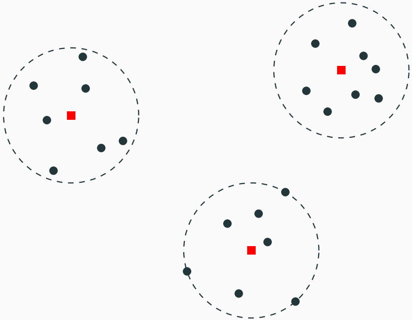
Clustering with Neighborhoods is Hard for Squares

Georgiy Klimenko and *Benjamin Raichel*

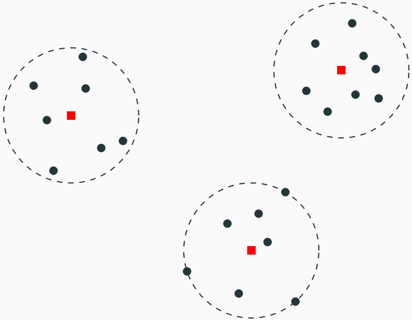
UT Dallas

k-center Clustering



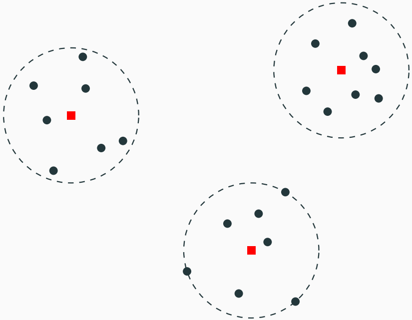
- Given a set of n points in the plane and a positive integer k.

k-center Clustering



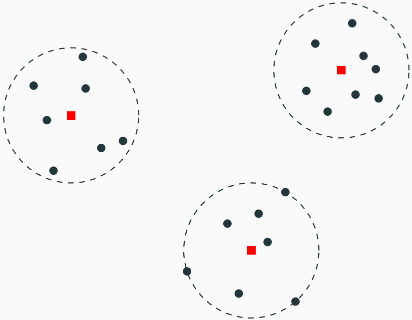
- Given a set of n points in the plane and a positive integer k.
- k-center problem: Find the smallest radius r such that there are k balls of radius r that contain all the points.

k-center Clustering



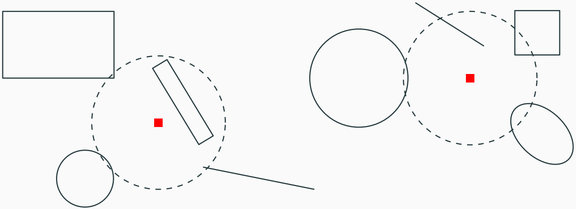
- Given a set of n points in the plane and a positive integer k.
- k-center problem: Find the smallest radius r such that there are k balls of radius r that contain all the points.
- k-center has several well known 2-approximations (k is a hard constraint, the approximation is on the radius).

k-center Clustering



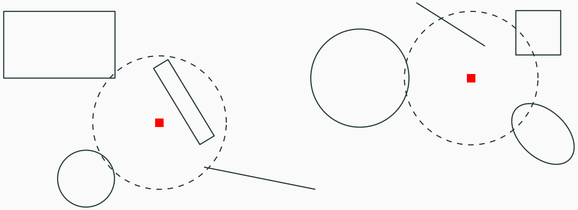
- Given a set of n points in the plane and a positive integer k.
- k-center problem: Find the smallest radius r such that there are k balls of radius r that contain all the points.
- k-center has several well known 2-approximations (k is a hard constraint, the approximation is on the radius).
- Feder and Greene showed it is ≈ 1.82 hard to approximate in the plane (in general metrics it is 2-hard).

Clustering with Neighborhoods



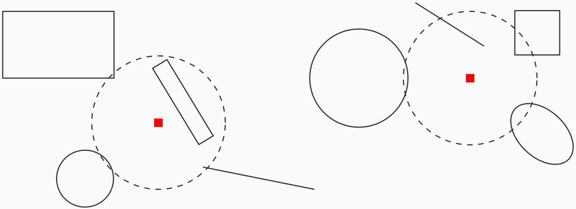
- Given a set of n disjoint objects in the plane (segments, disks, squares, etc.) and a positive integer k.

