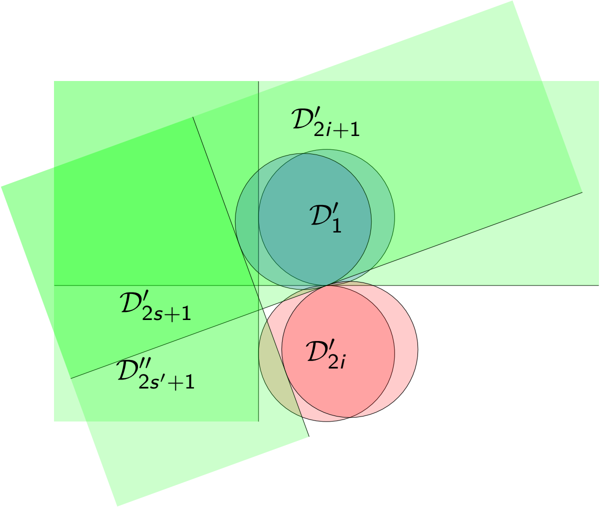


Same construction except $\mathcal{D}'_1$ intersects $\mathcal{D}'_{2s}$ and $\mathcal{D}'_{2s+1}$ is "co-tangent" to $\mathcal{D}'_1$ and $\mathcal{D}'_{2s}$.

Complement of many even cycles and one odd cycle



Sanity check: trying to stack complements of odd cycles



$\mathcal{D}''_{2s'+1}$ cannot possibly intersect $\mathcal{D}'_1$

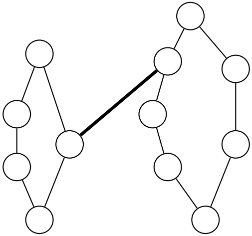
Going back to algorithms.

Can we solve Max Independent Set more efficiently if there are no two vertex-disjoint odd cycles as an induced subgraph?

Going back to algorithms.

Can we solve Max Independent Set more efficiently if there are no two vertex-disjoint odd cycles as an induced subgraph?

Another way to see it:
at least one edge between two vertex-disjoint odd cycles



Quasi-polynomial time approximation-scheme (QPTAS)

ocp(G): maximum size of an odd cycle packing.

Theorem (Bock et al. 2014)

PTAS for Max Independent Set for $ocp = o(n/\log n)$.

Quasi-polynomial time approximation-scheme (QPTAS)

ocp(G): maximum size of an odd cycle packing.

Theorem (Bock et al. 2014)

PTAS for Max Independent Set for $ocp = o(n/\log n)$.

Lemma

Let H *complement of a disk graph with* n *vertices.*
If $ocp(H) > n/\log^2 n$*, then vertex of degree at least* $n/\log^4 n$.

Proof.

Quasi-polynomial time approximation-scheme (QPTAS)

ocp(G): maximum size of an odd cycle packing.

Theorem (Bock et al. 2014)

PTAS for Max Independent Set for ocp $= o(n/\log n)$.

Lemma

Let H complement of a disk graph with n vertices.
If ocp$(H) > n/\log^2 n$, then vertex of degree at least $n/\log^4 n$.

Proof.

The shortest odd cycle has size at most $\log^2 n$.

Quasi-polynomial time approximation-scheme (QPTAS)

ocp(G): maximum size of an odd cycle packing.

Theorem (Bock et al. 2014)

PTAS for Max Independent Set for ocp $= o(n/\log n)$.

Lemma

Let H complement of a disk graph with n vertices.
If ocp$(H) > n/\log^2 n$, then vertex of degree at least $n/\log^4 n$.

Proof.

The shortest odd cycle has size at most $\log^2 n$.
There is a vertex of this cycle with degree at least $n/\log^4 n$. □

Quasi-polynomial time approximation-scheme (QPTAS)

ocp(G): maximum size of an odd cycle packing.

Theorem (Bock et al. 2014)

PTAS for Max Independent Set for ocp $= o(n/\log n)$.

Lemma

Let H complement of a disk graph with n vertices.
If ocp$(H) > n/\log^2 n$, then vertex of degree at least $n/\log^4 n$.

Proof.

The shortest odd cycle has size at most $\log^2 n$.
There is a vertex of this cycle with degree at least $n/\log^4 n$. □

Branching factor $(1, n/\log^4 n)$ (in $2^{\log^5 n}$), and PTAS otherwise.