Clustering with Neighborhoods



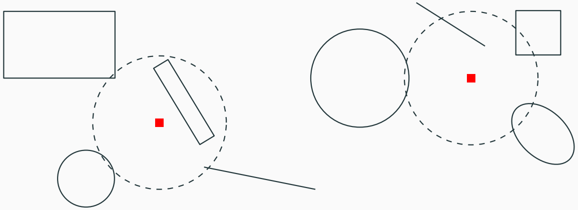
- Given a set of n disjoint objects in the plane (segments, disks, squares, etc.) and a positive integer k.
- Clustering with Neighborhoods (**CN**): Find the smallest radius r such that there are k balls of radius r that intersect all the objects.

Clustering with Neighborhoods



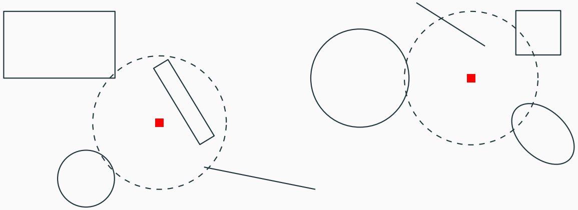
- Given a set of n disjoint objects in the plane (segments, disks, squares, etc.) and a positive integer k.
- Clustering with Neighborhoods (**CN**): Find the smallest radius r such that there are k balls of radius r that intersect all the objects.
- Known: For segments cannot approx within any computable factor.

Clustering with Neighborhoods



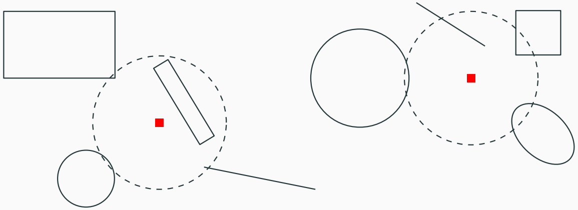
- Given a set of n disjoint objects in the plane (segments, disks, squares, etc.) and a positive integer k.
- Clustering with Neighborhoods (**CN**): Find the smallest radius r such that there are k balls of radius r that intersect all the objects.
- Known: For segments cannot approx within any computable factor.
- Known: For disks, ≈ 6.99 hard, and an ≈ 8.46 approx algorithm.

Clustering with Neighborhoods



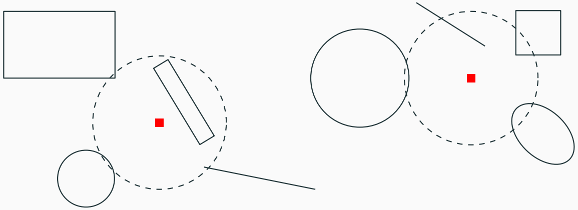
- Given a set of n disjoint objects in the plane (segments, disks, squares, etc.) and a positive integer k.
- Clustering with Neighborhoods (**CN**): Find the smallest radius r such that there are k balls of radius r that intersect all the objects.
- Known: For segments cannot approx within any computable factor.
- Known: For disks, ≈ 6.99 hard, and an ≈ 8.46 approx algorithm.
- We suspected would similarly be $O(1)$ hard for other “fat” objects.

Clustering with Neighborhoods



- Given a set of n disjoint objects in the plane (segments, disks, squares, etc.) and a positive integer k.
- Clustering with Neighborhoods (**CN**): Find the smallest radius r such that there are k balls of radius r that intersect all the objects.
- Known: For segments cannot approx within any computable factor.
- Known: For disks, ≈ 6.99 hard, and an ≈ 8.46 approx algorithm.
- We suspected would similarly be $O(1)$ hard for other "fat" objects.
- Surprisingly, we proved as hard for squares as for line segments.