Subexponential algorithm

Theorem (Györi et al. 1997)

A graph with odd girth at least δn has an odd cycle cover size $O((1/\delta)\log(1/\delta))$.

Subexponential algorithm

Theorem (Györi et al. 1997)

A graph with odd girth at least δn has an odd cycle cover size $O((1/\delta)\log(1/\delta))$.

Let G be the co-disk, Δ its degree, c its odd girth.
We can:

- branch in time $2^{\tilde{O}(n/\Delta)}$.
- solve in time $2^{O(\Delta c)}$.
- solve in time $2^{\tilde{O}(n/c)}$.

Subexponential algorithm

Theorem (Györi et al. 1997)

A graph with odd girth at least δn has an odd cycle cover size $O((1/\delta)\log(1/\delta))$.

Let G be the co-disk, Δ its degree, c its odd girth.
We can:

- branch in time $2^{\tilde{O}(n/\Delta)}$.
- solve in time $2^{O(\Delta c)}$.
- solve in time $2^{\tilde{O}(n/c)}$.

$2^{\tilde{O}(\min(n/\Delta, n/c, c\Delta))} \leqslant 2^{\tilde{O}(n^{2/3})}$ for $\Delta = c = n^{1/3}$.

Subexponential algorithm

Theorem (Györi et al. 1997)

A graph with odd girth at least δn has an odd cycle cover size $O((1/\delta)\log(1/\delta))$.

Let G be the co-disk, Δ its degree, c its odd girth.
We can:

- branch in time $2^{\tilde{O}(n/\Delta)}$.
- solve in time $2^{O(\Delta c)}$.
- solve in time $2^{\tilde{O}(n/c)}$.

$2^{\tilde{O}(\min(n/\Delta, n/c, c\Delta))} \leqslant 2^{\tilde{O}(n^{2/3})}$ for $\Delta = c = n^{1/3}$.
$2^{\tilde{O}(\sqrt{n})}$ if the degree or the odd girth is constant, polytime if both.

Filled ellipses and triangles

2-subdivisions: graphs where each edge is subdivided exactly twice
co-2-subdivisions: complements of 2-subdivisions

Theorem (technical)

For some α, Maximum Independent Set on 2-subdivisions is not α-approximable algorithm in $2^{n^{1-\varepsilon}}$, unless the ETH fails.

Filled ellipses and triangles

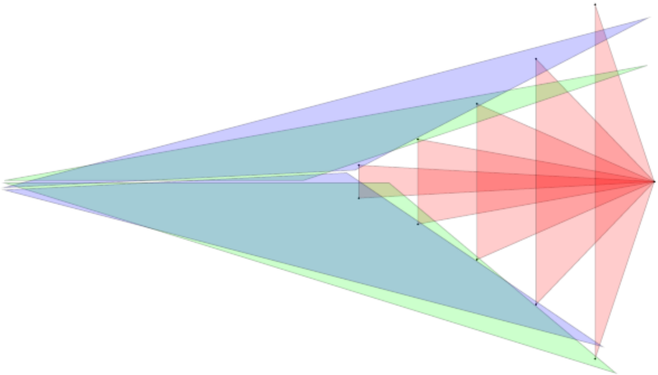
2-subdivisions: graphs where each edge is subdivided exactly twice
co-2-subdivisions: complements of 2-subdivisions

Theorem (technical)

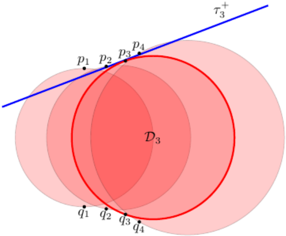
For some α, Maximum Independent Set on 2-subdivisions is not α-approximable algorithm in $2^{n^{1-\varepsilon}}$, unless the ETH fails.

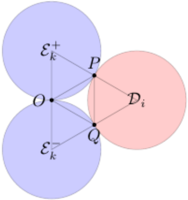
Graphs of filled ellipses or filled triangles contain all the co-2-subdivisions.

Filled triangles

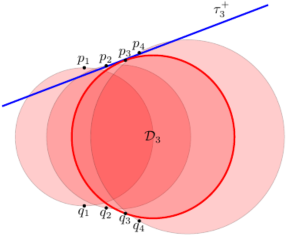


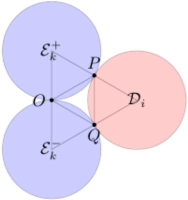
Filled ellipses





Filled ellipses





Thank you for your attention!