Clustering with Neighborhoods



- Given a set of n disjoint objects in the plane (segments, disks, squares, etc.) and a positive integer k.
- Clustering with Neighborhoods (**CN**): Find the smallest radius r such that there are k balls of radius r that intersect all the objects.
- Known: For segments cannot approx within any computable factor.
- Known: For disks, ≈ 6.99 hard, and an ≈ 8.46 approx algorithm.
- We suspected would similarly be $O(1)$ hard for other “fat” objects.
- Surprisingly, we proved as hard for squares as for line segments.
- Holds even if we require the squares be unit and axis-aligned.

Segment Intuition

- **P3VC**: Planar (straight line) instance of Vertex Cover, max degree 3.

Segment Intuition

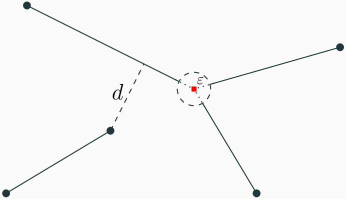
- **P3VC**: Planar (straight line) instance of Vertex Cover, max degree 3.
- **P3VC** is NP-hard. FG used to show k-center is 1.82 hard.

Segment Intuition

- **P3VC**: Planar (straight line) instance of Vertex Cover, max degree 3.
- **P3VC** is NP-hard. FG used to show k-center is 1.82 hard.
- To show **CN** NP-hard, just view the edges as segment objects. There is **P3VC** solution of size k iff a zero radius solution of size k for **CN**.

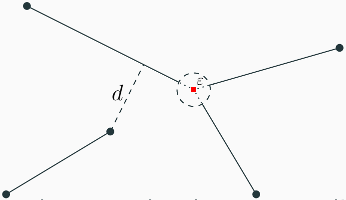
Segment Intuition

- **P3VC**: Planar (straight line) instance of Vertex Cover, max degree 3.
- **P3VC** is NP-hard. FG used to show k-center is 1.82 hard.
- To show **CN** NP-hard, just view the edges as segment objects. There is **P3VC** solution of size *k* iff a zero radius solution of size *k* for **CN**.
- To show hard to approx, just shrink each edge by ε around each vertex. Now there is a **P3VC** solution iff ε radius solution for **CN**.



Segment Intuition

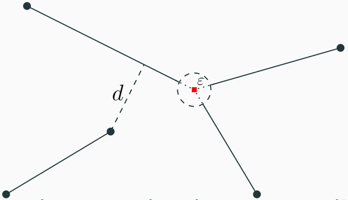
- **P3VC**: Planar (straight line) instance of Vertex Cover, max degree 3.
- **P3VC** is NP-hard. FG used to show k-center is 1.82 hard.
- To show **CN** NP-hard, just view the edges as segment objects. There is **P3VC** solution of size k iff a zero radius solution of size k for **CN**.
- To show hard to approx, just shrink each edge by ε around each vertex. Now there is a **P3VC** solution iff ε radius solution for **CN**.



- Let d be the distance between the closest non-adjacent edges. Note the correspondence with **P3VC** holds for covering balls with radius up to $d/2$, since such balls cannot cover non-adjacent edges.

Segment Intuition

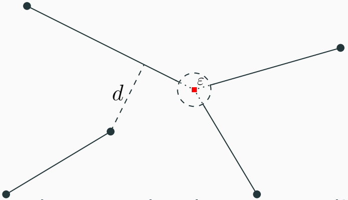
- **P3VC**: Planar (straight line) instance of Vertex Cover, max degree 3.
- **P3VC** is NP-hard. FG used to show k-center is 1.82 hard.
- To show **CN** NP-hard, just view the edges as segment objects. There is **P3VC** solution of size k iff a zero radius solution of size k for **CN**.
- To show hard to approx, just shrink each edge by ε around each vertex. Now there is a **P3VC** solution iff ε radius solution for **CN**.



- Let d be the distance between the closest non-adjacent edges. Note the correspondence with **P3VC** holds for covering balls with radius up to $d/2$, since such balls cannot cover non-adjacent edges.
- Thus we cannot approx within $(d/2)/\varepsilon$, as we could then distinguish between there being a radius ε solution or it requiring $> d/2$.

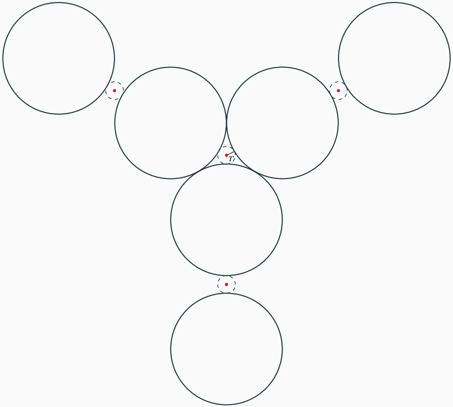
Segment Intuition

- **P3VC**: Planar (straight line) instance of Vertex Cover, max degree 3.
- **P3VC** is NP-hard. FG used to show k-center is 1.82 hard.
- To show **CN** NP-hard, just view the edges as segment objects. There is **P3VC** solution of size k iff a zero radius solution of size k for **CN**.
- To show hard to approx, just shrink each edge by ε around each vertex. Now there is a **P3VC** solution iff ε radius solution for **CN**.



- Let d be the distance between the closest non-adjacent edges. Note the correspondence with **P3VC** holds for covering balls with radius up to $d/2$, since such balls cannot cover non-adjacent edges.
- Thus we cannot approx within $(d/2)/\varepsilon$, as we could then distinguish between there being a radius ε solution or it requiring $> d/2$.
- However, we are free to choose ε to be arbitrarily small.

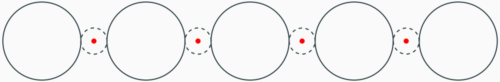
Disk Intuition



- Again start with an instance of **P3VC**.
- Replace each segment edge by a sequence of an odd number of disks. Degree three vertices are thus now three touching disks.
- Let r be the radius of the smallest ball touching all three disks at a vertex, then disks on edges spaced $2r$ apart.

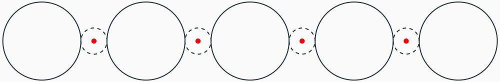
Disk Intuition

- A sequence of $2x + 1$ such disks, requires $x + 1$ radius r balls to cover it. If disk at either end (or both) is covered, then x balls required.



Disk Intuition

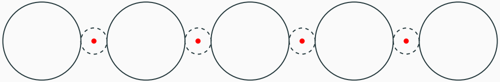
- A sequence of $2x + 1$ such disks, requires $x + 1$ radius r balls to cover it. If disk at either end (or both) is covered, then x balls required.



- Let $2x_i + 1$ be the number of disks for the ith edge, and let $X = \sum x_i$. If at least one end of every edge covered (using k "vertices") then we need X balls to cover the edges.

Disk Intuition

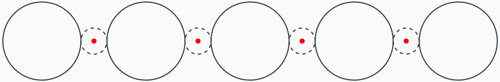
- A sequence of $2x+1$ such disks, requires $x+1$ radius r balls to cover it. If disk at either end (or both) is covered, then x balls required.



- Let $2x_i + 1$ be the number of disks for the ith edge, and let $X = \sum x_i$. If at least one end of every edge covered (using k "vertices") then we need X balls to cover the edges.
- So setting $\kappa = k + X$ for **CN**, where k was from the **P3VC** instance, then there is an r radius solution for **CN** iff a **P3VC** solution.

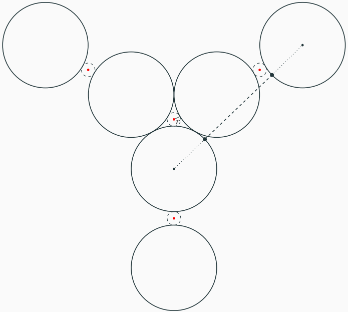
Disk Intuition

- A sequence of $2x+1$ such disks, requires $x+1$ radius r balls to cover it. If disk at either end (or both) is covered, then x balls required.



- Let $2x_i + 1$ be the number of disks for the ith edge, and let $X = \sum x_i$. If at least one end of every edge covered (using k "vertices") then we need X balls to cover the edges.
- So setting $\kappa = k + X$ for **CN**, where k was from the **P3VC** instance, then there is an r radius solution for **CN** iff a **P3VC** solution.

Correspondence holds for larger radii, up until non-adjacent disks covered by one ball. Happens at roughly $8.4 \cdot r$ ball radius.



Axis Aligned squares

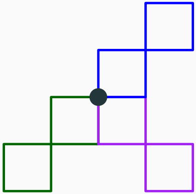
- Replacing disks with squares in the construction, similarly gives an O(1) hardness for squares.

Axis Aligned squares

- Replacing disks with squares in the construction, similarly gives an O(1) hardness for squares.
- There is an $O(1)$-approx algorithm for circles, so one might guess it applies to similarly for squares.

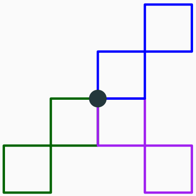
Axis Aligned squares

- Replacing disks with squares in the construction, similarly gives an O(1) hardness for squares.
- There is an $O(1)$-approx algorithm for circles, so one might guess it applies to similarly for squares.
- However, crucially, three squares can touch at a single point, allowing us to set the covering ball radius to an arbitrarily small ε, giving the same hardness level as for segments.



Axis Aligned squares

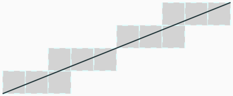
- Replacing disks with squares in the construction, similarly gives an O(1) hardness for squares.
- There is an $O(1)$-approx algorithm for circles, so one might guess it applies to similarly for squares.
- However, crucially, three squares can touch at a single point, allowing us to set the covering ball radius to an arbitrarily small ε, giving the same hardness level as for segments.



- Note we will draw squares as touching, though in the end we centrally shrink each by roughly ε

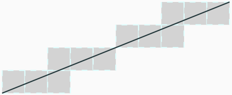
Pixelizing

- Replace each **P3VC** edge by the squares of the grid cells it overlaps.



Pixelizing

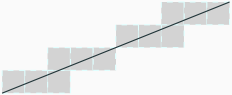
- Replace each **P3VC** edge by the squares of the grid cells it overlaps.



- On the interior of any edge we only want two squares adjacent at any point, so we clean up any degree three point.

Pixelizing

- Replace each **P3VC** edge by the squares of the grid cells it overlaps.

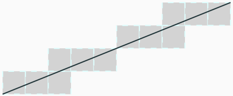


- On the interior of any edge we only want two squares adjacent at any point, so we clean up any degree three point.

- Number of squares may have wrong parity, so create a parity gadget.

Pixelizing

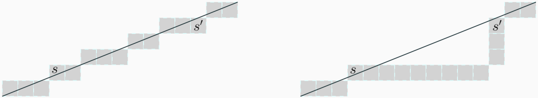
- Replace each **P3VC** edge by the squares of the grid cells it overlaps.



- On the interior of any edge we only want two squares adjacent at any point, so we clean up any degree three point.

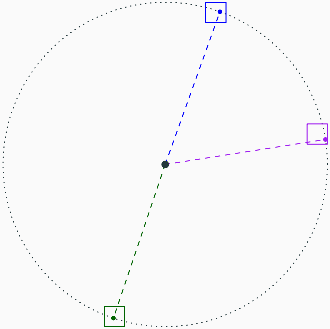
- Number of squares may have wrong parity, so create a parity gadget.

- For non-horizontal edges, can replace long stretch of squares with a horizontal then vertical section, where we can insert parity gadget. (Requires all edges have a large constant number of squares.)



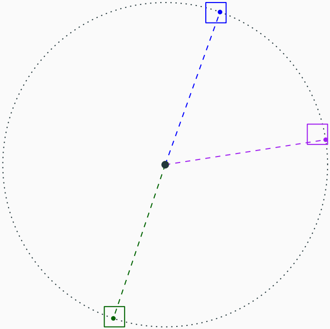
Aligning Near Vertices

- Scale so that around each **P3VC** vertex there is a large ball with no other vertex.



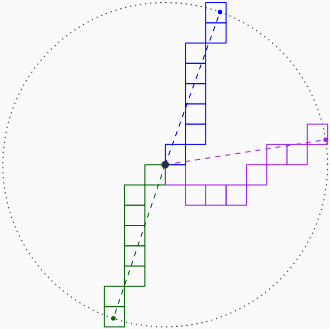
Aligning Near Vertices

- Scale so that around each **P3VC** vertex there is a large ball with no other vertex.
- Mark the grid cells where the edges intersect the ball.



Aligning Near Vertices

- Scale so that around each **P3VC** vertex there is a large ball with no other vertex.
- Mark the grid cells where the edges intersect the ball.
- If scaling sufficient these cell are far apart, and so one can route to them, with paths that stay far apart except at vertex. (Note simply pixelating may put the paths too close.)

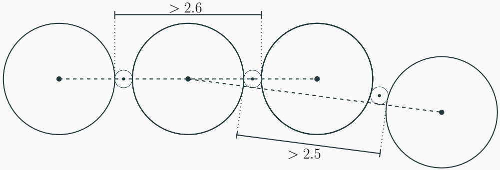


Edge Bending

- Started with **P3VC** instance with integer vertex coordinates. However, edges may be non-integer, or more precisely not the right length to odd number of equal space disks/points.

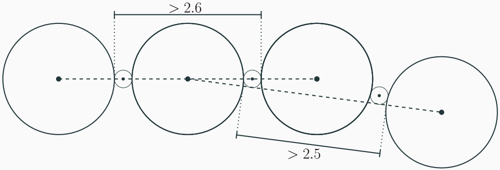
Edge Bending

- Started with **P3VC** instance with integer vertex coordinates. However, edges may be non-integer, or more precisely not the right length to odd number of equal space disks/points.
- In the original k-center reduction of Feder and Greene, and for **CN** with disks, rectified by "edge bending", i.e. with $O(1)$ disks an edge can be bent in any direction (and hence lengthened precisely).



Edge Bending

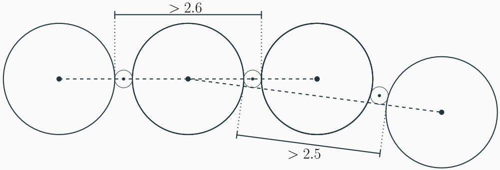
- Started with **P3VC** instance with integer vertex coordinates. However, edges may be non-integer, or more precisely not the right length to odd number of equal space disks/points.
- In the original k-center reduction of Feder and Greene, and for **CN** with disks, rectified by "edge bending", i.e. with $O(1)$ disks an edge can be bent in any direction (and hence lengthened precisely).



- For squares such bending presents multiple problems (axis-aligned, super constant hardness).

Edge Bending

- Started with **P3VC** instance with integer vertex coordinates. However, edges may be non-integer, or more precisely not the right length to odd number of equal space disks/points.
- In the original k-center reduction of Feder and Greene, and for **CN** with disks, rectified by "edge bending", i.e. with $O(1)$ disks an edge can be bent in any direction (and hence lengthened precisely).



- For squares such bending presents multiple problems (axis-aligned, super constant hardness).
- However, since we insisted on aligning squares with grid cells, bending is unnecessary.

Result

The rest of the analysis is similar to that shown for disks, yielding:

Theorem

CN cannot be approximated within any factor in polynomial time unless $\mathrm{P} = \mathrm{NP}$, even when restricting to the set of instances in which the objects are a set of axis aligned squares of the same size.

Thank You